

관계형 데이터베이스 기반의 RDF와 OWL의 저장 및 질의처리

(RDF and OWL Storage and Query Processing based on
Relational Database)

정 호 영 [†] 김 정 민 [†] 정 준 원 [†]

(Hoyoung Jeong) (Jungmin Kim) (Junwon Jung)

김 종 남 [†] 임 동 혁 [†] 김 형 주 ^{**}

(Jongnam Kim) (Donghyuk Im) (Hyoung-Joo Kim)

요 약 컴퓨터의 발전에도 불구하고 수많은 전자 문서가 범람하는 현재와 같은 상황에서 원하는 정보를 얻기는 더욱 어려워지고 있다. 이와 같은 상황에서 데이터를 얼마나 빠르게 처리하는지 보다 얼마나 의미 있는 정보를 얻을 수 있는지가 중요한 문제가 되고 있다. 이에 웹 자원에 대해 지식기반의 메타 데이터를 기술하고 이를 이용한 지능적인 정보 처리를 목적으로 하는 것이 시멘틱 웹이다 또한 시멘틱 웹이 발전함에 따라 웹 자원 못지않게 지식자원에 대한 중요성도 커질 수 밖에 없다. 본 논문에서는 시멘틱 웹을 기술하는 W3C표준 언어인 OWL에 대해 효율적이고 의미 있는 질의처리를 제공할 수 있는 데이터베이스의 저장 시스템 및 OWL-QL 질의 처리기를 제공한다.

키워드 : 시멘틱 웹, 온톨로지, OWL, 저장, 질의, OWL-QL

Abstract In spite of the development of computers, the present state that a lot of electronic documents are overflowing makes it more difficult for us to get appropriate information. Therefore, it's more important to focus on getting meaningful information than processing the data quickly. In this context, Semantic Web enables an intelligent processing by adding semantic metadata on your web documents. Also, as the Semantic Web grows, the knowledge resources as well as web resources are getting more and more importance. In this paper, we propose an OWL storage system aiming at an intelligent processing by adding semantic metadata on your web documents, plus a system aiming at an OWL-QL Query Processing.

Key words : Semantic Web, Ontology, OWL, Storage, Query, OWL-QL

1. 서 론

1989년 Tim Berners-Lee에 의해 제안된 월드 와이드

웹(World Wide Web, WWW)은 단순함과 함께 HTML과 같이 비교적 쉬운 언어를 이용하여 편리성을 추구한 덕분에 일반 사용자 누구나 쉽게 정보를 게시하거나 접근할 수 있게 되었고, 이는 폭발적인 정보의 증가를 가져왔다. 정보의 급속한 확산에 따라 컴퓨터분야의 연구는 용량과 속도에 대한 문제만큼이나 원하는 정보를 효율적으로 관리하고 이용하는 것에 대한 중요성이 증대되었다. 이렇게 정보를 효율적으로 관리하기 위한 방법으로 데이터베이스와 같이 데이터 자체를 구조화 시켜 저장하거나 XML처럼 데이터의 구조를 기술하고 이를 통해 효율적으로 데이터를 처리하는 방식들이 있다 [1,2]. 이와 같이 데이터에 구조를 부여해서 효율적으로 처리하는 것과는 또 다른 접근방법으로서 정보에 대한

· 본 연구는 두뇌한국21 사업과 정보통신부의 대학IT연구센터(ITRC) 지원을 받아 수행되었음

[†] 학생회원 : 서울대학교 전기.컴퓨터공학부

hyjung@oopsla.snu.ac.kr

jmkim@oopsla.snu.ac.kr

jwjung@oopsla.snu.ac.kr

jnkim@oopsla.snu.ac.kr

dhlhm@oopsla.snu.ac.kr

^{**} 종신회원 : 서울대학교 전기.컴퓨터공학부 교수

hjk@snu.ac.kr

논문접수 : 2004년 10월 5일

심사완료 : 2005년 4월 20일

의미적인 메타 데이터를 기술함으로써 정보의 검색이나 관리에 있어서 효과적인 처리를 도모하고자 하는 시맨틱 웹(Semantic Web)과 같은 방법이 있다. 전자책의 방식이 처리비용이나 속도를 고려한 데이터 관리 효율에 중점을 둔 방법이라면 후자의 방식은 의미가 있는 데이터를 구축하고 지능적인 처리를 수행하는 것에 중점을 둔 방식이다[3]. 차세대 웹이라 불리는 시맨틱 웹이 발전함에 따라 지식 자원에 대한 관리와 효율적인 정보 처리 기술은 점점 더 중요해 질 것이다.

본 논문에서는 RDF(Resource Description Framework), OWL(Ontology Web Language)에 대해 효율적이고 의미 있는 질의처리를 제공할 수 있는 데이터베이스의 저장 시스템 및 OWL-QL(OWL-Query Language) 질의 처리기를 제공한다. 2장에서는 온톨로지(Ontology)의 저장 모델 및 질의 처리에 대한 배경지식에 대해 살펴보고 3장에서는 관련 연구로서 HP의 Jena 시스템과 IBM의 SnoBase시스템에 대해 알아본다. 4장에서는 시스템 구현에 대한 구성이고 5장은 결론 및 향후연구과제로 구성된다.

2. 온톨로지 저장 모델 및 질의 처리

이 장에서는 관계형 데이터베이스를 이용한 범용적인 온톨로지의 저장 모델과 질의 처리에 대해 소개한다

2.1 저장 모델

RDF 모델은 기본적으로 Subject-Predicate-Object의 트리플 구조로 표현할 수 있으며 이는 저장 방법에 있어 중요한 역할을 한다. 그림 1은 RDF를 관계형 데이터베이스에 저장하는 여러 접근법에 대한 설계이다[4].

- 수평 모델(Horizontal Model): 트리플 모델을 통합적인 테이블에 저장하는 방법으로서 모든 트리플 모델은 하나의 Statement테이블에 저장되는 방식으로 모델은 간단하지만 데이터의 중복이 심하고 테이블의 크기가 커지는 단점이 있다.
- 수직 모델(Vertical Model): Subject-PredicateObject는 각각 별도의 테이블에 저장하며 트리플 모델을 표현하는 Statement테이블의 데이터들은 각각의 테이블을 참조하게 된다. 데이터의 중복은 없지만 질의를 처리하기 위해서는 많은 조인 연산을 요구한다

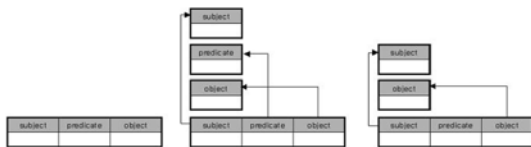


그림 1 수평 모델, 수직 모델, 혼합 모델의 관계형 데이터베이스 테이블 구조

- 혼합 모델(Hybrid Model): 위 두 가지 방법을 혼합한 형태로서 Subject와 Object는 별도의 테이블에 의해 관리되고 Statement테이블은 속성값을 가지며 Subject와 Object는 별도의 테이블을 참조하는 방식이다. 데이터의 중복과 조인 연산은 위 두 가지 방법의 중간이라 할 수 있다[5].

2.2 질의처리

온톨로지의 저장 시스템을 설계하는 목적은 대용량의 데이터에 대한 유지관리도 중요하겠지만 궁극적으로는 효율적인 지식추출을 위한 질의처리를 지원해야 하는 것은 분명할 것이다. 온톨로지에 대한 질의어로서는 RDF데이터 기반의 RDQL과 RDF(S)기반의 RQL, 그리고 RDF(S), OWL기반의 OWL-QL이 있다. RDF(S), OWL이 XML을 근간으로 하는 것을 고려하면 XPath(XQuery)도 질의어라 할 수 있는데 각각의 차이점에 대해 알아본다. XML, RDF, RDFS, OWL에 대한 질의를 위 세가지 언어에 기초하여 차이점을 설명하면 아래와 같다.

1. XML문서의 문법적 질의
2. RDF 트리플 데이터모델에서의 구조적 질의
3. RDFS, OWL 스키마모델에서의 의미적 질의

2.2.1 문법적 질의

RDF(S), OWL문서가 XML의 문법 규칙을 따르고 있기에 질의를 수행함에 있어서 XPath와 같은 XML 질의어를 사용할 수 있을 것이다. 하지만 XPath의 생성된 목적이 그러하듯이 질의어는 RDF를 더 이상 의미적인 문서가 아닌 XML문서로 볼 것이다. XML 모델과 RDF 모델의 차이점은 XML 모델은 트리 모델이고, RDF 모델은 간선에 의미(semantic label)가 부여된 그래프 모델이라는 점이다. 이는 Subject <http://147.46.121.163/go#GO_0016209>와 Object “antioxidant activity”간에 어떠한 관계에 있는지에 관한 관계질을 수행할 수 없다. 단지 노드 “antioxidant activity”가 노드 <http://147.46.121.163/go#GO_0016209>안에 포함되어 있다는 관계를 찾을 수 있을 것이다.

2.2.2 구조적 질의

RDF(S), OWL문서에 대한 구조적 질의를 수행하는 질의어로서는 대표적으로 RDQL과 같은 언어가 있다. RDQL은 모든 RDF(S)문서를 트리플 모델의 관점에서 질의를 수행하게 된다. 예를 들어 <http://147.46.121.163/go#name>이 “antioxidant activity”인 Subject를 추출하려면 다음과 같이 질의할 수 있다.

```
SELECT ?subject
Where (?subject, <http://147.46.121.163/go#name>,
“antioxidant activity”)
```

subject

<http://147.46.121.163/go#GO_0016209>

위와 같은 질의에 대해 결과<http://147.46.121.163/go#GO_0016209>를 기대할 수 있다. 하지만 <http://147.46.121.163/go#GO_0016209>가 클래스(class)인지 인스턴스인지는 알 수가 없다. 즉 질의를 함에 있어서 타입, 인스턴스와 클래스간의 관계 클래스와 클래스간의 관계와 같은 스키마 정보를 이용하지 못한다. 이는 RDQL과 같은 구조적 질의어는 모든 RDF(S), OWL문서를 트리플 모델의 데이터로만 인식하기 때문에 구조적으로 일치하고 값이 일치하는 노드나 간선을 찾아준다.

2.2.3 의미적 질의

RDF(S), OWL문서에 대한 의미적 질의를 수행하기에 적합한 언어로는 RQL, OWL-QL이 있다. RDQL이 RDF 모델에서 구조와 데이터 관점에 대한 질의를 잘 수행할 수 있음에도 불구하고 의미적 질의를 수행하기에 부족한 것은 스키마에 대한 처리 능력 부족일 것이다. RDFS, OWL에 기반한 의미적 질의는 RDF에 대한 구조나 제약조건, 타입 및 인스턴스 관계, 클래스나 객체들의 계층적 구조를 기술하는 것 이외에도 의미적 기반의 추론도 가능케 한다는 점에서 구조적 질의의 확장이라 할 수 있다.

3. 관련 연구

이 장에서는 온톨로지 관리 시스템인 HP 연구소의 Jena와 IBM 연구소의 SnoBase시스템의 저장 구조와 질의 처리에 대해 알아본다.

3.1 Jena

Jena는 Java로 개발되어졌고 HP 연구소에서 개발된 온톨로지 관리 시스템으로서 시멘틱 웹을 개발하는 사람이라면 누구에게나 공개적으로 제공되며 시멘틱 웹의 응용을 제작하는 사람들의 표준 개발틀이라 할 수 있다. 현재는 Jena1에 이어 Jena2가 발표되었으며, 시스템의 소개는 Jena2를 위주로 소개한다.

Jena는 RDF(S), OWL문서의 파서(Parser), RDF(S), OWL의 in-memory객체인 Model, OntModel과 데이터베이스를 이용한 Persistent객체(eg. Postgresql, MySQL, Oracle, Berkeley DB)를 지원하며, 질의어로는 RDQL과 그리고 개발자를 위한 여러 기능적인 API를 제공한다. 그림 2는 Jena2에서 제공하는 저장 시스템의 관계형 데이터베이스 모델이다[6].

Jena1이 테이블 모델 중 수직 모델을 사용했던 것에 반해 Jena2는 기본적으로는 수평 모델방식의 혼합 모델을 사용하였다. 이는 데이터의 중복에 의한 테이블 크기가 커지는 단점을 감수한 반면 Jena1의 단점이기도 했

Statement Table

Subject	Predicate	Object
mylib:doc1	dc:title	Jena2
mylib:doc1	dc:creator	HP Labs - Bristol
mylib:doc1	dc:creator	Hewlett-Packard
mylib:doc1	dc:description	101
201	dc:title	Jena2 Persistence
201	dc:publisher	com.hp/HPLaboratories

Literals Table

Id	Value
101	The description - a very long literal that might be stored as a blob.

Resources Table

Id	URI
201	hp:aResource-WithAnExtremelyLongURI

그림 2 Jena2 저장 시스템의 관계형 테이블 구조

던 조인 연산을 줄이기 위한 접근 방법이다 또한 그림 2의 'Resource Id 201'과 'Literal Id 101'과 같이 일부 데이터의 크기가 큰 자원에 대해서는 중복을 피하고자 테이블을 따로 관리하고 Statement 테이블에서 참조하는 방식을 사용하였다. 테이블로 관리되는 데이터는 적절한 임계점(threshold)을 넘는 크기를 가지는 데이터이다.

Jena시스템은 질의처리의 언어로 RDQL을 제공한다. <dc:title>의 값으로 "Jena2"를 가지는 Subject와 그 Subject에 대해<dc:creator>의 값을 구하고자 하는 질의는 다음과 같이 표현할 수 있다.

```
SELECT ?Var1, ?Var2
Where(Var1, <dc:title>, "Jena2"), (Var1, <dc:creator>, Var2)
```

이 질의에 대한 결과를 얻기 위해서 Jena는 RDQL질의를 적절한 SQL문장으로 변환하게 되는데 이를 SQL문장으로 변환하면

```
SELECT p1.subject, p2.object
FROM stmt_table p1, stmt_table p2
WHERE p1.predicate = 'dc:title' and p1.object = 'Jena2' and
p2.predicate = 'dc:creator' and p1.subject = p2.subject
```

위와 같이 된다. 이에 대한 결과는 아래와 같다.

Var1	Var2
<mylib:doc1>	"HP Labs-Bristol"
<mylib:doc1>	"Hewlett-Packard"

3.2 SnoBase

SnoBase도 Java로 개발되어졌고 IBM 연구소에서 개발된 온톨로지 관리 시스템으로서 Jena시스템과 유사한 점이 많이 있다. 그림 3은 SnoBase시스템의 관계형 데이터베이스 모델이다[7].

SnoBase 또한 테이블 모델 중 수평 모델을 사용하였

Statement 테이블

Subject	Verb	Object
http://147.46.121.163/go#GO_0016209	http://147.46.121.163/go#name	antioxidant activity
http://147.46.121.163/go#GO_0016209	http://www.w3.org/2000/01/rdf-schema#comment	Inhibition of the ...
http://147.46.121.163/go#GO_0016209	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://147.46.121.163/go#GO_0003674
http://147.46.121.163/go#GO_0016209	http://147.46.121.163/go#association	blank node1
blank node1	http://147.46.121.163/go/go#gene_product	blank node2
blank node2	http://147.46.121.163/go/~hyjung/go#name	4930414c22Rik

그림 3 SnoBase 저장 시스템의 관계형 테이블 구조

다. 이는 Jena시스템과 마찬가지로 구현의 단순함과 함께 여러 가지 장점을 가질 수 있다

SnoBase시스템은 질의처리의 언어로 RDQL과 SQL을 제공한다. <http://147.46.121.163/go#name>의 값으로 "antioxidant activity"를 가지는 Subject와 그 Subject에 대해 <http://www.w3.org/2000/01/rdf-schema#comment>의 값을 구하고자 하는 질의는 다음과 같이 표현할 수 있다.

```
SELECT ?Var1, ?Var2
Where(Var1, <http://147.46.121.163/go#name>,
      "antioxidant activity"),
      (Var1, <http://www.w3.org/2000/01/rdfschemas#comment>, Var2)
```

이 질의에 대한 결과를 얻기 위해서 SnoBase는 RDQL 질의를 적절한 SQL문장으로 변환하게 되는데 이를 SQL문장으로 변환하면 아래와 같다.

```
SELECT s1.Subject, s2.Object
FROM Statement s1, Statement s2
WHERE s1.Verb = 'http://147.46.121.163/go#name'
and s1.Object = 'antioxidant activity'
and s2.Verb = 'http://www.w3.org/2000/01/rdf-schema#comment' and s1.Subject = s2.Subject
```

위 질의는 SnoBase에서 지원하는 SQL질의어 이기도 하다. 이에 대한 결과는 아래와 같다.

```
Var1 Var2
<http://147.46.121.163/go#GO_0016209> "Inhibition of the ..."
```

4. 시스템 설계 및 구현

3장에서 Jena와 SnoBase시스템에 대해 알아보았다. 두 시스템은 공통적으로 영속적인(persistent) 데이터를 관리하는데 있어서 수평 모델을 사용하였는데 단순함에서

오는 몇몇 장점을 취할 수는 있으나 데이터의 관점에서 처리를 함으로써 질의 처리에 있어서 스키마 정보를 이용하는 의미적인 질의를 해결할 수 없는 부족함이 있다.

그림 4는 시스템의 전반적인 구조이다 시스템의 구성은 온톨로지의 검색 및 브라우징을 지원하는 웹 기반의 응용 프로그램과 API 및, equivalent, inverseOf, subClassOf, intersectionOf등 클래스간의 관계를 유추해 낼 수 있는 추론기(reasoner), 그리고 효율적인 질의 처리를 수행하기 위한 색인 모듈이 있다 또한 질은 색으로 표시된 논문에서 제안하는 RDF(S), OWL 저장소와 OWL-QL 질의 처리기로 구성된다. 그림 5는 RDF(S), OWL 모델을 위한 관계형 데이터베이스의 테이블 구조로서 RDF에 대해서는 수직 모델을 사용하였고 스키마기반의 질의처리를 지원하기 위해 RDFS, OWL에대한 테이블 구조를 제안한다.

4.1 RDF(S), OWL 저장소

RDF(S), OWL을 위한 저장 시스템은 크게 두 부분으로 나뉜다. 데이터 기반인 RDF 저장을 위한 rdfre

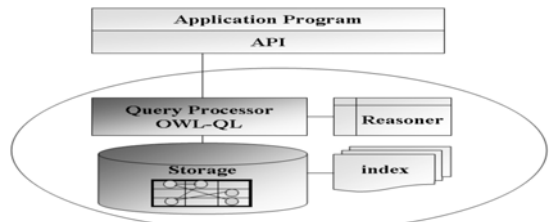


그림 4 시스템 구조

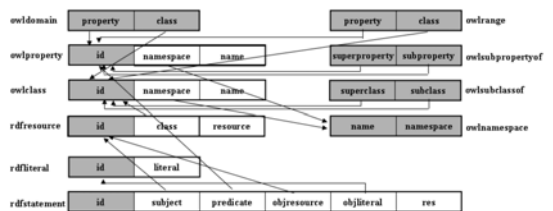


그림 5 RDF, OWL 모델을 위한 관계형 데이터베이스 테이블 구조

source, rdfs:literal, rdfs:statement 테이블과 스키마 기반인 RDFS, OWL 저장을 위한 owl:namespace, owl:class, owl:property, owl:subclassof, owl:subpropertyof, owl:domain, owl:range 테이블로 나뉜다. 예를 들어 'gene_product'의 타입을 가지는 인스턴스 '4930414C22Rik'는 리소스 테이블을 저장하는 데이터 기반의 rdfs:resource 테이블은 자원의 고유 식별자, 자원의 타입인 클래스의 식별자, 그리고 자원의 인스턴스 값을 가지게 되며 그 인스턴스의 타입 정보를 가지는 owl:class 테이블은 클래스의 고유 식별자, 클래스가 정의된 네임 스페이스 그리고 클래스 이름을 가지게 된다. 다음은 Jena 파서를 이용하여 생물 정보학 온톨로지인 GO(Gene Ontology)를 각 테이블에 저장한 예제와 함께 테이블에 대한 설명을 보인다

owl:namespace: 네임 스페이스를 관리하기 위한 테이블이다. 네임 스페이스는 RDF(S), OWL의 클래스와 인스턴스가 기술되는 영역이다 즉 이 네임 스페이스로 인해 여러 상이한 영역의 RDF문서를 저장 및 처리할 수 있으며 RDF의 질의가 미치는 기본 영역이기도 하다. 네임 스페이스의 이름인 name과 영역인 namespace로 구성되어 있다. 네임 스페이스의 name과 namespace는 주키이며 색인되어 있다.

id	namespace
base	http://147.46.121.163/go#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#
go	http://147.46.121.163/go#

owl:class: OWL의 클래스 식별자, 네임 스페이스와 그 클래스의 이름을 저장하는 테이블이다. 클래스의 식별자인 id, 네임스페이스 namespace, 이름인 name으로 구성되어 있다. id는 주키이며 색인되어 있다.

id	namespace	name
1	go	GO_0003673
2	go	GO_0003674
3	go	GO_0016209
4	go	evidence_code
5	go	database_symbol
6	go	gene_product
7	go	reference

owl:subclassof: OWL 클래스에 대한 'is_a' 관계를 저장하는 테이블이다. 상위 클래스의 식별자인 superclass와 하위 클래스의 식별자인 subclass로 구성되어 있다.

다. superclass와 subclass는 주키이며 색인되어 있고 owlclass의 id를 참조한다.

superclass	subclass
1	2
2	3

owl:property: OWL의 속성(property)에 대한 식별자, 네임스페이스와 속성의 이름을 저장하는 테이블이다. 속성의 식별자인 id, 네임 스페이스 namespace와 이름인 name으로 구성되어 있다. id는 주키이며 색인되어 있다.

id	namespace	name
1	go	association
2	go	dbxref
3	go	gene_association
4	go	dbxreference
5	go	name

owl:subpropertyof: OWL 속성에 대한 'is_a' 관계를 저장하는 테이블이다. 상위 속성의 식별자인 superproperty와 하위 속성의 식별자인 subproperty로 구성되어 있다. superproperty와 subproperty는 주키이며 색인되어 있고 owlproperty의 id를 참조한다.

owl:domain: OWL 속성의 정의역(domain)을 저장하는 테이블이다. 속성의 식별자인 property와 도메인의 식별자인 class로 구성되어 있다. property와 class는 주키이며 색인되어 있고 각각 owlproperty의 id와 owlclass의 id를 참조한다.

owl:range: OWL 속성의 치역(range)을 저장하는 테이블이다. 속성의 식별자인 property와 치역의 식별자인 class로 구성되어 있다. property와 class는 주키이며 색인되어 있고 각각 owlproperty의 id와 owlclass의 id를 참조한다.

property	class
1	4
2	5
3	6
4	7

rdfs:resource: RDF 트리플 모델의 자원(Resource)을 저장하는 테이블이다. 자원은 RDF 트리플 구문에서 Subject이거나 Subject를 기술하는 Object중 다른 트리플 구조에서 Subject로 표현되어질 수 있는 자원으로 테이블에 저장되는 값은 중복을 피하게 된다. 자원의 식

별자인 id, 타입인 class와 URI 값인 resource로 구성되어 있다. id는 주키이며 색인되어 있다. class는 owlclass의 id를 참조한다.

id	class	resource
1	4	ISS
2	5	SP_KW
3	6	4930414C22Rik
4	7	Antioxidant

rdfliteral: RDF 트리플 모델의 Object중에서 리터럴 값을 저장하는 테이블이다. 리터럴은 RDF 트리 구조에서 중단 노드에 해당하는 노드이다. 리터럴의 식별자인 id와 리터럴의 값인 literal로 구성되어 있다. id와 literal은 주키이며 색인되어 있다.

id	literal
1	molecular_function
2	antioxidant activity
3	glutathione dehydrogenase

rdfstatement: RDF구문의 트리플 모델을 저장하는 테이블이다. 트리플 모델의 식별자인 id와 subject, predicate, objresource와 objliteral, 그리고 Object가 지원인지 리터럴인지의 플래그 값인 res로 구성되어 있다. 트리플 모델의 id는 주키이며 색인되어 있고 subject, predicate, objresource와 objliteral은 각각rdfresource, rdfpredicate, rdfresource와 rdfliteral의 id를 참조한다.

id	subject	predicate	objresource	objliteral	Res
1	5	2	2		1
2	6	3	4		1

4.2 OWL-QL 질의 처리기

본 시스템의 저장소는 Jena나 SnoBase시스템과 달리 스키마 기반의 질의를 지원하기 위해 RDF와 함께 RDFS, OWL에 대한 저장을 지원한다. 질의어로는 OWL-QL을 사용하고 관계형 데이터베이스를 사용하는 저장소에서 원하는 질의를 수행하기 위해서는 SQL로의 변환이 필요하다. 질의는 크게 다음과 같은 세 부분으로 나눌 수 있다.

1. 인스턴스간의 관계 Subject-Property-Object의 인스턴스와 속성 속성값으로 표현되는 데이터 기반의 트리플 관계
2. 인스턴스와 클래스간의 관계 인스턴스-타입-클래스의 트리플 관계
3. 클래스간의 관계 클래스와 클래스간 계층적 구조 관계

다음은 시스템에서 지원하는 간단한 OWL-QL의 예로서 “항산화 작용을 하는 어떤 유전자 산출물(Gene Product)의 상위 term(작용, 개념, 클래스)을 찾는 질의”이다. GO에서 term은 OWL의 클래스에 해당되며 RDQL과 같은 데이터 기반의 질의에서는 처리 할 수 없는 형태의 질의이다.

- i. (go:name ?x "antioxidant activity")
 - ii. (type ?x, ?y)
 - iii. (subClassOf ?y ?z)
- must-bind ?z

i.의 질의는 인스턴스간의 질의로서 rdfstatement 테이블에서 중간결과를 구할 수 있다. <go:name>이라는 속성값으로 "antioxidant activity"를 가지는 모든 Subject를 찾는 질의로서 OWL-QL에 대한 질의는 다음과 같은 SQL의 형태로 표현된다.

```
Select r.resource
from rdfstatement s, rdfresource r, owlproperty p,
rdfliteral l, owlnamespace n
where s.predicate = p.id and
p.name = 'name'and
p.namespace = n.name and
n.namespace = 'go' and
s.objliteral = l.id and
l.literal = 'antioxidant activity' and
s.res = 2 and
s.subject = r.id
```

ii.의 질의는 인스턴스와 클래스간의 질의이다 이는 rdfresource 테이블에서 구할 수 있다. i.에서 구한 중간값이 모든 Subject에 대해 모든 타입을 추출하는 질의로서 다음과 같은 SQL의 형태로 표현된다.

```
Select class
from rdfresource
where resource = subject
```

iii.의 질의는 클래스간의 질의이다. 이는 owlsubclassof 테이블을 통해 구할 수 있다. ii.에서 구한 중간값인 모든 class에 대해 상위 클래스를 추출하는 질의로서 다음과 같은 SQL의 형태로 표현된다.

```
Select c.name
from owlclass c, owlsubclassof s, owlnamespace n
where s.subclass = class and
```

c.id = s.superclass and
 c.namespace = n.name and
 n.namespace = 'go'

i. ii. iii. 질의에 대한 결과는 아래와 같다.

z
 GO_0003674

위와 같이 OWL-QL의 질의는 크게 세 부분으로 나누어 볼 수 있는데 그림 6은 질의 처리기 구현의 과정을 나타낸다.

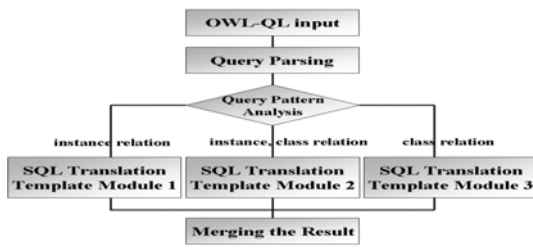


그림 6 질의 처리 과정

사용자는 웹 응용 프로그램을 통해 키워드 검색을 하면 적절한 온톨로지 질의어로 변환된 OWL-QL을 입력으로 한다. 질의 입력은 Query Parsing 모듈을 통해 다시 트리플 모델별로 질의 패턴을 분석하게 된다. 질의 분석은 각각의 트리플 질의를 데이터간의 질의, 데이터와 클래스간의 질의, 클래스와 클래스간의 질의별로 i. ii. iii.과 같은 SQL 변환 템플릿 모듈을 통해 각 단계별의 중간 결과를 산출해 내고 각 중간값을 종합하여 결과를 얻는다.

5. 결론 및 향후 연구과제

본 논문에서는 웹 자원에 대해 지식구조를 기술하는 RDF(S), OWL에 대한 저장 시스템과 의미적 기반의 질의를 수행할 수 있는 OWL-QL 질의 처리기에 대한 설계 및 구현을 제시한다. 제안하는 저장방식은 시멘틱 웹이 발전함에 따라 증가할 수 있는 온톨로지 스키마와 데이터를 효율적으로 저장, 관리, 지식 추출을 할 수 있으며 특히 대용량의 지식기반 문서관리에 유용하다 또한 시멘틱 웹에서 사용할 수 있는 웹 자원 전자 문서, 오프라인 서적에 이르는 특히 대용량의 콘텐츠 관리 시스템에 대해 큰 도움이 될 것으로 보인다.

본 시스템에서 제안하는 저장방식은 수직 모델을 사용하였는데 조인 연산 비용이 많이 드는 단점이 있다. 이는 그래프 모델을 관계형 데이터베이스 모델에 매핑

하는 근본적인 문제점이기도 한데 추후 데이터 색인 경로 색인에 대한 연구가 이루어져야 할 것이다

참고 문헌

[1] Bebee, B.R.; Mack, G.A., "Distributed meta data objects using RDF", Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999.(WET ICE '99) Proceedings. IEEE 8th International Workshops on, 1999.

[2] Sean B. Palmer. "The Semantic Web:An Introduction", <http://infomesh.net/2001/swintro/>

[3] Dieter Fensel; Ian Horrocks; Frank van Harmelen; Stefan Decker; Michael Erdmann; Michel C. A. Klein. "OIL in a Nutshell",. EKAW 2000, pp. 1-16.

[4] James Hendler. "Agents and the Semantic Web", IEEE Intelligent Systems Journal, March/April 2001.

[5] Yannis Papakonstantinou and Victor Vianu, "DTD inference for views of XML data", Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.

[6] Kevin Wilkinson, Craig Sayers, Harumi Kuno, Dave Reynolds, "Efficient RDF Storage and Retrieval in Jena2.

[7] <http://www.alphaworks.ibm.com/tech/snobase>

정 호 영

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 4 호 참조

김 정 민

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 3 호 참조

정 준 원

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 4 호 참조

김 종 남

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 4 호 참조

임 동 혁

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 4 호 참조

김 형 주

정보과학회논문지 : 컴퓨팅의 실제
 제 11 권 제 3 호 참조