

## 시맨틱 웹 문서를 위한 관계형 저장 스키마 설계 및 질의 처리 기법

이 순 미\*

### Design of Relational Storage Schema and Query Processing for Semantic Web Documents

Lee, Soonmi\*

#### 요 약

최근 들어 온톨로지 문서의 활용이 증가하고 있는 추세 속에서 시맨틱 정보를 효율적으로 검색하기 위해서는 온톨로지 데이터를 효과적으로 저장 및 질의 처리를 할 수 있는 관리 시스템이 필요하다. 본 논문에서는 W3C에서 제안한 온톨로지 언어인 RDF/RDFS를 기반으로 하는 시맨틱 웹 문서를 관계형 데이터베이스에 저장하고 효율적으로 검색하기 위한 저장 스키마를 제안한다. 특별히 제안한 저장 스키마는 계층 정보를 효과적으로 검색할 수 있도록 설계하여 질의 처리의 효율성을 증가시킨다. 또한 본 논문에서는 RQL 시맨틱 질의를 SQL로 변환하여 질의를 처리하는 메커니즘을 기술하며 MS-ACCESS를 사용하여 데이터베이스를 구축 및 구현한다. 구현 결과를 통하여 트리플 모델에 기반한 데이터 질의 뿐만 아니라 스키마나 계층정보에 대한 질의도 간단하게 SQL로 변환됨을 알 수 있다.

#### Abstract

According to the widespread use of ontology documents, a management system which store ontology data and process queries is needed for retrieving semantic information efficiently. In this paper I propose a storage schema that stores and retrieves semantic web documents based on RDF/RDFS ontology language developed by W3C in a relational databases. Specially, the proposed storage schema is designed to retrieve efficiently hierarchy information and to increase efficiency of query processing. Also, I describe a mechanism to transform RQL semantic queries to SQL relational queries and build up database using MS-ACCESS and implement in this paper. According to the result of implementation, we can know that not only data query based on triple model but also query for schema and hierarchy information are transformed simply to SQL.

▶ Keyword : 시맨틱 웹(Semantic Web), 온톨로지(Ontology), RDF, RDFS, RQL

• 제1저자 : 이순미

• 투고일 : 2008. 11. 11, 심사일 : 2008. 11. 11, 게재확정일 : 2009. 1. 20.

\* 경인여자대학 정보미디어학부 부교수

※ 본 연구는 경인여자대학의 연구비 지원으로 수행되었음.

## I. 서론

오늘날의 월드 와이드 웹(World Wide Web)은 빠른 속도로 진보를 거듭하며 발전하고 있다. 그러나 웹 콘텐츠의 양이 점점 증가함에 따라서 야후나 구글과 같은 키워드 기반의 검색 엔진을 사용해서는 사용자가 원하는 정확한 정보를 얻을 수 없는 문제점이 발생한다. 이러한 문제점을 개선하기 위하여 팀 버너스-리(Tim Berners-Lee)가 시맨틱 웹(Semantic Web) 개념을 제안하였다. 시맨틱 웹을 구축하기 위해서는 온톨로지(Ontology) 언어를 사용해야 하는데 대표적인 온톨로지 언어로는 RDF(1), RDFS(RDF Schema)(2), DAML+OIL(3), OWL(4) 등이 있다.

최근 들어 온톨로지 문서의 활용이 증가하고 있는 추세 속에서 시맨틱 정보를 효율적으로 검색하기 위해서는 온톨로지 데이터를 효과적으로 저장 및 질의 처리를 할 수 있는 온톨로지 관리 시스템이 필요하다. 따라서, 본 논문에서는 RDF/RDFS를 기반으로 하는 온톨로지 문서를 현재 상용화되고 있는 데이터베이스 시스템 중에서 가장 안정적이면서도 널리 사용되고 있는 관계형 데이터베이스에 저장 및 검색하는 기법을 제안하고자 한다. 특별히 계층 정보를 효과적으로 검색할 수 있는 저장 스키마를 제안하였으며 제안한 방법을 따라 저장된 데이터에 대한 시맨틱 질의를 처리하는 메카니즘을 기술하였다. 본 논문에서는 시맨틱 질의어로 RQL(5)을 사용하였는데 그 이유는 RQL은 트리플 기반의 데이터 검색 뿐만 아니라 계층

정보를 포함한 스키마 정보에 대한 질의도 가능하기 때문이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하며 3장에서는 본 논문에서 제안한 온톨로지 저장 기법을 소개한다. 4장에서는 RQL 질의 처리 방법을 기술하며 5장에서는 RQL 질의가 SQL로 변환되어 처리되는 과정을 질의 유형별로 나누어 구현하고 마지막으로 6장에서 결론을 맺는다.

## II. 관련연구

온톨로지 질의어는 문법적 질의어, 구조적 질의어, 의미적 질의어로 분류할 수 있다. 문법적인 질의어로는 XPath가 있으며 구조적 질의어로는 HP Lab.에서 개발한 RDQL이 있다. 의미적 질의어로는 RQL(5)과 OWL-QL(6)이 있다. 의미적 질의어는 RDF에 대한 구조나 제약조건, 타입과 인스턴스 관계, 클래스나 속성의 계층관계를 질의에 반영시킬 수 있다. RDQL이 스키마나 계층정보에 대한 처리가 미흡한데 비하여 RQL 질의어는 RDF 뿐만 아니라 RDFS를 위한 의미적 질의어로서 경로표현이나 스키마 인식 등이 가능하다.

본 논문에서는 시맨틱 질의어로 RQL을 사용한다. FORTH의 Institute of Computer Science에서 개발된 RQL은 NON-SFW(NON SELECT-FROM-WHERE)와 SFW 구조로 표현이 가능하다. SFW 질의의 경우에, SELECT 절에서는 결과 값으로 반환되어야 하는 변수를 기술하며 FROM 절에서는 경로식(path expression)을 기술한다. 이때에 RDF/RDFS 그래프 상에 있는 경로를 따라 패

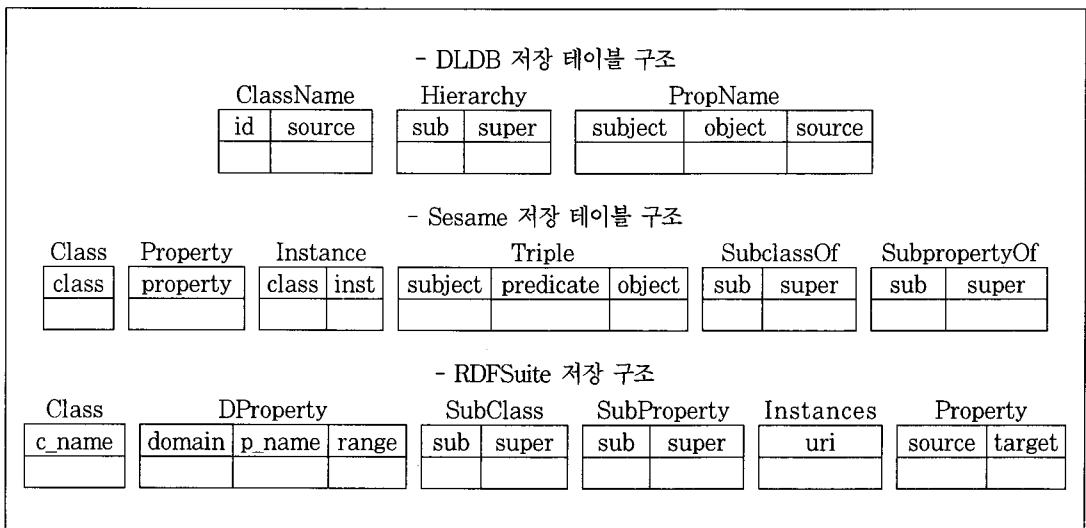


그림 1. 관련 연구의 저장 스키마  
Fig. 1. Storage Schema of Related Works

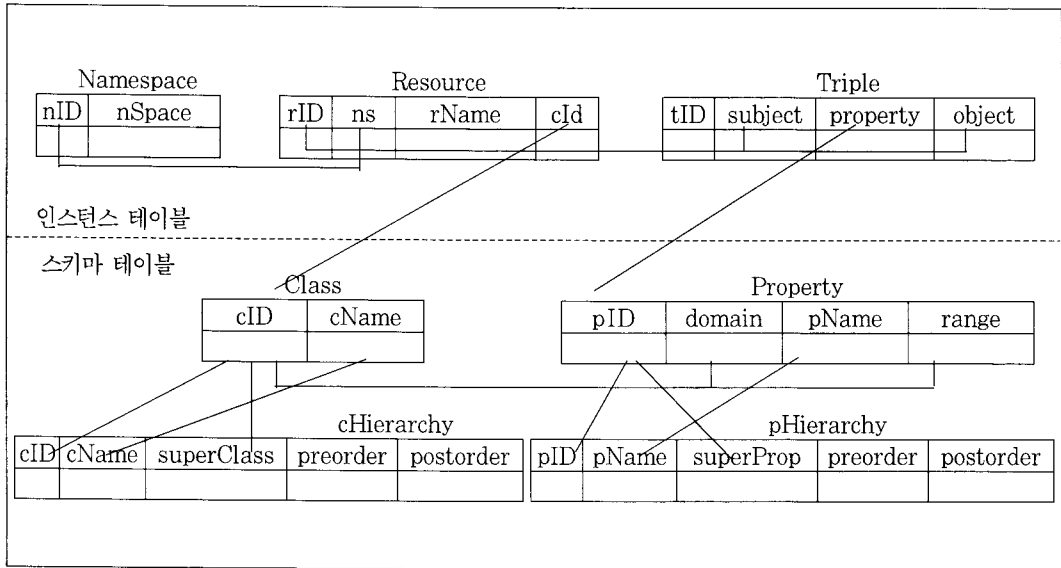


그림 2. RDF/RDFS 저장을 위한 데이터베이스 스키마  
Fig. 2. Database Schema for storing RDF/RDFS

턴이 대응된다. WHERE 절에서는 조건식이 기술된다.

지금까지 개발된 시맨틱 정보 관리 시스템으로는 DLDB[7], Sesame[8], Jena[9], RDFSuite[12] 등이 있다. Lehigh University의 Semantic Web & Agent Technology Lab에서 개발한 DLDB[7]는 RDBMS를 사용하여 온톨로지 데이터를 저장한 최초의 온톨로지 관리 시스템이다. DAML+OIL을 지원하기 위해 고안되었으며 DBMS로는 MS-Access를 사용한다. DLDB는 그림 1에서와 같이 문서상에 존재하는 클래스나 프로퍼티 각각에 대해 테이블을 생성하는 저장 스키마를 가진다. 클래스 테이블에는 해당 클래스의 인스턴스를 저장하고 각 인스턴스가 가지는 프로퍼티 값에 대한 정보들은 해당 프로퍼티 테이블을 사용하여 저장한다. 또한 클래스나 프로퍼티에 대한 계층 정보를 검색하기 위하여 클래스/프로퍼티의 이름(SUB)과 그것의 상위 클래스/프로퍼티 이름(SUPER)을 저장하는 테이블을 별도로 유지한다. 특정 클래스의 하위 클래스를 검색하는 방법은 그 클래스 이름을 상위 애트리뷰트(SUPER) 값으로 갖는 튜플을 찾아야 한다. 그리고 그 튜플의 SUB 애트리뷰트를 다시 상위 애트리뷰트 값으로 갖는 튜플을 찾는 작업을 더 이상의 하위 정보가 존재하지 않을 때까지 반복한다. 이 검색 방법은 계층 구조가 깊어질수록 하위 구조를 찾아가는 루틴이 반복되어 질의 처리의 성능이 떨어진다. 또한, 선언된 모든 클래스와 프로퍼티 각각에 대해서 테이블을 생성하는 것은 한 테이블 내에서 원하는 데이터를 찾기 위한 탐색 시간을 줄일 수 있으나 테이블의 수가

불필요하게 많아짐에 따라 테이블 사이의 많은 조인으로 인한 오버헤드를 발생시켜 질의 처리의 성능을 저하시키는 또 다른 요인으로 작용한다.

Sesame[8]은 유럽의 IST 프로젝트인 On-To- Knowledge의 일부로 개발되었으며 RDF와 RDFS로 기술된 문서를 저장 관리하는 데이터베이스 시스템이다. 온톨로지 질의어로는 RQL을 사용한다. 또한 BOR[10]패키지를 설치함으로써 OWL 및 DAML+OIL을 저장 및 관리할 수 있도록 하고 있다. DBMS로는 MySQL, PostgreSQL, Oracle 9i를 사용할 수 있으며 웹 인터페이스를 통해서 쉽게 온톨로지 문서를 저장 및 삭제할 수 있도록 한다. Sesame에서는 문서상에 존재하는 클래스와 프로퍼티에 대한 정보를 저장하기 위해 그림 1과 같이 클래스 테이블과 프로퍼티 테이블을 사용한다. 또한 클래스의 인스턴스에 대한 정보를 저장하기 위해 하나의 테이블을 유지하며, 각 인스턴스가 가지는 프로퍼티 값에 대한 정보를 별도의 테이블에 저장한다. 또한, DLDB에서 처럼 클래스의 계층 정보를 위한 테이블과 프로퍼티의 계층 정보를 위한 테이블을 각각 유지하는데 계층 구조가 깊어질수록 탐색 시간이 길어지는 문제점이 있다.

RDFSuite[12]는 ICS-FORTH에서 개발한 RDF 저장 시스템으로서 RDF 데이터를 객체 관계형 데이터베이스 시스템에 저장하며 온톨로지 질의어로는 RQL을 사용한다. RDF 데이터가 저장될 때 RDF 스키마 데이터의 클래스나 프로퍼티의 계층 구조 및 속성의 정보를 뽑아서 별도의 XML 문서

로 저장한다. DBMS로는 eXcelon XIS를 사용하며 데이터의 저장 구조는 그림 1과 같다.

[13]의 연구에서는 OWL 온톨로지 문서를 관계형 데이터 베이스로 저장하기위한 저장 스키마를 제안하며 온톨로지 질의어 RDQL을 사용하여 RDQL을 SQL로 변환하는 질의 변환 기법을 기술하고 있다. 그러나 RDQL은 RDF 데이터 관점의 웹 질의어이기 때문에 RDF에 대한 구조나 제약조건, 타입과 인스턴스 관계, 클래스나 속성의 계층관계를 반영시킬 수가 없기 때문에 스키마나 계층정보에 대한 데이터 검색이 제한되는 문제점이 있다.

본 논문에서는 기존의 연구의 단점을 보완한 새로운 온톨로지 관리 시스템을 제안하고자 하는데 연구의 목적은 다음과 같다. 첫째, 계층 구조가 깊어질 때에 질의 처리 성능이 저하되는 문제점을 해결하는 저장 스키마를 제안하여 질의 처리의 효율성을 증가시킨다. 둘째, 제안한 방법을 따라 구축된 데이터베이스에 대하여 RDF 데이터에 관한 질의뿐만 아니라 RDF 스키마에 대한 질의도 처리가 가능하도록 한다. 이를 위하여 RQL 시맨틱 질의를 SQL 관계형 질의로 변환하는 기법을 제안하고자 한다.

### III 온톨로지 저장 스키마

본 논문에서는 앞의 관련연구에서 살펴본 시맨틱 정보 관리 시스템들의 단점을 보완하고 온톨로지 문서들을 효율적으로 저장 및 검색할 수 있으며 계층 정보를 검색할 때 질의 처리의 효율성을 증가시킨 관계형 데이터베이스 스키마를 제안하고자 한다.

본 논문에서 제안한 저장구조는 다음을 중점적으로 고려하여 설계하였다. 첫째, 온톨로지 문서가 정보의 손실 없이 관계형 테이블로 저장되도록 한다. 둘째, 계층 정보를 효율적으로 검색하여 질의 처리의 성능을 향상시킨다.

온톨로지 문서를 관계형 데이터베이스로 저장하여 질의를 처리하는 기존의 연구는 계층정보를 위하여 별도로 XML 문서를 생성하여 XPath로 계층정보를 처리하는 방법[12]을 사용하거나 스키마에 관련된 의미적 정보의 처리를 배제한 상태에서 트리플 모델에 기반한 RDF 데이터 위주의 질의 처리를 다른 방법[13]이었다. 또한, DLDB나 Sesame에서의 계층 정보 추출 방법은 계층 구조가 깊어질수록 하위 구조를 찾아가는 루틴이 계속 반복되어 비효율적인 문제점이 있다. 따라서 본 논문에서는 RDF 문서의 트리플 데이터뿐만 아니라 스키마 정보의 처리가 가능하면서도 계층구조가 깊을지라도 효율적으로 검색이 가능한 온톨로지 저장 및 검색기법을 제안한다.

#### 1. 저장 테이블의 설계

그림 2는 온톨로지 문서를 효율적으로 저장 및 검색하기 위하여 본 논문에서 제안한 데이터베이스 스키마로서 RDF/RDFS를 구성하는 주요 클래스와 프로퍼티의 내용이 반영 및 저장되도록 설계하였다. 특히, 본 논문에서 제안한 cHierarchy와 pHierarchy 테이블은 클래스와 프로퍼티의 계층정보를 전위값(preorder)과 후위값(postorder)을 통하여 효율적으로 검색할 수 있게 해주어 질의 처리 속도를 향상시킨다.

제안한 데이터베이스 스키마에서는 온톨로지 문서를 저장하기 위해 그림 2과 같이 7개의 테이블을 사용하는데 각각의 테이블들에 대한 설명은 다음과 같다.

- Namespace: 네임스페이스를 관리하기 위한 테이블로서 온톨로지 일관성을 보장하는 목적으로 사용된다.
- Resource : 트리플 모델의 자원을 저장하는 테이블이다. 자원은 URI 값으로 표현되어지며 RDF 구문의 주어(subject)와 목적어(object)에 해당하는 부분이다. cid는 해당 자원이 속한 클래스의 타입을 나타낸다.
- Triple : RDF 구문의 트리플 모델을 저장하는 테이블이다. subject에는 RDF 구문의 주어부에 해당하는 리소스 아이디(rid)를 저장하며 pName에는 해당 프로퍼티를 저장한다. object는 프로퍼티의 값에 해당하며 다른 리소스나 문자값이 저장될 수 있다.
- Class : 온톨로지 문서에 정의된 클래스의 이름을 저장하고 있는 테이블이다. 스키마 관련 질의를 처리할 때에 사용된다.
- Property : 프로퍼티의 이름과 정의역, 공역에 관한 정보를 저장하고 있는 테이블이다.
- cHierarchy : 클래스의 계층 정보를 저장하고 있는 테이블이다. superClass는 상위 클래스를 나타내주며 preorder와 postorder는 계층정보를 나타내주는 필드로서 전위, 후위 정보가 저장된다. 계층 정보 검색에 대한 상세한 내용은 다음의 2절에서 다룬다.
- pHierarchy : 프로퍼티의 계층 정보를 저장하고 있는 테이블이다. cHierarchy와 마찬가지로 preorder와 postorder는 계층정보를 나타내주는 필드이다.

Namespace, Resource, Triple 테이블은 온톨로지 문서의 내용 인스턴스와 관련된 테이블이며 Class, Property, cHierarchy, pHierarchy는 온톨로지의 구조 스키마와 관련된 테이블이다.

## 2. 계층 정보의 저장

본 논문에서는 계층정보의 추출 시에 계층 구조가 깊어질 수록 하위 구조를 찾아가는 루틴이 계속 반복되어 효율성이 저하되는 문제를 개선하기 위하여 Dietz's Numbering[11] 기법을 이용하였다.

온톨로지 문서에 선언된 클래스/프로퍼티에 대한 계층 정보를 나타내는 테이블인 cHierarchy/ pHierarchy를 제안하였으며 이 테이블에 preorder와 postorder 필드를 두어 조상/자손(ancestor/descendant)의 관계를 나타내도록 하였다.

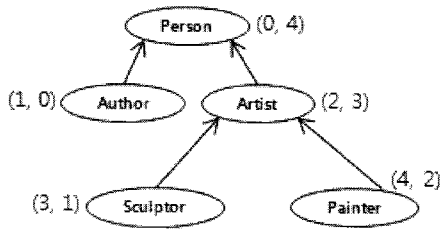


그림 3. 클래스의 계층 구조  
Fig. 3. Hierarchy of Class

표 1. 클래스 계층정보를 위한 테이블  
Table 1. Table for Class Hierarchy Information

cHierarchy				
cID	cName	superClass	preorder	postorder
c1	Person	Person	0	4
c2	Author	Person	1	0
c3	Artist	Person	2	3
c4	Sculptor	Person	3	1
c5	Painter	Person	4	2
c6	Artifact	Artifact	0	3
c7	Book	Artifact	1	0
c8	Painting	Artifact	2	1
c9	Sculpture	Artifact	3	2

그림 3은 문화 포털을 예로 들어 클래스 간의 계층구조를 나타내고 있다. 여기서 각 노드에 대해 전위(Preorder)와 후위(Postorder) 숫자의 순서쌍을 대응시켜 계층 정보로 활용한다. 전위의 넘버링 방법은 상위 레벨에서 하위 레벨 순서로, 좌측에서 우측 순서로 번호가 부여되며 후위의 넘버링은 좌측

하위 노드에서 시작하여 좌에서 우로 하위에서 상위 레벨 순으로 번호가 부여된다. 표 1의 cHierarchy 테이블에서 c1~c5 레코드는 그림 3의 클래스 계층 구조를 저장한 내용이다. superClass 필드에는 자신의 최상위 클래스를 저장하고 preorder와 postorder 필드에는 전위와 후위 값을 각각 저장한다. cHierarchy 테이블을 사용하여 하위 클래스를 검색하는 방법은 superClass 값이 자신의 superClass 값과 같으며 preorder 값이 자신의 전위 값보다 크고 postorder 값이 자신의 후위 값보다 작은 클래스를 모두 찾으면 된다. 예를 들어, Artist 클래스의 하위 클래스를 검색하는 질의는 다음과 같은 SQL 질의로 표현된다.

```

SELECT cName
FROM cHierarchy
WHERE superClass='Person' And
preorder>=2 And postorder<=3;
    
```

기존의 연구에서는 계층정보를 저장하는 테이블에 전위와 후위 정보가 없이 단순히 하위클래스/프로퍼티와 상위클래스/프로퍼티 정보만을 유지하고 있으므로 계층구조를 반영해야 하는 질의를 SQL과 같은 관계형 질의어로 표현하여 처리하기에 어려운 점이 있었다. 그러나, 본 논문에서 제안한 cHierarchy/pHierarchy 테이블을 이용하면 간단하게 계층 정보를 검색하여 활용할 수 있다.

## IV. RQL 질의 변환

### 1. 질의 처리 과정

본 논문에서는 질의 처리를 위하여 RQL을 사용한다. RQL은 인스턴스 데이터뿐만 아니라 스키마 정보까지도 처리가 가능한 시맨틱 질의어이다.

RQL 질의의 처리 과정은 그림 4와 같다. 사용자가 입력한 RQL 질의는 RQL 파서(Parser)를 통하여 데이터 질의, 클래스 질의, 프로퍼티 질의, SFW(Select From Where) 질의로 분류된다. 분류된 질의는 각각에 해당하는 변환기를 통과하면서 SQL 관계형 질의로 변환되어 본 논문의 3장에서 설계한 관계형 데이터베이스에 저장되어 있는 정보를 추출하여 질의 결과를 생성한다.

파싱 과정에 하나의 질의는 여러 개의 서브 질의로 나뉘어 질 수 있으며 나누어진 서브 질의들을 각각 SQL으로 전환한

후에 조인(join) 연산을 수행하여 최종 질의 결과를 얻게 된다.

## 2. RQL의 SQL 변환

RQL 질의는 데이터 질의, 클래스 질의, 프로퍼티 질의, SFW(Select From Where) 질의로 분류될 수 있다. 데이터 질의, 클래스 질의, 프로퍼티 질의는 검색 대상이 각각 데이터, 클래스, 프로퍼티이며 Select-From- Where 형식을 취하지 않는 질의이다. 본 절에서는 RQL 질의를 SQL 질의로 변환하는 방법을 Non-SFW((Non Select From Where)와 SFW(Select From Where) 질의로 나누어서 기술하고자 한다.

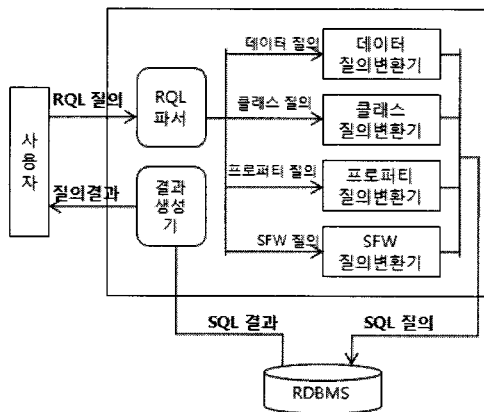


그림 4. RQL 질의처리기  
Fig. 4. RQL Query Processor

### 2.1 Non-SFW 질의 변환

#### ▶ 데이터 질의

데이터 질의는 질의의 결과로 리소스 데이터가 검색되는 질의이다. 예를 들어, 그림 5의 문화 포탈 예제에서 클래스 Artist에 속한 리소스를 구하는 질의를 RQL로 표현하면 다음의 표에서와 같이 해당 클래스의 이름을 그대로 쓰면 된다. 질의 처리 결과로는 &r1과 &r5이 검색된다. 이때에 하위 클래스에 속한 인스턴스도 결과에 포함된다. 데이터 질의는 클래스 이름이 주어지고 해당 리소스를 구하는 질의이며 하위 클래스 정보도 필요하므로 그림 2의 테이블 중에서 cHierarchy 와 Resource 테이블간의 조인 연산으로 변환시킬 수 있다.

RQL	Artist
SQL 변환	<pre>SELECT Resource.rName FROM cHierarchy JOIN Resource ON cHierarchy.cID = Resource.cID WHERE cHierarchy.superclass='person' And cHierarchy.preorder&gt;=2 And cHierarchy.postorder&lt;=3</pre>

#### ▶ 클래스 질의

클래스 질의는 질의의 결과로 클래스가 검색되는 질의로서 TypeOf, SuperClassOf, SubClassOf, Domain 등이 있다.

TypeOf는 리소스를 주고 그것의 클래스 타입을 구하는 질의로서 그림 2의 테이블에서 Class와 Resource의 조인으로 변환될 수 있다.

RQL	typeof(www.culture.net#picasso132)
SQL 변환	<pre>SELECT class.cName FROM class INNER JOIN Resource ON class.cID = Resource.cID WHERE rName='www.culture.net#picasso132';</pre>

SuperClassOf는 주어진 클래스의 상위 클래스를 구하는 질의이다. 본 논문에서 제안한 cHierarchy 테이블을 이용하여 다음과 같이 간단히 구할 수 있다.

RQL	superclassof(painter)
SQL 변환	<pre>SELECT cName FROM cHierarchy WHERE superclass='person' And preorder&lt;3 And postorder&gt;1;</pre>

#### ▶ 프로퍼티 질의

프로퍼티 질의는 질의의 결과로 프로퍼티가 검색되는 질의로서 SuperPropertyOf, SubPropertyOf가 있다.

SubPropertyOf는 주어진 프로퍼티 하위 프로퍼티를 구하는 질의이다.

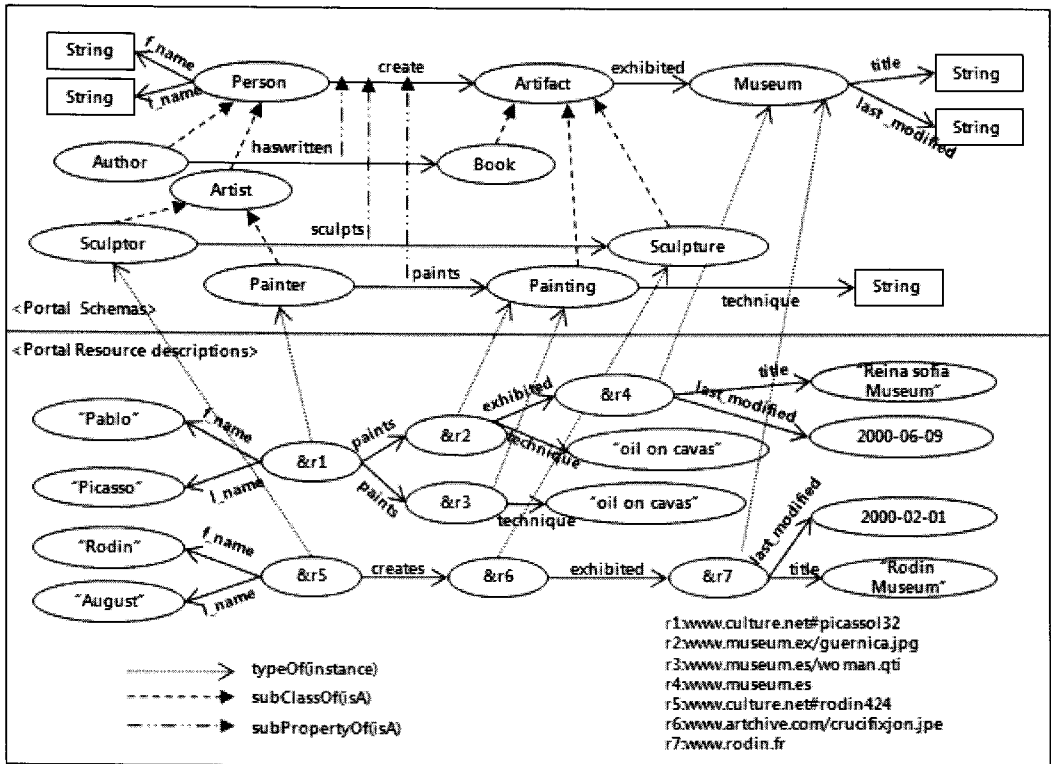


그림 5. 문화 포탈에 대한 RDF 리소스 기술  
Fig. 5. RDF resource descriptions for Culture Portal

RQL	SubPropertyOf(create)
SQL 변환	<pre>SELECT pName FROM pHierarchy WHERE superprop='create' And preorder&gt;0 And postorder&lt;3;</pre>

## 2.2 SFW 질의 변환

RQL의 SFW 질의는 SELECT-FROM-WHERE 절로 구성되어 있다. SELECT 절에서는 결과 값으로 반환되어야 하는 변수를 기술하며 FROM 절에서는 경로식(path expression)을 기술한다. 경로식은 트리플 구조를 따르며 이때 RDF/RDFS 그래프 상에 있는 경로를 따라 패턴이 대응된다. WHERE 절에는 조건식이 있는 경우 여기에 기술된다. 예를 들어, Painter 클래스를 도메인으로 하는 프로퍼티의 주어(subject)와 프로퍼티값(object)를 구하는 RQL 질의는 다음과 같다.

```
SELECT X, Y .....(질의1)
WHERE {X:Painter}@p{Y}
```

여기서 X, Y는 구하고자 하는 데이터 변수이고 WHERE 절에 있는 {X:Painter}@p{Y}는 경로식이며 트리플 구조를 하고 있다. @p는 프로퍼티 변수이며 X는 프로퍼티의 주어(트리플 모델의 Subject)인데 타입이 Painter이며 Y는 프로퍼티의 값(object)을 나타낸다.

SFW 질의를 SQL로 변환하는 방법은 다음과 같다. 첫째, 질의를 SELECT, FROM, WHERE 부분으로 나눈다. 둘째, FROM 절의 경로식을 단순 경로식으로 분리한다. 셋째, 세분화된 각각의 경로식을 SQL 질의로 변환한다. 이때 WHERE 절의 조건식이 SQL의 조건절에 포함된다. 넷째, 생성된 SQL 질의를 조인한다. 이때 SELECT 절은 RQL 경우와 같은 내용이 대응된다.

위의 (질의1)를 예로 들어 SQL로 변환해 보자. 우선, 경로식을 분리하여 얻은 {X:Painter}@p를 SQL로 변환하면 다음과 같이 도메인이 painter인 프로퍼티를 구하는 질의문이 된다.

```
SELECT pName
FROM property          .....(질의2)
WHERE Domain='painter';
```

(질의2)의 결과는 paints이고 이것을 (질의1)에 대입하면 WHERE 절이 {X}paints{Y}로 변형된다. 즉, 프로퍼티가 paints인 트리플 문에서 주어(object)와 프로퍼티값(object)를 구하는 식이 되며 이것을 SQL로 변환하면 다음과 같다.

```
SELECT subject, object
FROM Triple          .....(질의3)
WHERE property='paints';
```

마지막 단계로 (질의2)과 (질의3)를 조인하여 최종 SQL 질의로 변환된다.

```
SELECT subject, object
FROM property INNER JOIN triple ON
      property.pName=triple.property
WHERE property.domain='painter';
```

SFW 질의는 SFW 스키마 질의와 SFW 데이터 질의로 분류할 수 있다.

▶ SFW 스키마 질의

SFW 스키마 질의는 클래스, 프로퍼티, 계층 정보, 정의역, 공역과 같은 스키마 정보와 관련이 있는 SFW 질의이다. 다음의 예제 질의는 클래스 painter의 property와 range를 구하는 질의이다. 이 질의는 우선 painter 클래스의 상위 클래스를 추출한 후에 이들에 대한 property와 range도 결과값에 포함시켜야한다. 따라서, 계층 정보를 나타내는 cHierarchy 테이블과 프로퍼티의 정의역과 공역을 나타내주는 property 테이블을 조인하여 SQL 질의로 변환된다.

RQL	SELECT @P, range(@P) FROM {\$C}@P WHERE \$C = 'painter'
SQL 변환	SELECT property.pname, property.range FROM cHierarchy JOIN property ON cHierarchy.cName=property.domain WHERE cHierarchy.superclass='person' And cHierarchy.preorder<=3 And cHierarchy.postorder >= 1

▶ SFW 데이터 질의

SFW 데이터 질의는 SFW 형식을 갖추고 있으면서 질의의 결과로 리소스 데이터가 검색되는 질의이다. 2000년 이후에 고쳐진 museum 리소스와 날짜를 구하는 RQL 질의는 다음과 같다.

```
SELECT X, Y
FROM museum(X).last_modified{Y}
WHERE Y >=2000-01-01
```

SQL로 변환하는 과정은 우선, 경로식을 분리하여 얻은 museum(X)를 SQL로 변환한다. 이때에, museum(X)는 클래스 museum을 타입으로 갖는 리소스를 추출하는 것을 의미하며 이것을 SQL 문으로 구현하기 위해서는 그림 2의 본 논문에서 제안한 저장 모델의 class와 Resource 테이블을 사용해야 한다.

```
SELECT Resource.rID
FROM class, Resource
WHERE class.cName= 'museum' And
class.cID=resource.cID
```

다음 단계로 FROM 절의 나머지 경로식 last\_modified{Y}을 SQL로 변환한다. 이 과정에서 WHERE 절의 Y >=2000-01-01' 조건식이 반영된다.

```
SELECT subject, object
FROM triple
WHERE property='last_modified' And
object>='2000-01-01'
```

마지막 단계로 위의 두 SQL 질의를 조인하여 얻은 최종결과를 다음과 같다.

SQL 변환	SELECT subject, object FROM class, Resource, triple WHERE class.cName='museum' And triple.object>='2000-01-01' And triple.property='last_modified' And class.cID=resource.cID And triple.subject=Resource.rID;
--------	--



## V. 구현 및 비교 평가

### 1. 구현

본 논문에서는 제안한 저장 스키마를 사용하여 RDF/RDFS 문서를 데이터베이스로 구축하였다. 구축한 데이터베이스에 관한 시맨틱 정보를 검색하기 위하여 RQL 질의를 SQL로 변환하는 질의 처리 변환 기법을 MS-ACCESS에 적용하여 구현하였다. 구현에 사용된 데이터는 그림 5의 문화 포털 RDF 및 RDF 스키마의 예제를 사용하였다. 다음의 그림 6은 문화 포털 RDF/RDFS 문서를 관계형 데이터베이스로 구축한 내용이다.

class	cID	cName
c1	author	
C10	museum	
c2	person	
c3	artist	
c4	painter	

property	PID	domain	pName	range
p1	person	create	artifact	
p10	painting	technique	String	
p2	author	haswritten	book	
p3	painter	paints	painting	
p4	sculptor	sculpts	sculpture	

Resource	rID	rName	cID
&r1	www.culture.net#picasso132		c4
&r2	www.museum.ex/guernica.jpg		c8
&r3	www.museum.es/woman.qti		c8
&r4	www.museum.es		c10
&r5	www.culture.net#rodin424		c5

triple	tID	subject	property	object
t1		&r1	paints	&r2
t10		&r4	title	Reina Sofia Museum
t11		&r7	last_modified	2000-02-01
t12		&r7	title	Rodin Museum
t13		&r1	f_name	Pablo
t14		&r1	L_name	Picasso
t15		&r5	f_name	Rodin
t16		&r5	L_name	August
t2		&r1	paints	&r3

cHierarchy	cID	cName	superclass	preorder	postorder
c1	author	person		1	0
C10	museum	museum		0	0
c2	person	person		0	4
c3	artist	person		2	3
c4	painter	person		3	1
c5	sculptor	person		4	2
c6	artifact	artifact		0	3

pHierarchy	PID	pName	superprop	preorder	postorder
p1	create	create		0	3
p10	technique	technique		0	0
p2	haswritten	create		1	0
p3	paints	create		2	1
p4	sculpts	create		3	2
p5	f_name	f_name		0	0
p6	L_name	L_name		0	0
p7	exhibited	exhibited		0	0
p8	title	title		0	0
p9	last_modified	last_modified		0	0

그림 6. RDF/RDFS의 관계형 DB 저장  
Fig. 6. Relational DB for RDF/DRFS

RQL 질의들이 본 논문에서 제안된 저장 스키마를 사용하여 구축된 데이터베이스 상에서 SQL로 변환되어 질의를 처리한 결과는 다음과 같다.

Q1 : 'www.culture.net#picasso132' 리소스의 클래스 타입을 구하시오.

RQL : typeof(www.culture.net#picasso132)  
SQL 변환 및 실행 결과 :

```
SELECT class.cName
FROM Resource INNER JOIN class ON Resource.cID=class.cID
WHERE rName='www.culture.net#picasso132'
```

Num	cName
*	painter

Q2 : create 프로퍼티의 도메인을 구하시오.

RQL : Domain(create)  
SQL 변환 및 실행 결과 :

```
SELECT property.domain
FROM pHierarchy INNER JOIN property
ON pHierarchy.pID=property.pID
WHERE pHierarchy.superprop='create' And
pHierarchy.preorder <= 0 And pHierarchy.postorder <= 3;
```

domain
person
author
painter
sculptor

Q3 : painter 클래스의 range와 property를 구하시오.

RQL : SELECT @P, range(@P)  
FROM (\$C)@P  
WHERE \$C = 'painter'  
SQL 변환 및 실행 결과 :

```
FROM cHierarchy INNER JOIN property
ON cHierarchy.cName=property.domain
WHERE cHierarchy.superclass='person' And
cHierarchy.preorder <= 3 And cHierarchy.postorder >= 1;
```

pName	range
f_name	String
L_name	String
create	artifact
paints	painting

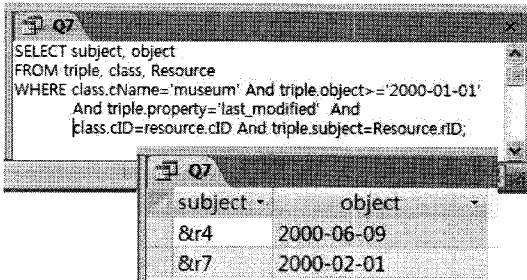
표 2. 타 시스템과 비교  
Table 2. Comparison of Other systems

	DLDB	Sesame	RDFSuite	[13] 연구	본 연구
온톨로지 질의어	KIF, JAVA	RQL	RQL	RDQL	RQL
DBMS	관계형 (MS-ACCESS)	관계형 (Oracle 9i)	객체관계형 (eXcelon XIS)	관계형 (MySQL)	관계형 (MS-ACCESS)
계층구조 검색시 루틴 반복	반복	반복	XML 정보 활용	계층구조검색불 가	반복안함
스키마 질의	가능	가능	가능	불가능	가능

Q4 : 2000년 이후에 고쳐진 museum 리소스와 날짜를  
구하시오.

```
RQL : SELECT X, Y
      FROM museum(X).last_modified{Y}
      WHERE Y >=2000-01-01
```

SQL 변환 및 실행 결과 :



2. 비교 평가

본 연구를 타 시스템과 기능면에서 분석한 결과는 표 2과 같다. 본 연구는 현재 상용되고 있는 데이터베이스 시스템 중에서 가장 안정적이면서도 널리 사용되고 있는 관계형 데이터베이스를 사용하고 있으며 특히 계층정보 검색 시에 DLDB 나 Sesame의 경우에는 계층 구조가 깊어질수록 하위 구조를 찾아가는 루틴이 반복되며, RDFSuite의 경우에는 별도의 XML 문서를 통하여 계층 정보를 검색하는 데에 비하여 본 연구에서는 제안한 cHierarchy/pHierarchy 테이블을 사용하여 한번의 SQL 문으로 계층 정보를 검색할 수 있다(III.2 참조). 또한 [13] 연구의 경우에는 트리플로 표현된 RDF 데이터에 관한 질의만 지원되는 데에 비하여 본 연구에서는 클래스 타입이나 계층 구조와 같은 스키마에 관한 시맨틱 질의도 처리가 가능하다.

VI. 결론

정확한 정보를 검색하기위한 목적으로 개발된 시맨틱 웹이 지속적으로 발전하기 위해서는 온톨로지 데이터를 효과적으로 저장 및 검색할 수 있는 시스템이 필요하다.

본 논문에서는 RDF/RDFS, OWL과 같은 온톨로지 언어를 관계형 데이터베이스에 저장 및 검색하기 위한 스키마 모델을 제안하였다. 제안한 데이터베이스 스키마에서는 계층정보를 위한 테이블인 cHierarchy과 pcHierarchy을 제안하여 클래스와 프로퍼티의 계층정보를 효율적으로 검색할 수 있도록 설계하여 질의 처리의 효율성을 증가시켰다. 또한, 시맨틱 웹 질의 언어인 RQL을 질의 유형별로 나누어 관계형 SQL 질의로 변환시키고 이를 구현하였다. 구현 결과에서 볼 수 있듯이 트리플 구조의 데이터 질의 뿐만 아니라 클래스 타입, 도메인, 클래스나 프로퍼티의 계층 관계에 관한 시맨틱 질의도 SQL 구문으로 간단하게 처리됨을 알 수 있다.

본 논문에서는 온톨로지를 위한 관계형 데이터베이스 스키마를 구축하고 그에 적절한 질의 처리 기법을 제안 및 구현하고 타 시스템과 기능적인 비교를 한 것을 연구 범위로 하고 있지만 추후에는 타 시스템과 질의 처리의 성능을 비교 분석하는 연구가 진행되어야 한다.

## 참고문헌

- [1] Dave beckett and et al., "RDF/XML Syntax Specification," W3C Recommendation, 2004.
- [2] Dan Brickely, R. V. Guha and et al., "RDF Vocabulary Description 1.0 RDF Schema," W3C Recommendation 10 Feb. 2004.
- [3] Deborah L. McGuinness and et al., "DAML+OIL: An Ontology Language for The Semantic Web," IEEE Internet Systems, pp. 72-80, 2002.
- [4] Deborah L. McGuinness, Frank van Harmelen, "OWL Web Ontology Language Overview," W3C Recommendation, Feb. 2004.
- [5] Gregory Karvounarakis and et al., "RQL: A Declarative Query Language for RDF," 11<sup>th</sup> WWW Conference, pp. 592-603. 2002.
- [6] Richard Fkies, Patrick Hayes, and Ian Horrocks, "OWL-QL : A Language for Deductive Query Answering on the Semantic Web," Web Semantics, Vol2, pp. 19-29, December 2004.
- [7] Z. X. Pan and J. Heflin, "DLDB: Extending Relational Databases to Support Semantic Web Queries," Workshop on Practical and Scalable Web Systems, pp. 109-113, 2003.
- [8] J. Broekstra, A. Kampman, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," Lecture Notes in Computer Science, Vol.2342, pp. 54-68, 2002.
- [9] K. Wilkinson, C. Sayers, H. Kuno, "Efficient RDF Storage and Retrieval in Jena2," SWAD, pp. 131-150, 2003.
- [10] S. Alexaki, and et al., "The RDFSuite: Managing Voluminous RDF Description Bases," Technical Report, ICS\_FORTH, 2001.
- [11] Kiri Simov, Stanislav Jordanov, "BOR: a Pragmatic DAML +OIL Reasoner," On-To-Knowledge project, 2002.
- [12] 김학수, 손현진, "OWL 인식 관계형 모델에서 SQL 기반의 시맨틱 질의 처리," 정보과학회논문지, 제35권, 제1호, 44-53쪽, 2008년 2월.
- [13] Paul F. Dietz, "Maintaining Order in A Linked List," Proceedings of the 14<sup>th</sup> ACM Symposium on Theory of Computing, pp. 122-127, 1982.
- [14] 김연희, 임해철, "시맨틱 웹 데이터의 키워드 질의 처리를 위한 인덱싱 및 저장 기법," 한국컴퓨터정보학회논문지, 제12권, 제5호, 93-107쪽, 2007년 11월.
- [15] 윤보현, 서창호, "시맨틱웹을 위한 효율적인 온톨로지 객체 모델," 한국컴퓨터정보학회논문지, 제11권, 제2호, 7-13쪽, 2006년 5월.

## 저자소개



### 이 순 미

1984: 이화여자대학교수학과(학사)  
 1986: 이화여자대학교 대학원 수학과  
 전산전공(석사)  
 1997: 홍익대학교 대학원 전산학과  
 (박사)  
 현재: 경인여자대학 정보미디어학부  
 부교수  
 관심분야: 데이터베이스, 분산시스템,  
 시맨틱 웹