

시맨틱 웹의 효율적 검색을 지원하는 저장구조의 요소기술 설계 (Designing Requisite Techniques of Storage Structuresupporting Efficient Retrieval in Semantic Web)

신판섭(Panseop Shin)¹⁾

요 약

차세대 웹 환경을 위한 시맨틱 웹에 대한 관심이 날로 증대되고 있다. 이와 더불어 시맨틱 웹에서 자원의 의미적인 관계를 표현하기 위한 언어들에 대한 연구가 활발히 진행 중이다. 특히 이러한 연구의 시작점으로 RDF, DAML+OIL같은 여러 온톨로지 언어들에 등장하였지만 자원의 특성에 대한 기술이나 그들 간의 관계에 대한 정의를 명확하게 표현하기에는 미흡한 점이 많다. 그러나 최근에 제안된 OWL은 RDF나 RDF 스키마를 확장하여 보다 의미있는 자원의 관계를 정의할 수 있다. 본 논문에서는 OWL로 표현된 문서를 저장하기 위한 구조를 관계형 데이터베이스를 기반으로 제안한다. 본 논문에서 제안한 저장 구조는 기존의 RDF와 RDF 스키마를 위한 저장 구조를 확장하여, OWL을 통해 제공되는 클래스나 속성간의 동일성, 또는 이질성, 여러 클래스의 집합 관계 등의 추가적 기능을 지원하는데 목적을 두고 있다. 또한 제안한 저장구조를 이용한 OWL 문서의 질의 형태를 분석하고 추가 질의 형태를 제안하고, OWL 문서의 저장 모듈과 사용자 인터페이스, 질의 처리 모듈로 구성된 OWL 저장 시스템을 설계하고 구현한다.

ABSTRACT

Semantic Web is getting popular to next web environment. Additionally, ontology language research is also activating to represent semantic relation of resource in semantic web. Specially, Ontology language as RDF and DAML+OIL appear on start point of research. But Ontology Language limited to describing characters of resource and to making a clear definition of relation of resource. So W3C suggest OWL at the next standard language for describing resource. OWL supply the lack of representation for RDF and RDF Schema. In this paper, we make Ontology to implement Online Retrieval System using OWL and propose the structure of storing Ontology document at the RDB. The structure support characters of OWL that are equivalent relationship, heterogeneous relationship, inverse relationship, union relationship and one of relationship between classes or properties. In this paper, we classify the extended elements for OWL from RDF Schema. And we propose the method of storing OWL using RDB for interoperability with many applications based on RDB. Finally, implement the storage and retrieval system based on OWL to provide advanced search function.

논문접수 : 2006. 5. 10.

심사완료 : 2006. 5. 29.

1) 정회원 : 대전대학교 컴퓨터공학과

이 논문은 2005년도 대전대학교 학술연구비 지원으로 연구되었음

1. 서론

·1RDF는 웹 상에 존재하는 자원들의 메타데이터를 기술하기 위한 언어이다. RDF는 XML 문법을 따르고 있으며, 기본적으로 자원-속성-값의 기본 구조를 이용하여 자원의 메타데이터를 기술한다[3]. RDF는 XML에서 문제점으로 제시되었던 태그 사용에 있어서의 호환성이 여전히 존재한다. 예를 들면, 동음이의의 태그 관계나 이음동의의 태그 관계를 정확히 표현하지 못한다. 그래서 RDF에서는 RDF 스키마를 사용하여 이러한 문제를 해결하고 있다. RDF 스키마는 기술 대상이 되는 자원의 클래스 타입과 그 자원을 표현하기 위해 필요한 속성들을 정의하고 클래스와 클래스간의 관계, 속성과 속성간의 의미적 관계를 명확히 정의한다[2]. 그러나, RDF 스키마는 클래스간 또는 속성간의 동일 관계, 이질 관계, 역관계의 정의 그리고 union, intersection 등과 같이 클래스간의 관계 정의나 속성에 대한 제약을 표현하는 데는 한계가 있다. 따라서 W3C에서는 RDF와 RDF 스키마의 부족한 모델링 요소를 보완하여 OWL을 제안하였다[1]. OWL의 풍부한 표현력 때문에 많은 사람들이 OWL에 관심을 가지게 되었고 이와 관련한 연구가 증가하는 추세이다. 하지만 대부분 RDF와 RDF 스키마를 위한 저장 및 질의시스템에 대한 연구가 주를 이룬 반면, OWL 문서의 저장과 관련된 연구가 미진한 상태이므로 OWL의 특성을 지원하는 저장 구조가 필요하다.

따라서 본 논문에서는 첫째, RDF 스키마에 비해 확장된 OWL의 기능을 분류하였다. 둘째, OWL의 확장된 특성을 지원할 수 있는 관계형 데이터베이스 기반의 저장구조를 제안한다. 셋째, RDF와 RDF 스키마에서 확장된 OWL의 특성에 관한 질의가 가능하도록 기존 메타데이터의 질의 어인 RQL을 SQL로 변환하는 기법을 제안하였다. 마지막으로 본 논문에서는 온톨로지 기반의 저장 및 질의 시스템을 설계하고 이를 구현하였다.

본 논문의 구성은 다음과 같다.

2장에서는 OWL의 모태인 RDF 데이터 모델에 대해 소개하고 이를 이용한 저장 시스템인

RDFSuite와 Sesame 시스템에 대해 설명한다. 그리고 본 논문에서 온톨로지 구성 언어로 사용하고 있는 OWL에 대해 소개한다.

3장에서는 OWL이 클래스나 속성간의 관계 표현에 있어 RDF와 RDF 스키마에 비해 확장된 개념들에 대해 분류한다. 4장에서는 OWL로 작성된 온톨로지 문서를 실제 RDB에 저장 할 수 있는 저장 구조를 제안한다. 5장에서는 RDB에 저장된 데이터를 대상으로 질의 가능하도록 RQL을 SQL로 변환하는 기법을 제시한다. 6장에서는 설계 및 구현된 온톨로지 시스템을 보여준다. 7장에서는 결론을 맺는다.

2. 관련연구

2.1 RDF(Resource Description Framework)

기존의 웹 환경에서는 HTML을 이용하여 문서의 내용과 의미를 나타내는 시맨틱 정보를 표현하기가 어렵다. RDF는 W3C의 가장 기본적인 시맨틱 웹 언어로서 웹에 존재하는 자원에 대한 메타데이터를 기술하기 위한 언어이다. RDF는 모든 웹에 자원을 리소스로 표현하고, 각 리소스의 특성이나 다른 리소스와의 관계를 주어-동사-목적어의 서술문 형식으로 나타낸다.

2.2. 저장시스템 및 키워드 검색

본 절에서는 OWL의 이전 버전이라 할 수 있는 RDF 데이터 모델에 대해 살펴본 후, RDF의 저장 시스템인 RDFSuite과 Sesame 시스템에 대해 살펴본다. 또한 본 논문에서 온톨로지를 구현하는 언어로 사용하고 있는 OWL에 대하여 살펴본다.

2.2.1 RDF 데이터 모델

기존의 웹 환경에서는 HTML을 이용하여 문서의 내용과 의미를 나타내는 시맨틱 정보를 표현하기가 어려우며, 따라서 사람이 아닌 프로그램 또는 어플리케이션이 자동으로 문서로부터 의미를 추출하기가 어렵다. 시맨틱 웹은 메타데이터의 개념을 통하여 웹 문서에 시맨틱 정보를 덧붙이고 이를 이용하여 어플리케이션이 의미 정보를 자동으로

추출할 수 있는 정책이나 기법을 조성하는 것이다. 부수적으로 의미 정보의 자동 추출뿐 아니라 정보의 확장이나 공유 등도 가능하게 될 것이다[9].

시맨틱 웹의 궁극적인 목적은 웹에 있는 정보를 컴퓨터가 좀 더 이해할 수 있도록 도와주는 표준과 기술을 개발하여 시맨틱 검색, 데이터 통합, 네비게이션, 작업의 자동화 등을 지원하는 것이다.

2.2.2 RDF 저장 시스템

1) RDFSuit

ICS-FORTH의 RDFSuite 시스템은 RDF 문서의 파싱을 위한 파서인 VRP, RDF의 저장 구조인 RSSDB 그리고 객체-관계 데이터베이스에 기반한 RDF의 질의 처리기로 구성되어 있다. RDFSuite는 클래스와 속성 정보를 저장한 테이블들과 그들의 포함관계를 포함한 subClassOf와 subPropertyOf 테이블을 유지하고 각 속성마다 테이블을 만들어 관계있는 자원들을 연결함으로써 객체-관계 데이터베이스를 기반으로 저장 구조에 저장하고 질의 가능하도록 개발하였다[6].

RDFSuite 시스템은 RDF 데이터를 객체 관계형 데이터베이스 시스템에 저장하고 RQL 질의가 들어오면 이를 SQL3로 변환하여 결과를 가져온다. 이 시스템은 기반 데이터베이스 시스템으로 객체 관계형 데이터베이스를 사용하였으므로, 저장할 때 데이터베이스 스키마에 맞게 데이터를 구성하여 저장하여야 하고, 검색할 때에도 여러 테이블간의 조인 등의 오버헤드가 있다[6].

2) Sesame

On-to-Knowledge 프로젝트의 일부인 Sesame 시스템은 객체-관계 데이터베이스뿐만 아니라 관계형 데이터베이스에서도 저장 가능한 구조를 제공하고 있다. Sesame의 저장 구조는 RDFSuite와 거의 유사하다. 다만, 속성의 도메인과 레인지 테이블을 각각 따로 유지한다는 점이 차이점이다[5].

Sesame 시스템은 RDF와 RDF 스키마 정보를 위한 저장 시스템으로서, 어떠한 하부 저장 구조와 호환이 가능하도록 RAL(Repository Abstraction

Layer)라는 미들웨어와 RDF 처리를 위한 3가지 기능성 모듈인 RQL query 모듈, Admin 모듈, RDF Export 모듈로 구성되어 있다. Sesame 시스템은 대부분의 저장 구조와 호환이 가능하다는 장점을 가지고 있으나, 이를 위하여 모든 처리 과정은 하부의 저장 구조에 맞게 변환하는 과정이 반드시 필요하므로 이에 따른 오버헤드가 크다[5].

2.2.3 OWL

OWL은 의미 있는 검색을 지원하기 위하여 하나의 온톨로지 또는 다수의 분산형 온톨로지에 근거하여 웹 자원의 의미를 기술하는 온톨로지 언어이다. OWL은 클래스의 구성원들에 관한 사실과 구성원들 간의 관계를 기술하며, 이러한 클래스 정보를 모아 만들어진 온톨로지는 구문적으로 정의되지 않은 사실의 논리적 유추를 가능하게 한다.

OWL은 RDF의 트리플 형태로 표현되며, OWL은 표현력이 서로 다른 세 개의 하위 언어인 OWL Lite, OWL DL, OWL Full로 구성되어 있다[1].

1) OWL Lite

클래스 분류 계층과 간단한 제약 사항 표현을 필요로 하는 사용자들을 위한 언어이다. 예를 들면, OWL Lite는 관계차수 제약 사항의 표현을 지원하지 않지만, 차수의 값으로 0 또는 1만 사용할 수 있도록 제한한다. OWL Lite를 지원하는 도구를 제작하는 것은 다른 하위 언어를 지원하는 도구를 제작하는 것 보다 상대적으로 쉽다. OWL Lite는 유의어 사전이나 여타 분류 체계의 표현 언어를 빠르고 손쉽게 OWL화하기 위한 용도로 적합하다[1].

2) OWL DL

연산 완전성(Computational Completeness)과 결정가능성(Decidability)을 유지하면서 최대의 표현력을 활용하고자 하는 사용자에게 적합하다. 완전성은 모든 결론이 계산될 수 있다는 특성이고, 결정가능성은 모든 계산이 유한한 시간 안에 끝나는 특성이다. OWL DL은 OWL의 모든 어휘를 포함하고 있다. 그러나 어휘의 사용에 있어 정해진 제약 사항을 준수해야 한다.

3) OWL Full

계산학적인 어떤 보장 없이 최대의 표현력과 RDF의 유연한 문법을 모두 활용하고자 하는 사용자에게 적합하다. 예를 들면, OWL Full에서 클래스는 개체(Individual)의 집합인 동시에 그 자체가 하나의 개체가 될 수도 있다. OWL Full은 사전 정의된 (RDF 또는 OWL) 어휘의 의미를 확장하는 온톨로지를 작성하도록 허용한다. OWL Full의 모든 기능에 대하여 완전한 추론을 지원하는 추론 소프트웨어는 만들 수 없을 것으로 보인다[1].

OWL DL이나 OWL FULL은 표현력에 있어서 OWL Lite 보다 뛰어나지만 일반 사용자들이 잘 사용하지 않는 기능이 많기 때문에 본 논문에서는 OWL Lite에 초점을 맞추고, OWL DL나 OWL Full에서 필수 기능이라고 생각되어 지는 unionOf, intersactionOf 등 클래스들의 집합 관계에 대한 표현 부분만을 포함하였다.

3. OWL의 확장 요소 분류

본 장에서는 OWL이 클래스나 속성간의 관계를 정의하는데 있어 RDF, RDF 스키마에 비해 확장된 개념들에 대하여 분류한다. 특히 클래스간 또는 속성간의 관계 표현에 필요한 의미의 동일성과 이질성에 대한 분류와 클래스 집합에 대한 표현은 RDF나 RDF 스키마에서는 지원하지 않는다.

<표 1>은 OWL에서 확장 요소를 분류하고 그들의 기능을 간단히 설명한다[1][7].

구분	Property	설명
동등성	equivalentClass	- 두 클래스가 동등함을 표현
	equivalentProperty	- 두 속성이 동등함을 표현
이질성	differentFrom	- 두 개체가 서로 다른 사실임을 정의
	AllDifferent	- 여러 개체가 서로 다른 사실을 한꺼번에 기술
클래스 집합	unionOf	- 가용이었던 클래스들을 하나의 클래스로 조합하여 새로운 클래스를 정의
	intersectionOf	- 알려진 모든 조건을 동시에 만족시켜야 하는 자를 모두 고 있음
	complementOf	- 특정 클래스에 속하지 않는 클래스를 정의
	oneOf	- 한 클래스를 구성하는 구성들을 열거
	disjointWith	- 한 클래스의 구성원이 다른 클래스의 구성원이 될 수 없다는 사실을 규정
역연계	inverseOf	- 두 속성의 역속성의 명사기 기능

<표 1> OWL의 확장 요소 분류

4. OWL 문서의 저장

4.1 저장구조

본 절에서는 OWL이 제공하는 기본적인 기능뿐만 아니라 클래스나 속성의 동치 관계, 이질 관계, 역관계, 클래스들의 집합 관계의 추가적 기능을 위해 관계형 데이터베이스에 기반한 저장구조를 제안한다.

본 논문에서 제안한 저장구조는 Class, Property, NameSpace, subClassOf, subPropertyOf, Tag, EqualClass, EqualProperty, ComplexClass 테이블로 구성되어 있다.

구성 요소	주요기능
class	• OWL 문서에 정의된 클래스의 이름과 각 클래스를 식별하는 네임스페이스의 정보 저장
property	• OWL 문서에 정의된 속성의 이름과 각 속성을 식별할 수 있는 네임스페이스, 각 속성의 도메인과 레인지 정보 저장
subProperty	• 각 클래스간의 포함 관계 저장
subClass	• 각 프로퍼티간의 포함 관계 저장
equalityClass	• 클래스의 이름은 다르지만 의미는 같은 클래스간의 관계 저장
equalityProperty	• 이름은 다르지만 의미는 같은 속성간의 관계 저장
nameSpace	• 클래스나 속성들의 고유 식별자 역할을 담당하는 네임스페이스의 정보 저장
Tag	• 각 기능별 속성의 이름 분류를 위한 코드 정보 저장
ComplexClass	• 클래스 집합에 대한 unionOf, intersectionOf, oneOf 그리고 disjointWith 정의의 사용 여부와 집합 내 구성 요소인 클래스나 속성들을 함께 저장하여 명확한 관계 표현을 저장

<표 2> OWL의 저장 테이블 구성

Class 테이블에는 OWL 문서에서 정의된 클래스의 이름과 각 클래스를 식별할 수 있는 네임스페이스 정보를 저장한다. Property 테이블에는 OWL 문서에 정의된 속성의 이름과 각 속성을 식별할 수 있는 네임스페이스를 정보, 각 속성의 도메인과 레인지 정보를 저장한다. Class나 Property들의 고유한 식별자 역할을 하는 네임스페이스 정보는 NameSpace 테이블에 저장한다.

<Class>	cid	ns	name		
<Property>	pid	ns	name	domain	range
<EqualityClass>	cid	tag	target		
<EqualityProperty>	cid	tag	target		
<ComplexClass>	cid	tag	T_target		
<Tag>	tid	tag			
<subClassOf>	cid	superClass			
<subPropertyOf>	pid	superProperty			
<NameSpace>	nid	namespace			

[그림 1] OWL 저장 테이블의 속성 구조

각 클래스간의 포함관계는 subClassOf 테이블에, 각 Property의 포함관계는 subPropertyOf 테이블에 저장한다. 또한 자원의 동일성, 이질성 클래스 집합에 대한 표현을 위해 Tag 테이블, EqualClass 테이블, EqualProperty 테이블, 그리고 ComplexClass 테이블을 유기적으로 연관시켜 상관관계를 표현하였다. EqualClass 테이블은 클래스의 이름은 다르지만 의미는 같은 클래스들의 관계를 저장하고, EqualProperty 테이블은 동일한 의미를 가지는 속성들에 대한 정보를 저장한다. EqualClass과 EqualProperty 테이블을 이용하여 동음이의 또는 이음동의어들에 대해 보다 정확한 결과를 얻을 수 있다. Tag 테이블에서는 각 개별 속성을 분류하기 위한 코드 정보를 저장한다. 마지막으로 ComplexClass 테이블은 클래스 집합에 대한 표현 방법인 unionOf, intersectionOf, Complement Of, oneOf 그리고 disjointWith중 사용된 속성을 구분하고, 집합 내 구성 요소인 클래스나 속성들을 저장하여 그들 간의 명확한 관계 표현이 가능하도록 한다.

4.2 저장전략

본 절에서는 OWL로 작성된 통합 뉴스 스키마인 온톨로지 문서를 RDB에 저장하기 위한 저장 전략을 제안한다. RDF로의 저장은 기존에 RDF가 지니고 있는 트리플 구조를 유지하면서, OWL의 확장된 기능을 지원할 수 있도록 저장해야 한다.

규칙	저장 규칙
1	• OWL의 클래스 생성 규칙에 의해 생성된 클래스는 클래스 테이블에 저장
2	• 클래스 명으로 시작하여 이미 생성된 클래스의 하위 클래스를 생성할 때 사용되는 클래스 역시 1번과 동일한 형태로 클래스 테이블에 저장하고 이는 subClassOf의 속성으로 간주하고 이 관계를 subClassOf 테이블에 저장
3	• OWL의 속성 생성 규칙에 의해 생성된 속성은 속성 테이블에 저장
4	• 각 클래스와 속성의 포함관계를 나타내는 subClassOf나 subPropertyOf 관계는 각각 subClassOf와 subPropertyOf 테이블에 저장
5	• 집합관계를 표현하는 unionOf, intersectionOf, ComplementOf, disjointWith 정보는 ComplexClass 테이블에 저장
6	• 클래스에 대한 동음이의어나 이음동의어에 대한 관계 표현을 위해 사용되는 equivalentClass, differentFrom, Alldifferent 등은 EqualityClass 테이블에 저장
7	• 속성에 대한 동음이의어나 이음동의어에 대한 관계 표현을 위해 사용되는 equivalentProperty 속성은 equalProperty 테이블에 저장
8	• 클래스의 제약조건을 명시한 Restriction에 대한 저장은 실제 기사를 저장하여 검색하는 데는 그 사용의 효율성이 부족하므로 무시

<표 3> RDB에 저장하기 위한 저장 규칙

[그림 2]는 OWL을 이용하여 와인에 대한 온톨로지를 구성한 문서의 일부분이다. [그림 3]은 [그림 2]의 OWL 문서를 본 논문에서 제안한 저장구조로 저장한 결과를 보여준다. 동치 관계에 대한 부분은 Property테이블이나 Class 테이블에서 테이블에 밀출을 그어 표현했다.

```

<rdf:RDF
  xmlns = "http://www.example.org/wine#"
  xmlns:food = "http://www.example.org/food#"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdfs = "http://www.w3.org/1999/02/22/rdf-schema#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  <owl:Class rdf:ID="Wine" />
  <owl:Restriction
    <owl:Property rdfs:resource="hasColor" />
    <owl:cardinality 1</owl:cardinality>
    <owl:Restriction>
      <rdf:subClassOf>
        <owl:Restriction>
          <owl:oneOfProperty rdfs:resource="hasMaker" />
          <owl:allValuesFrom rdfs:resource="Wineery" />
        <owl:Restriction>
          <rdf:subClassOf>
            <owl:ObjectProperty rdfs:ID="hasColor" />
            <rdf:subPropertyOf rdfs:resource="hasWineDescriptor" />
            <rdf:domain rdfs:resource="Wine" />
            <rdf:range rdfs:resource="WineCakes" />
          <owl:ObjectProperty>
            <owl:SubPropertyOf rdfs:ID="hasSugar" />
            <rdf:domain rdfs:resource="Wine" />
            <owl:ObjectProperty>
            <owl:SubPropertyOf rdfs:resource="hasWineDescriptor" />
            <rdf:domain rdfs:resource="WineSinger" />
            <owl:ObjectProperty>
            <owl:Class rdf:ID="WineColor" />
            <rdf:subClassOf rdfs:resource="WineDescriptor" />
            <owl:range rdfs:resource="Collection" />
            <owl:Item rdfs:resource="White" />
            <owl:Item rdfs:resource="Rose" />
            <owl:Item rdfs:resource="Red" />
          <owl:Class>
            <owl:Class rdf:ID="Wineery" />
            <owl:Class rdf:ID="Region" />
            <owl:Class rdf:ID="WineSinger" />
            <rdf:subClassOf rdfs:resource="WineTaste" />
            <owl:instanceOf rdfs:resource="Collection" />
            <owl:Class rdf:about="WineDry" />
            <owl:Class rdf:about="WineDry" />
            <owl:UnionOf>
            <owl:Class>
            <Wineery rdfs:ID="Stabak" />
            <Wineery rdfs:ID="White hallLane" />
            <Wineery rdfs:ID="White hallLane" />
            <Region rdfs:ID="New ZealandRegion" />
            <Region rdfs:ID="Napa" />
            <Region rdfs:ID="NapaRegion" />
            <Wine rdfs:ID="StabakWine" />
            <domain rdfs:resource="New ZealandRegion" />
            <hasColor rdfs:resource="Stabak" />
            <hasColor rdfs:resource="White" />
            <hasWine>
            <Collection rdfs:ID="WhitehallLaneCobernetFrom" />
            <Scotsdin rdfs:resource="Wine" />
            <hasMaker rdfs:resource="White hallLane" />
            <hasSugar rdfs:resource="Wine" />
            <Collection>
            <DessertWine rdfs:ID="WhitehallLanePissavers" />
            <Scotsdin rdfs:resource="WinehallLane" />
            <hasMaker rdfs:resource="White hallLane" />
            <hasSugar rdfs:resource="Sweet" />
            <DessertWine>
            <Region rdfs:ID="Napa" />
            <owl:equivalentClass rdfs:resource="NapaRegion" />
            <Region>
            <Wineery rdfs:ID="White hallLane" />
            <owl:differentFrom rdfs:resource="WhitehallLane" />
            <Wineery>
            <rdf:RDF>
  
```

[그림 2] OWL을 이용한 와인 온톨로지 문서

<Class>			<Property>					<EqualityProperty>		
id	in	name	id	in	name	domain	range	id	tag	target
1	2	Wine	1	2	hasColor	1	2	4	5	6
2	2	WineColor	2	2	hasMaker	1	6			
3	2	WineDescriptor	3	2	hasSugar	1	4			
4	2	WineSugar	4	2	hasColor	1	7			
5	2	WineTaste	5	2	hasWineDescriptor	1	3			
6	1	Winery	6	2	hasColor	1	7	21	3	22
7	1	Winery	7	3	hasColor	6	17	18	4	19
8	1	White	8	3	hasColor	6	18			
9	1	Red	9	3	hasColor	6	19			
10	1	Red	10	3	hasColor	7	20			
11	1	Sweet	11	3	hasColor	7	21			
12	1	Citry	12	3	hasColor	7	22			
13	1	Dry	13	3	hasColor	14	23			
14	3	IceWine	14	3	hasColor	15	24			
15	1	CabernetWine	15	3	hasColor	16	26			
16	1	DessertWine	16	1	hasColor	23	20			
17	1	Sekia	17	1	hasColor	23	17	2	1	8
18	1	WhitehallLane	18	1	hasColor	24	18	3	2	5
19	1	WhitehallLane	19	1	hasColor	24	8	2	1	9
20	1	NewZealandRegion	20	1	hasColor	24	27	2	1	10
21	1	Napa	21	1	hasColor	24	18	3	2	5
22	1	NapaRegion	22	1	WineTaste	24	13	3	2	2
23	1	SekiaWine	23	1	hasColor	25	22	4	2	11
24	1	WhitehallLaneCabernetWine	24	1	hasColor	25	19	4	2	12
25	1	WhitehallLanePinotWine	25	1	hasColor	25	11	4	2	13

<EqualityClass>		
id	tag	target
21	3	22
18	4	19

<subClassOf>		
id	superClass	
2	3	
5	3	

<ComplexClass>		
id	tag	target
2	1	9
2	1	10
3	2	5
3	2	2
4	2	11
4	2	12
4	2	13

<Namespace>		<Tag>		<subPropertyOf>	
id	namespace	id	tag	id	superProperty
1	http://www.example.org/wine#	1	one of	1	5
2	http://www.w3.org/2002/07/owl#	2	unionOf	2	5
3	http://www.w3.org/1999/02/22-rdf-syntax-ns#	3	separatesClass	3	5
4	http://www.w3.org/2000/01/rdf-schema#	4	differentFrom	4	5
		5	equivalentProperty	4	5

[그림 3] OWL문서의 저장 예

5. 질의 전략

기존의 RDF와 RDF 스키마의 질의어인 RQL에서는 스키마에 대한 질의, 인스턴스에 대한 질의, 스키마와 인스턴스를 함께 고려한 기본적인 질의 형태를 제공하고 있다[4]. 그렇지만 RQL은 OWL에서 제공하는 확장된 기능들에 대한 질의 형태는 정의하지 않았다. 따라서 본 논문에서는 관계형 데이터베이스에 저장된 OWL과 RDF문서를 대상으로 SQL을 이용해 질의 할 수 있도록 RQL을 SQL로 변환하는 과정을 제안하고, RQL을 이용하여 질의 할 수 없었던 부분들을 SQL을 이용하여 질의 할 수 있도록 제안하였다.

5.1 질의 패턴 분류

RQL은 스키마를 세 종류의 질의 패턴을 가지고 있다. 첫째는 스키마를 이용한 질의, 둘째는 인스턴스를 이용한 질의, 마지막으로 스키마와 인스턴스를 동시에 이용한 질의가 있다. 본 절에서는 본 연구에서 제안한 전략으로 관계형 데이터베이스에 저장한 데이터를 대상으로 RDF 질의 언어인 RQL을 이용하여 질의 가능한 형태를 분류하고 이를 SQL로 변환하는 전략을 제안하였다.

본 절에서는 본 연구에서 제안한 전략으로 관계형 데이터베이스에 저장한 데이터를 대상으로 RDF 질의 언어인 RQL을 이용하여 질의 가능한 형태를 분류하고 이를 SQL로 변환하는 전략을 제안하였다.

OWL은 RDF나 RDF 스키마보다 클래스간 또는 속성간의 관계를 보다 명확하게 표현할 수 있다. 특히 클래스나 속성의 동일 관계, 이질 관계, 역관계의 정의 그리고 클래스들의 집합 관계에 대한 표현들은 각 자원들의 의미있는 관계를 보다 정확하게 표현할 수 있다. 기존의 RDF와 RDF 스키마의 질의어인 RQL에서는 스키마에 대한 질의, 인스턴스에 대한 질의, 스키마와 인스턴스를 함께 고려한 기본적인 질의 형태를 제공하고 있다[4]. 그렇지만 기존의 RQL은 OWL에서 제공하는 확장된 기능들에 대한 질의 형태는 정의하지 않았다. 따라서 본 논문에서는 기존의 RQL에서 제공하는 기본적인 질의 형태에 OWL이 추가적으로 제공하는 기능에 대한 질의 형태를 보장하고, 기존 RQL을 이용해 표현할 수 있도록 RQL을 확장한다. [그림 4]는 기본적인 OWL의 질의 형태와 기존 RDF 저장 구조에서는 가능하지 않았던 OWL만의 특성에 관한 새로운 질의 형태를 분류하고 이를 기존의 RQL를 이용하여 표현한 예이다[4].

Q1, Q2, Q3, Q5, Q8, Q9의 질의문은 기존에 RDF, RDF 스키마 기반에의 질의에서도 가능한 질의들이다. 그렇지만 Q4, Q6, Q7, Q10는 RQL에서는 지원하지 않는 새로운 형태의 질의이다.

Q4의 질의문은 Winery Class의 하위클래스를 검색하는 질의문이다. 직관적으로 보기에는 2개의 Winery를 포함하는 것처럼 보인다. 하지만 [그림 2]의 OWL 문서에서 WhitehallLane라는 양조장은 differentFrom을 사용하여, 이름은 비슷하지만 다른 양조장이라는 정보를 제공하고 있다. 따라서 기존의 저장 구조와 RQL 질의 형태로는 differentFrom 속성의 특징을 지원하지 못하기 때문에 이와 같은 새로운 질의에 대한 정확한 결과를 얻을 수 없다[4].

질문	OWL문서에 대한 RQL 표현	질의 패턴
Q1	hasGtr 속성의 Range를 찾는다. Select \$C From hasGtr(\$C)	- 속성의 Range 검색
Q2	Winey 클래스의 인스턴스 도메인을 찾는다. Select @_range(@) From (\$C) @ Where \$C = "Winey"	- 클래스의 인스턴스 속성의 검색
Q3	Wine 클래스의 도메인 SubClass를 검색한다. subClassOf(Wine)	- 하위 클래스 검색
Q4	Winey 클래스의 하위 클래스의 숫자를 찾는다. Select count(*) From (\$C) subclassOf(\$C) Where \$C = "Winey" and exist equality(Yinclud range(differentfrom))	- 하위 클래스의 수 계산 - 이질성에 대한 검색
Q5	Wine의 색이 인스턴스인 Wine을 찾는다. Select X Y From (\$hasGtr(Y) Wine \$Y = "Wine"	- 조건에 인스턴스 클래스 검색
Q6	Napa는 지역에서 생산되는 Wine을 찾는다. Select X From (\$hasGtr(\$C) Where \$C = "Napa" and exist equality(\$C include range(equivalentClass))	- 조건에 인스턴스 대상 검색 - 이질성에 대한 검색
Q7	Wine의 길이(Dry와 Whiteall Lane 지역에서 생산되는 Wine을 찾는다. Select X Y From hasSign(\$hasMiter(Y) Where \$Y = "Dry" and \$Y = "Whiteall Lane" and exist equality(Yinclud range(differentfrom))	- 조건에 인스턴스 대상 검색 - 이질성에 대한 검색
Q8	hasMiter 속성의 Range가 ticks인 클래스를 찾는다. Select \$C From hasGtr(\$C) Where \$C = "ticks"	- 속성의 Range 중 조건에 인스턴스 클래스 검색
Q9	Winey 클래스가 있는 속성의 Domain을 찾는다. Select X From (\$hasGtr(\$C) Where \$C = "Winey"	- 조건에 인스턴스 속성의 Domain 검색
Q10	Napa 지역에서 생산되는 Wine과 인스턴스 도메인을 찾는다. Select @_range(@) From hasGtr(\$C) @ Where \$C = "Napa" and exist equality(\$C include range(equivalentClass))	- 조건에 인스턴스 클래스와 인스턴스 도메인 속성의 검색 - 이질성에 대한 검색

[그림 4] OWL 문서에 대한 질의 패턴과 RQL 표현

Q6는 Napa 지방에서 생산되는 와인을 찾는 질의문이다. [그림 2] 문서에서 Napa와 NapaResion 지역은 동일한 지역임을 설명하고 있다. 따라서 Napa 지역에 대한 질의 시 NapaResion 지역의 와인도 질의 결과에 포함되어야 한다. Q7 역시 Q4와 마찬가지로 클래스의 이질적인 부분을 고려해야 하므로 기존에 RQL 질의 형태로는 표현할 수 없다. Q10번의 경우도 Q6와 마찬가지로 클래스간의 동일성 여부를 체크해야 하므로 기존 RQL의 질의 형태로는 표현할 수 없다.

본 논문에서는 Q4, Q6, Q7, Q10과 같은 질의를 지원하기 위하여 RQL에 새로운 "exist equality"라는 기능을 추가한다. 만약 동일한 의미의 클래스나 속성이 존재하면 질의 처리 시 고려하여 보다 정확한 질의 결과를 얻을 수 있도록 한다. 또한 클래스나 속성의 이름은 비슷하지만 다른 의미를 가지고 있는 요소들은 제거하여 정확한 질의 결과를 유도할 수 있다. Q4, Q6, Q7, Q10에서 "exist

equality"를 이용해 확장된 RQL의 예를 볼 수 있다.

5.2 질의 변환 전략

본 절에서는 OWL 스키마를 기반으로 RDF로 작성된 문서 (예 : 뉴스 기사)들을 검색하는 질의 기법에 대해 설명한다. RDF로 작성된 뉴스기사 검색에 있어 기존의 RQL을 관계형 데이터베이스에 저장된 데이터에 대해 질의 가능한 SQL의 형태로 변환하는 방법이 필요하다.

① 선택

사용자가 원하는 클래스나 속성 값을 지정

② 추출

선택한 클래스와 속성 값을 바탕으로 문서에서 특정 요소들을 추출

③ OWL 확장 기능 검사

추출된 클래스와 속성과 동일, 또는 이질의 관계에 가진 요소가 있는지 검사

④ 동일 요소 추출 및 이질 요소 제거

사용자가 원하는 질의 값과 동일 요소가 있다면, 그 요소를 추출하고 만약 이질의 요소들이 같이 추출된다면 이를 결과 값에서 제외시킴

⑤ 조합

추출된 클래스와 속성 값과 그 동일 요소들의 값을 통합하거나 이질 요소들의 값을 제거시킴

5.3 질의 변환 단계

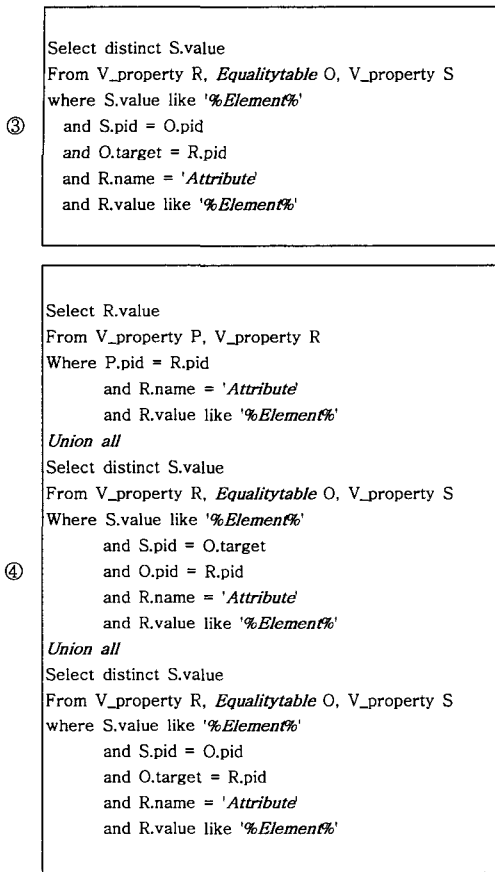
제안된 SQL 변환의 실제 실행 단계는 [그림 5]와 같다.

①

```
Select R.value
From V_property R
Where R.name = 'Attribute'
and R.value like '%Element%'
```

②

```
Select distinct S.value
From V_property R, Equalitytable O, V_property S
Where S.value like '%Element%'
and S.pid = O.target
and O.pid = R.pid
and R.name = 'Attribute'
and R.value like '%Element%'
```



[그림 5] RQL-to-SQL 변환 과정

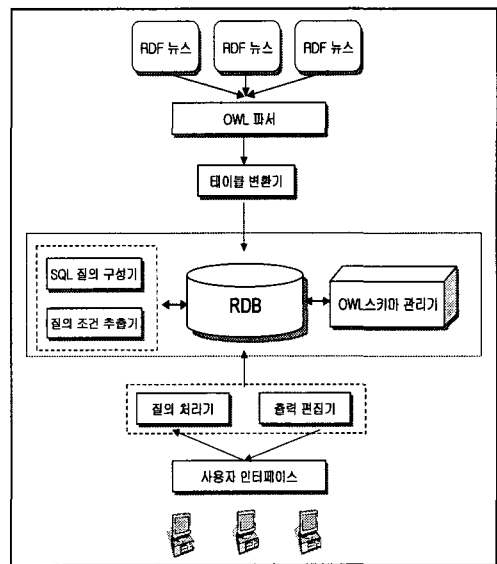
[그림 5]는 실제로 SQL로 변환하는 과정을 네 단계로 분류해 놓은 것이다.

①번 단계에서는 사용자가 입력한 엘리먼트를 기반으로 입력된 조건에 맞는 엘리먼트 Range 값으로 가지고 있는 Domain을 찾아낸다. ②번 단계에서는 ①번에서 검색의 조건으로 사용된 클래스나 엘리먼트가 EqualityClass나 EqualityProperty 테이블에 존재하는지를 체크한다. ③번 단계에서는 반대로 조건으로 제시된 클래스나 속성을 동일요소로 선택한 클래스나 속성을 존재 여부를 검사한다. 마지막으로 ④번 단계에서 위의 모든 과정을 Union all을 사용하여 통합한다.

6. 구현

본 장에서는 제안된 저장 구조와 질의 변환 전략을 이용하여 실제 도메인에서 그 효율성을 실험했다. 본 장에서는 온라인 뉴스 검색을 위한 통합 OWL 스키마를 구현하고 이를 바탕으로 작성된 뉴스들을 OWL의 확장된 개념들을 적용하여 좀더 정확한 검색이 가능하도록 하였다.

[그림 6]는 본 논문에서 제안한 시스템의 전체 구성도이다.



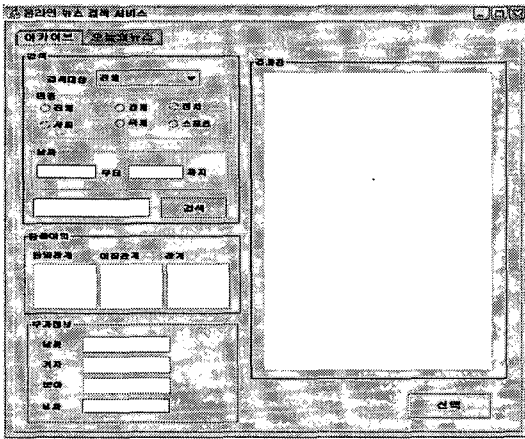
[그림 6] 시스템 전체 구성도

6.1 구현 환경

운영체제로 Windows XP를 사용하였고, 사용자를 위한 인터페이스 구현에는 Java를 이용하였으며, 관계형 데이터베이스는 MySQL을 사용하였다.

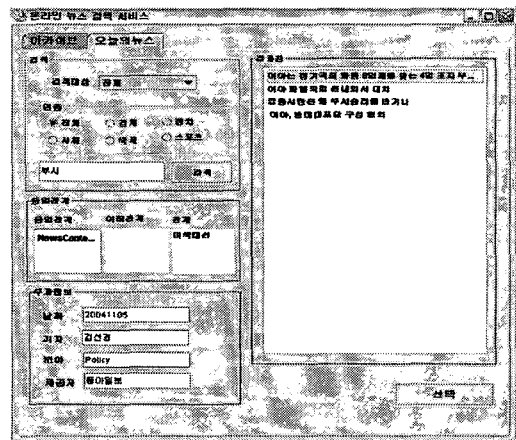
6.2. 질의 인터페이스 및 결과

구현된 시스템에서는 질의 예제로 두 종류의 뉴스 서비스를 제공하였다. 첫 번째는 그 날의 뉴스를 검색하는 것이고, 두 번째는 지난 뉴스들에 대한 아카이브 서비스를 제공하고 있다.

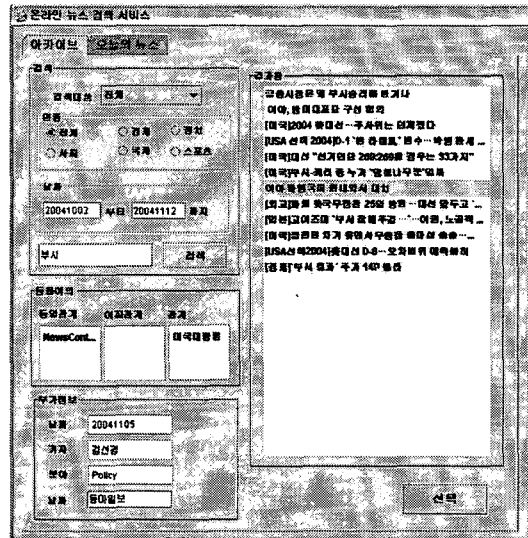


[그림 7] 뉴스 검색 시스템 초기화면

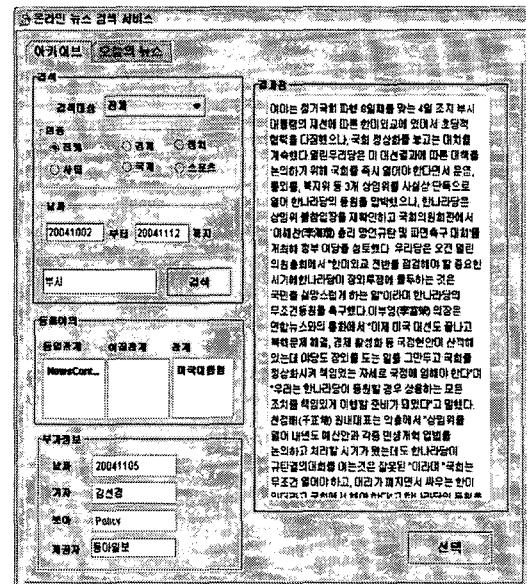
[그림 7]은 온라인 뉴스 검색 시스템의 초기화면이다. 이 곳에서 오늘의 뉴스를 검색할 것인지 아카이브 서비스를 받을 것인지를 결정한 후 질의를 입력한다. [그림 8]은 조건을 입력하여 “부시”와 관련된 기사들을 검색하였다. 오늘의 뉴스 중 부시에 관련된 모든 기사의 기사 제목들을 결과 창에서 볼 수 있다. [그림 9]는 아카이브 서비스에서 조건을 입력하여 검색된 기사의 제목을 보여주고 있다. [그림 10]은 아카이브 서비스 검색 결과로 나온 제목들 중 사용자가 원하는 결과를 선택하여 그 기사를 출력한 예이다.



[그림 8] 오늘 뉴스 조건 입력 후 결과



[그림 9] 아카이브 서비스 조건 입력 후 결과



[그림 10]. 기사 검색 결과

7. 결론

OWL은 RDF나 RDF 스키마보다 클래스간 또는 속성간의 관계를 보다 명확하게 표현하기 위해 동치 관계, 이질 관계, 역관계, 클래스들간의 집합

관계 표현 등의 다양한 기능을 제공하고 있다. 따라서 본 논문에서는 OWL로 표현된 문서에 쉽게 접근하고 질의 처리를 하기 위하여 기존의 RDF나 RDF 스키마로부터 동일 관계, 이질 관계, 클래스들의 집합 관계에 대한 기능 등이 추가된 OWL 문서를 관계형 데이터베이스에 저장하기 위한 구조를 제안한다. 또한 사용자가 보다 정확하게 OWL 문서의 내용에 대해 질의할 수 있도록 기존의 RDF와 RDF 스키마에 대한 질의 형태를 보완하고 추가 질의 형태를 제안한다. 또한 본 논문에서 제안한 저장 구조와 질의 형태를 이용하여 OWL 문서의 저장 모듈과 사용자 인터페이스, 질의 처리 모듈로 구성된 OWL 저장 시스템을 설계하고 구현한다.

참고문헌

- [1] W3C, OWL Web Ontology Language Overview, <http://www.W3C.org/TR/owl-features/>
- [2] W3C, RDF vocabulary Description Language 1.0 : RDF Schema, <http://www.w3.org/TR/rdf-schema/>
- [3] W3C, RDF Primer, <http://www.W3.org/TR/2003/WD-rdf-primer-20030123>
- [4] G. Karvounarakis, A. Magganaraki, S. Alexaki, V. Christophides, D. Plexousakis, Michel Scholl, Karsten Tolle, "Querying the Semantic Web with RQL", Computer Network, Vol.42, No. 5, 2003.
- [5] Jeen Broekstra, Arjohn Kampman, Frank van Harmelen, "Sesame : A Generic Architecture for Storing and Querying RDF and RDF Schema", International Semantic Web Conference, P54 ~ P68, 2002.
- [6] Sofia Alexaki, Vassilis Christophides, Gregory Karvounarakis, Dimitris Plexousakis, and Karsten Tolle, "The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases", SemWeb 2001.
- [7] 오삼균, "Web Ontology Language와 그 활용에 관한 고찰", 데이터베이스 연구회 논문집, 18권 3호, P63 ~ P79, 2002.
- [8] NewsML Version 1.1 Functional Specification, "http://www.newsml.org/IPTC/NewsML/1.1/specification/NewsML_1.1-spec-functionalspec_6.html", 2002.
- [9] Peter F. Patel-Schneider, Jérôme Siméon, "The Yin/Yang web: XML syntax and RDF semantics", WWW 2002, pages 443-453.
- [10] Mounia Lalmas and Ian Ruthven, "A Model for Structured Document Retrieval: Empirical Investigations", In Proceedings of the Hypertext, Information Retrieval, and Multimedia, 1997.
- [11] 고명석, 김연희, 김병곤, 임해철, "OWL 문서의 저장과 질의 형태에 관한 연구", 한국정보과학회 추계학술발표대회 논문집, 31권 2호, p244~p246, 2004

신 판 섭



- 1992년 홍익대학교 전자계산학과 (이학사)
- 1994년 홍익대학교 전자계산학과 (이학석사)
- 2000년 홍익대학교 전자계산학과 (이학박사)
- 2002년~현재 : 대전대학교 컴퓨터공학과 조교수

- 관심분야 : 분산·객체 데이터베이스, XML, 멀티미디어 시스템, e-러닝 시스템