

RDF 문서의 저장소와 RDQL 질의 처리기의 설계 및 구현

(Design and Implementation of RDF Storage and RDQL Query Processor)

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 Yim) (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 웹 자원에 대해 지식구조를 기술하여 지능적인 정보 처리를 목적으로 하는 RDF에 대해 관계형 데이터베이스 모델을 설계하고 저장 시스템을 통해 의미 있는 질의를 처리할 수 있는 질의 처리기를 설계 및 구축하여 효율적인 메타 데이터의 저장, 관리 및 질의를 지원하고자 한다. 저장을 위한 모델은 관계형 모델을 사용함으로써 객체형 모델이나 메모리 모델을 사용하는 한계점을 극복할 수 있을 것이다.

키워드 : 시멘틱 웹, RDF, 저장, 질의, RDQL

Abstract In spite of computer's development, the present state of a lot of electronic documents overflowed it's going to be more difficult to get appropriate information. Therefore it's more important to get meaningful information than to focus on the speed of processing. Semantic web enables and intelligent processing by adding semantic meta data on your web documents. Also as the semantic web grows, the knowledge resource is more important.

In this paper, we propose a RDF storage system using relational database model aimed at intelligent processing by adding semantic meta data on your web documents, also a query processor aimed at query processing through the storage system. By using relational model, we could overcome a weakness of object or memory model.

Key words : Semantic Web, RDF, Storage, Query, RDQL

1. 서론

정보의 급속한 확산에 따라 컴퓨터분야의 연구는 용

량과 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 메타 데이터를 통해 데이터의 구조를 부여하고 이를 통해 효율적으로 데이터를 처리하는 방식들이 있다[1,2]. 이와 같이 데이터에 구조를 부여해서 효율적으로 처리하는 것과는 또 다른 접근방법으로서 정보에 대한 의미적인 메타 데이터를 기술함으로써 효과적인 처리를 도모하고자 하는 시멘틱 웹과 같은 방법이 있다. 전자의 방식이 처리비용이나 속도를 고려한 데

· 본 연구는 BK-21 정보기술 사업단과 정보통신부 ITRC(e-Biz 기술 연구 센터)에 의해 지원되었음

[†] 학생회원 : 서울대학교 전기컴퓨터공학부
 hyjeong@idb.snu.ac.kr
 jmkim@idb.snu.ac.kr
 jwjung@idb.snu.ac.kr
 jnkim@idb.snu.ac.kr
 dhlhm@idb.snu.ac.kr

^{**} 종신회원 : 서울대학교 컴퓨터공학부 교수
 hjk@idb.snu.ac.kr

논문접수 : 2004년 1월 29일

심사완료 : 2006년 3월 30일

이타 관리 효율에 중점을 둔 방법이라면 후자의 방식은 의미가 있는 데이터를 구축하고 지능적인 처리를 수행하는 것에 중점을 둔 방식이다[3]. 물론 구조적 정보를 이용하는 기법도 정보를 처리해서 의미정보를 추출하거나 지능적인 처리를 수행할 수도 있겠지만 처리 수행에 의존적이며 그 결과도 지능적이라고 하기에는 부족한 점이 많다. 시멘틱 웹과 같은 방식은 의미와 연관성을 이용한 처리라는 특성 외에도 정보 기술 능력에 있어 기존의 데이터를 확장하기에 좋은 특성을 가진다. XML이 데이터베이스에 비해 구조의 자율성을 가지지만 구조적 정보를 이용한 처리를 위해서는 처음부터 XML구문을 염두에 두고 작성해야 한다. 따라서 이미 보편화되어있는 HTML 웹 뿐만 아니라 수많은 종류의 데이터를 XML화 한다는 것은 현실적으로 어렵다. 반면 시멘틱 웹은 기존의 HTML이나 XML같은 웹에 대해서뿐만 아니라 DB의 특정 레코드와 같이 자료의 위치를 기술할 수 있는 모든 대상에 대해서 의미, 연관적 정보와 같은 메타 데이터를 기술하는 것이 가능하다[4,5].

시멘틱 웹은 다양한 모델을 통해 기술 될 수 있다. 현재 대표적인 시멘틱 웹으로서 W3C에서 웹 자원에 대한 의미 정보를 기술하기 위해 제정한 RDF(Resource Description Framework), DARPA의 의미적 정보 기술 및 지능형 에이전트 지원을 위한 DAML+OIL[6], ISO에서 표준이나 지식관리 분야에 중점을 두고 개발한 TopicMap[7]이 있다. W3C에서는 최근 RDF와 DAML+OIL을 확장한 OWL(Ontology Web Language)을 표준화하였다. 이 모델들은 정보의 표현방법에 있어서 차이를 가지지만 의미적 정보와 연관관계, 그 외에 지능적인 처리를 위한 부가 정보들을 표현하는데 있어서는 동일한 표현능력을 가진다. 다만 이 모델들이 개발된 동기에 차이가 있는 만큼 활용 분야에 따라서 조금씩 다른 특성을 보인다. 웹이 발전함에 따라 웹 자원이 방대해진 것과 마찬가지로 시멘틱 웹이 발전함에 따라 이러한 웹 자원에 대한 지식구조를 기술하고 색인을 제공할 수 있는 메타 데이터도 증가 하게 되는데 웹 자원에 대한 효율적인 저장과 관리가 중요한 것처럼 메타 데이터를 관리하고 효율적인 질의를 처리할 수 있는 저장 시스템에 대한 중요성 또한 높아지고 있다.

본 논문에서는 앞에서 말한 여러 가지의 웹 자원에 대한 지식구조 기술 모델 중 웹 자원을 지칭하는 URI(Uniform Resource Identifier)를 이용하여 이 자원을 표현하는 부가 정보와 다른 자원과의 연관관계를 제시하는 RDF 모델에 대해 효율적인 저장 시스템과 RDF 질의(RDQL)를 처리할 수 있는 시스템을 설계한다.

논문의 구성은 2장에서 RDF에 관한 기본 배경 지식과 저장 및 질의 처리기에 대한 관련 연구에 대해 알아

보고 3장에서는 본 논문에서 제안하는 RDF 저장 시스템에 대해 설명한다. 4장에서는 RDQL에 대해 설명하고 5장에서 결론 및 향후 연구과제에 대해 알아본다.

2. 관련 연구

색인(index), 용어해설(glossary), 시소러스(thesaurus)는 책이나 다른 정보자원에 담겨있는 지식 구조를 매핑하는 방법들이다. 인공지능 연구분야에서는 예전부터 인간과 기계사이에 의사전달을 위해 지식을 어떻게 표현할 것인가에 대해 연구를 해 왔는데 지식 표현에 널리 사용된 방법 중의 하나는 개념(concepts)과 개념들 간의 관계(conceptual relationship)로 구성된 개념도(conceptual graphs)이다. 이 개념도는 실제 AI시스템에서 'semantic network', 'associative network', 'partitioned network' 등 다양한 이름으로 구현되어 왔으며 이 semantic network의 기본 모델은 전통적인 분류 기법인 색인, 용어해설, 시소러스에서 나타나는 토픽, 연관관계와 매우 유사하기 때문에 이 두 기법을 접목함으로써 정보관리와 지식관리 분야에 많은 이점을 제공할 수 있고 이것이 시멘틱 웹 발전의 토대가 되었다.

W3C에서는 웹 페이지들에 대해 그 페이지 자체를 설명하는 메타정보-페이지에 담겨있는 내용의 주제, 만든 일자, 관련페이지 등-를 기술함으로써 웹 페이지의 하이퍼링크 뿐만 아니라 시멘틱 링크에 의해서 보다 정확한 검색을 도모하고자 하는 연구를 하였다. 그 결과로 RDF라는 모델이 제시되었으며 웹 자원의 시멘틱 맵을 구축하기 위한 도구로 사용되고 있다.

본 장에서는 본 논문의 주제가 되는 RDF 모델에 대해 살펴 보고 RDF의 저장방식과 RDF 질의에 대한 접근 방식에 대해 알아본다.

2.1 RDF 모델

RDF의 기초는 지정된 특성과 그 값을 표현하기 위한 모델로서 다양한 데이터 표현분야에서 확립된 원칙에 따라 표현된다. RDF의 특성은 자원의 속성이라고 생각할 수 있고, 이런 점에서 전통적인 속성-값에 해당되고 아울러 자원간의 관계를 표현하며 따라서 RDF 모델은 엔티티-관계(ER) 그림과 유사할 수 있다(더 정확히 표현하면, 그 자체로 RDF 데이터 모델의 사례인 RDF 스키마는 ER 그림이다). 객체지향 설계용어로 표현하면 자원은 객체에 대응되고, 특성은 사례 변수에 대응된다. RDF 데이터 모델은 RDF 표현의 의미를 나타내는 구문 중립적인 방법이다.

기본적인 데이터 모델은 그림 1에서 보는 바와 같이 세 개의 객체 유형(triple model)으로 구성된다. 자원은 자원식별기호(Resource Identifier)에 의해 식별되고 자원식별기호는 임의의 앵커 ID를 결합한 URI이다. 특성

| | |
|----|--|
| 자원 | RDF 형식으로 기술되는 모든 것을 자원이라고 한다. 예컨대 HTML 문서인 <http://oopsla.snu.ac.kr/~hyjung>과 같이, 자원은 웹 페이지 전체일 수 있고 또는 문서자원에 포함된 특정한 HTML 요소나 XML 요소와 같이, 자원은 웹 페이지의 일부일 수 있다. 아울러 웹사이트 전체와 같이 자원은 페이지 전체의 집합일 수 있다. 또 자원은 인쇄본과 같이 웹을 통해 직접 접근할 수 없는 객체일 수도 있다. 자원에는 항상 URL과 선택 요소인 앵커 ID가 합쳐져서 이름이 부여된다. 어떤 것이라도 URI를 가질 수 있으며, URI의 확장성으로 인해 생각할 수 있는 어떤 개체에 대해서도 식별기호를 부여할 수 있다. |
| 특성 | 특성은 자원을 기술하기 위해 사용된 특정한 관점, 특징, 속성, 관계이다. 각 특성은 특정한 의미를 가지며, 허용되는 값, 특성이 기술하는 자원의 유형, 다른 특성과의 관계를 정의한다. |
| 문 | 특정 자원과 지정된 특성, 그리고 그 특성의 값을 RDF 문이라고 하고 문을 구성하는 이들 세 부분을 각각 주부(subject)와 술부(predicate), 객체(object)라고 한다. 문의 객체(즉 특성 값)는 다른 자원일 수 있고, 혹은 리터럴(literal)일 수 있다. 즉 자원(URI로 지정된)이거나 단순히 용어일 또는 XML에서 정의된 기타 기본적인 데이터 유형일 수 있다. RDF 용어로 말하면, 리터럴은 XML 마크업인 내용을 가질 수 있으나 RDF 프로세서에 의해 더 이상 평가되지 않는다. 리터럴에서 마크업을 표현하는 방법에서 약간의 구문상의 제한사항이 있다. |

그림 1 RDF 데이터 모델

은 자원에 의미를 부여하는 속성이고 문은 자원에 대한 특성값으로서 이는 또 다른 자원일 수도 있고 리터럴일 수도 있다.

간단한 예로서 다음 문장을 생각해 보자. 리터럴 “Jung Hoyoung”는 자원 <http://oopsla.snu.ac.kr/~hyjung>의 이름 <http://www.w3.org/2001/vcard-rdf/3.0#FN>이다 라는 문장은 그림 2와 같이 나타낼 수 있다.

| | |
|---------------------|---|
| Subject(Resource) | <http://oopsla.snu.ac.kr/~hyjung> |
| Predicate(Property) | <http://www.w3.org/2001/vcard-rdf/3.0#FN> |
| Object(Literal) | “Jung Hoyoung” |

그림 2 RDF 예제

이 문서에 방향 그래프와 레이블을 부여하여 RDF문을 그림으로 도시할 수 있다(“노드와 아크 도표”라고도 한다). 이 도표에서 노드(타원형으로 그려진)는 자원을 표현한 것이고, 아크는 이름이 부여된 특성을 표현한 것이다. 문자열 리터럴을 표현한 노드는 직사각형으로 표현된다. 따라서 위의 문장은 그림 3과 같이 표현할 수

있다. 화살표의 방향이 중요하다. 아크는 항상 주부에서 시작하여 문의 객체를 지시한다. 그림 3은 자원 <http://oopsla.snu.ac.kr/~hyjung>는 특성 <http://www.w3.org/2001/vcard-rdf/3.0#FN>에 대해 문 “Jung Hoyoung”을 가진다 라고 읽는다. 일반적으로 “자원 <subject>는 특성 <predicate>에 대해 문 “object”를 가진다” 라고 읽을 수 있다.

RDF 모델은 트리 구조로 표현될 수 있으며, 그림 3의 노드와 아크로 표현되는 트리플 구조는 트리를 구성하는 부분이 된다. 자원은 노드, 특성은 아크로 표현하며 문은 다시 다른 객체의 자원일 경우 내부 노드(internal node)와 리터럴인 단말 노드(leaf node)로 표현된다.

2.2 저장 및 질의에 관한 접근

시멘틱 웹이 발전함에 따라 웹 자원에 대한 지식구조를 기술하고 색인을 제공할 수 있는 RDF와 같은 메타 데이터 문서도 증가 하게 되었다. 이는 HTML이나 XML과 같은 웹 자원의 저장 과 관리가 중요하듯 RDF 문서의 효율적인 저장 및 관리가 중요하게 된다. 또한

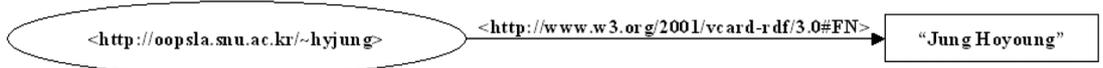


그림 3 단순한 노드와 아크 도표

RDF에서 의미적인 정보를 추출하기 위한 질의 처리 또한 중요한 의미를 가지게 되며 이는 데이터베이스의 스키마와 매우 밀접한 관계가 있다.

이 장에서는 요즘 RDF의 저장과 질의 처리에 관한 접근에 대해 알아본다. RDF의 저장에 관한 모델은 크게 세가지로 나누어 볼 수 있다. 첫째는 전통적인 파일 시스템을 이용하는 방법이 있을 수 있고 둘째는 데이터베이스의 객체형 모델이나 Large Object 모델을 사용하는 방법이 있을 수 있다. 후자의 경우에는 HP Lab.에서 개발한 공개형 모델이 존재한다[8]. 이 두 가지 방법들은 구조가 간단하다는 장점이 있으나 저장시스템에서는 질의를 제공할 수 없기 때문에 인메모리 객체를 생성해야 하는데 대용량의 문서에 대한 처리가 불가능하다. 마지막으로 본 논문에서 제시하는 관계형 데이터 모델을 사용하는 방법이 존재한다. 앞에서도 말한 바와 같이 RDF 트리 모델은 자원, 특성, 문의 트리플 구조로 표현할 수 있으며 이는 관계형 저장 모델에 있어서 중요한 역할을 한다. 다음은 관계형 모델을 사용하여 접근할 수 있는 모델들이고 그림 4와 같다[9].

Horizontal approach: 자원, 특성, 문의 트리플 모델을 통합적인 테이블에 저장하는 방법, 모든 트리플 모델은 하나의 레코드에 저장되는 방식으로 모델은 간단하지만 자원, 특성의 중복이 심하고 테이블의 크기가 커지는 단점이 있다.

Vertical approach: 자원, 특성, 문을 각각 별도의 테이블에 저장되며 트리플 모델을 표현하는 테이블의 갈림값은 자원, 특성, 문 테이블을 참조하게 된다. 데이터의 중복은 없지만 질의를 처리하기 위해서는 값비싼 조인 연산을 요구한다.

Hybrid approach: 위 두가지 방법을 혼합한 형태로서 자원과 문은 별도의 테이블에 의해 저장되고 트리플 모델은 특성값을 저장하고 자원과 문은 특성값을 저장하는 테이블에서 참조하는 방식이다. 데이터의 중복과 조인 연산은 위 두 가지 방법의 중간이라 할 수 있다[10].

RDF, RDFS(RDF Schema)에 관한 대표적인 질의어

로는 HP Lab.에서 제안된 RDQL과 FORTH에서 제안된 RQL이 있다. RDQL이 RDF문서의 구조적인 특성인 트리플 모델에 기반한 질의어라 하면, RQL은 RDFS에 기반한 의미적인 질의처리를 가능하게 하는 언어이다. 본 논문에서 설계하는 저장 시스템은 RDF에 근간한 것이며 이는 RDQL의 질의 처리를 제공할 수 있음을 의미한다.

3. RDB 저장 모델의 설계

데이터베이스는 데이터를 저장하고 원시 데이터에 질의를 처리하는데 있어서는 뛰어나지만 논리적인 데이터의 추출은 사용자의 몫이다. 데이터베이스에 RDF를 저장하면 RDF의 의미적인 질의와 데이터베이스의 저장 및 질의 처리에 의해 시멘틱 웹을 구축하는데 많은 도움을 줄 수 있을 것이다.

본 장에서는 2장에서 살펴본 RDF에 대해 RDF의 저장 모델중 관계형 데이터베이스 모델을 사용하여 효율적으로 질의를 처리할 수 있는 저장 모델에 대해 기술한다. RDF는 XML로 표현함에도 불구하고 XML이 비교적 구조가 자유로운 반면 RDF는 반드시 subject, predicate, object 의 트리플 모델을 만족하여야 한다는 제약 사항을 가지고 있다. 이러한 이유로 RDF를 저장할 때 일반적인 XML저장고를 이용한다면 RDF 제약사항의 간결함에서 얻을 수 있는 여러 장점들을 이용할 수 없게 된다. 또한 XML의 질의에 이용되는 XPath나 XQuery가 아닌 RDQL 질의 언어를 효과적으로 처리할 수 있도록 저장 시스템의 모델을 제시한다.

3.1 테이블 구조

그림 5는 RDF 문서를 저장하기 위해 본 논문에서 제안하는 데이터베이스의 테이블 모델이다.

rdnamespace: 네임 스페이스를 관리하기 위한 테이블이다. 네임 스페이스는 RDF가 기술되는 영역이다. 즉 이 네임 스페이스로 인해 여러 상이한 영역의 RDF를 저장 할 수 있으며 RDF의 질의가 미치는 기본 영역이기도 하다. 네임 스페이스의 id와 이름인 nsname으로

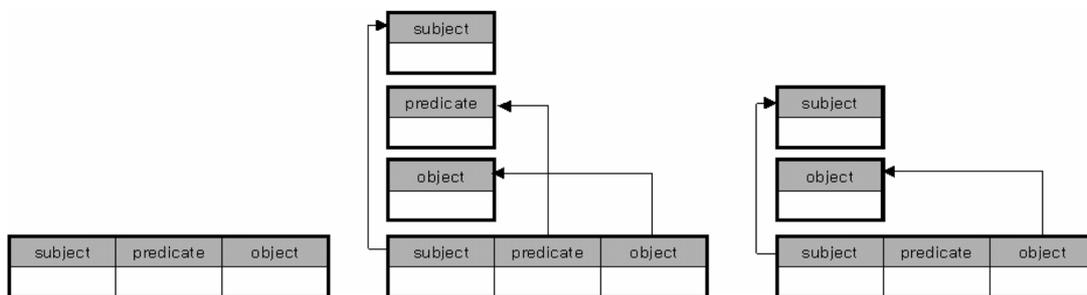


그림 4 Horizontal, Vertical, Hybrid 데이터베이스 테이블 구조

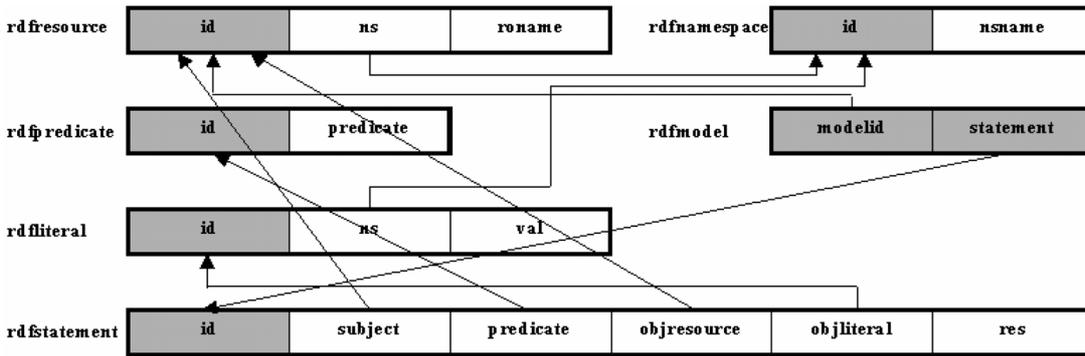


그림 5 RDF 모델을 위한 데이터베이스 테이블 구조

구성되어 있다. 네임 스페이스의 id는 주키이며 색인되어 있다.

rdfsresource: 트리플 모델의 자원을 저장하는 테이블이다. 자원은 RDF 구문에서 주어부에 해당하는 부분으로 URI값으로 표현되어 질 수 있다. 자원은 RDF트리플 구문에서 subject이거나 subject를 기술하는 object 중 다른 트리플 구조에서 subject로 표현되어질 수 있는 자원으로서 테이블에 저장되는 값은 중복을 피하게 된다. 자원의 id, ns와 URI 값인 roname으로 구성되어 있다. 자원의 id는 주키이며 색인되어 있고 ns는 rdfsnamespace의 id를 참조한다.

rdfspredicate: 트리플 모델의 특성값을 저장하는 테이블이다. 특성은 W3C에서 제공하는 특성으로서 주어부의 성격을 결정한다. 특성의 id와 속성값인 predicate으로 구성되어 있다. 특성의 id는 주키이며 색인되어 있다.

rdfsliteral: 트리플 모델의 리터럴을 저장하는 테이블이다. 리터럴은 RDF 트리 구조에서 단말 노드에 해당하는 노드이다. 리터럴의 id, namespace와 리터럴의 값인 val로 구성되어 있다. 리터럴의 id는 주키이며 색인되어 있고 ns는 rdfsnamespace의 id를 참조한다.

rdfsstatement: RDF 구문의 트리플 모델을 저장하는 테이블이다. 트리플 모델은 주어부인 subject와 특성인

predicate, 술어부인 object로 구성되어 있고 object는 자원이거나 리터럴일 수 있다. 트리플 모델의 id, 주어부인 subject, 특성인 predicate, 술어부인 objresource와 objliteral, 그리고 술어부가 자원인지 리터럴 인지의 플래그 값인 res로 구성되어 있다. 트리플 모델의 id는 주키이며 색인되어 있고 subject, predicate, objresource와 objliteral은 각각 rdfsresource, rdfspredicate, rdfsresource와 rdfsliteral의 id를 참조한다.

rdfsmodel: 자원이하의 트리플 모델 구문인 statement를 저장하는 테이블이다. 이는 질의를 할 때 네임 스페이스 전체를 기본으로 하지만 자원이하의 서브 트리에 질의를 가능하게 한다. 자원인 modelid와 statement로 구성되어 있다. 자원의 modelid와 구문 statement는 주키이며 색인되어 있고 modelid와 statement는 각각 rdfsresource와 rdfsstatement의 id를 참조한다.

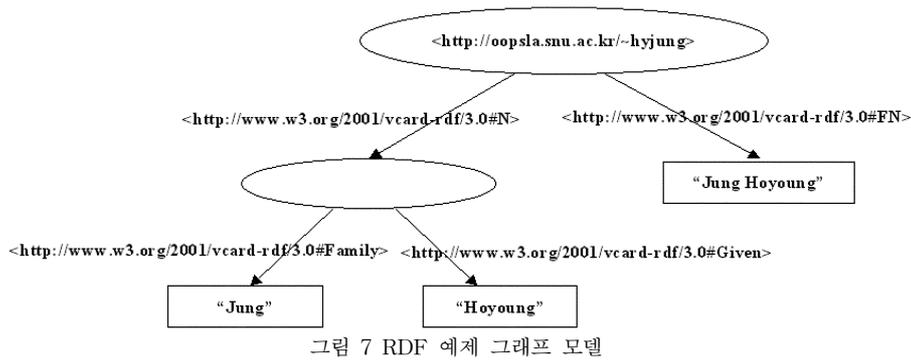
3.2 RDF 문서의 저장 예제

3.1장에서는 RDF를 관계형 모델에 저장하기 위한 테이블 스키마 구조에 대해 알아보았다. 이 장에서는 간단한 RDF 문서를 저장하는 예제에 대해 알아본다.

그림 6은 간단한 예제 RDF 문서를 나타내고 이를 트리 모델로 표현하면 그림 7과 같이 나타낼 수 있다. 그림에서 알 수 있듯이 자원 <http://oopsla.snu.ac.kr/~

```
<?xml version="1.0" encoding="UTF-8" ?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">
    <rdf:Description rdf:about="http://oopsla.snu.ac.kr/~hyjung">
      <vCard:FN>Jung Hoyoung</vCard:FN>
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Jung</vCard:Family>
        <vCard:Given>Hoyoung</vCard:Given>
      </vCard:N>
    </rdf:Description>
  </rdf:RDF>
```

그림 6 RDF 예제 문서



The figure shows six screenshots of Oracle SQL Developer tables representing the storage of RDF data:

- Table: "HYJUNG", RDFNAMESPACE**

| ID | NSNAME |
|----|--------|
| 1 | OOPSLA |
- Table: "HYJUNG", RDFRESOURCE**

| ID | NS | RONAME |
|----|----|---------------------------------|
| 1 | 1 | http://oopsla.snu.ac.kr/~hyjung |
| 2 | 1 | anon:763f5d19420a303c-7ff |
- Table: "HYJUNG", RDFPREDICATE**

| ID | PREDICATE |
|----|---|
| 1 | http://www.w3.org/2001/vcard-rdf/3.0#FN |
| 2 | http://www.w3.org/2001/vcard-rdf/3.0#N |
| 3 | http://www.w3.org/2001/vcard-rdf/3.0#Given |
| 4 | http://www.w3.org/2001/vcard-rdf/3.0#Family |
- Table: "HYJUNG", RDFLITERAL**

| ID | NS | VAL |
|----|----|--------------|
| 1 | 1 | Jung Hoyoung |
| 2 | 1 | Hoyoung |
| 3 | 1 | Jung |
- Table: "HYJUNG", RDFSTATEMENT**

| ID | SUBJECT | PREDICATE | OBJRESOURCE | OBJLITERAL | RES |
|----|---------|-----------|-------------|------------|-----|
| 1 | 1 | 1 | 1 | 2 | 2 |
| 2 | 1 | 2 | 2 | 1 | 1 |
| 3 | 2 | 3 | 2 | 2 | 2 |
| 4 | 2 | 4 | 3 | 2 | 2 |
- Table: "HYJUNG", RDFMODEL**

| MODELID | STATEMENT |
|---------|-----------|
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 2 | 4 |
| 2 | 3 |

그림 8 RDF 예제 문서의 RDF 저장

hyjung>는 특성 <http://www.w3.org/2001/vcard-rdf/3.0#FN>에 대해 문 “Jung Hoyoung”과 특성 <http://www.w3.org/2001/vcard-rdf/3.0#N>에 대한 문 <blank node>를 가진다. <blank node>는 특성 <http://www.w3.org/2001/vcard-rdf/3.0#Family>에 대해 문 “Jung”과 <http://www.w3.org/2001/vcard-rdf/3.0#Given>에 대해 문 “Hoyoung”을 가진다. 여기에서 자원은 <http://oopsla.snu.ac.kr/~hyjung>, <blank node>이고 특성은 <http://www.w3.org/2001/vcard-rdf/3.0#FN>, <http://www.w3.org/2001/vcard-rdf/3.0#N>, <http://www.w3.org/2001/vcard-rdf/3.0#Family>, <http://www.w3.org/2001/vcard-rdf/3.0#Given>이며 리터럴은 “Jung Hoyoung”, “Jung”, “Hoyoung”이다. 이 RDF 예제의 네임 스페이스를 “OOPSLA”라고 하고 3.1장의 테이블 구조를 이용하여 저장하면 그림 8과 같이 저장된다.

4. RDQL 질의어 처리기

RDQL은 HP Lab.에서 개발된 RDF에 대한 질의 언어이다. RDQL이 비록 RQL과 같이 RDFS에 기반하여 의미적인 질의를 제공하지는 못한다 할지라도 RDF문에 대한 질의어로서 표준으로 잡았다. 본 장에서는 RDQL에 대해 알아보고 3장에서 제안한 RDF의 저장 시스템에서 RDQL질의어 처리기를 제공한다. RDQL은 SQL과 비슷한 “SELECT”, “FROM”, “WHERE”, “AND”, “USING”절로 이루어져 있고 변수는 ‘?’로 시작한다. RDQL에서 사용하는 키워드는 다음과 같다[8].

SELECT: 추출하고자 하는 자원, 특성, 문의 변수를 사용한다.

FROM: URI로 특정 모델로부터 추출할 수 있다.

WHERE: 조건이 되는 트리플 패턴을 구체화 한다.

AND: 블리언 표현을 사용할 수 있다.

USING: URI를 사용할 때 지정해 놓으면 매치되는 URI에 대해 지정된 짧은 표현을 사용할 수 있다.

4.1 질의 처리의 예

그림 6,7 참조

예제 1. 자원 추출 : 자원이 <http://oopsla.snu.ac.kr/~hyjung>이고 특성이 <http://www.w3.org/2001/vcard-rdf/3.0#N>인 자원 추출

```
SELECT ?name
WHERE (<http://oopsla.snu.ac.kr/~hyjung>,
<http://www.w3.org/2001/vcard-rdf/3.0#N>, ?name")
```

name

```
=====
<anon:763f5d:f9420a303c:-7fff>
```

예제 2. 자원 추출 : 특성 <http://www.w3.org/2001/vcard-rdf/3.0#FN>에 대해 리터럴이 “Jung Hoyoung”인 자원 추출

```
SELECT ?x
WHERE (?x, <http://www.w3.org/2001/vcard-rdf/3.0#FN>, “Jung Hoyoung”)
```

```
x
=====
<http://oopsla.snu.ac.kr/~hyjung>
```

예제 3. 자원 리터럴 추출 : 특성 <http://www.w3.org/2001/vcard-rdf/3.0#FN>에 대해 자원과 리터럴 추출

```
SELECT ?x, ?fname
WHERE (?x, <http://www.w3.org/2001/vcard-rdf/3.0#FN>, ?fname)
```

```
x | fname
=====
<http://oopsla.snu.ac.kr/~hyjung> | “Jung Hoyoung”
```

예제 4. 리터럴 추출 : 특성 <http://www.w3.org/2001/vcard-rdf/3.0#Family>가 “Jung”인 자원의 특성 <http://www.w3.org/2001/vcard-rdf/3.0#Given>의 리터럴을 추출 (조인 연산)

```
SELECT ?givenName
WHERE (?x, <http://www.w3.org/2001/vcard-rdf/3.0#Family>, “Jung”),
(?x, <http://www.w3.org/2001/vcard-rdf/3.0#Given>, ?givenName)
```

```
givenName
=====
“Hoyoung”
```

4.2 구현

RDQL은 RDF에 질의를 처리할 수 있는 질의어로서 기존의 연구는 RDF에 질의를 하기 위해서는 파일이든 데이터베이스의 Large Object객체를 메모리 객체로 표현하여 이를 통해 원하는 결과를 추출할 수 있었다. 이는 RDF에 질의를 하기 위해서는 모든 문서를 메모리에 올려야 하는데 대용량의 문서에 대해서는 쉽지 않음을 알 수 있다. 본 장에서는 본 논문에서 제안한 RDF 저

장 시스템을 이용하여 사용자의 질의 요구를 데이터베이스 레벨에서 추출할 수 있는 질의 처리기를 제안한다.

RDQL은 4.1장에서 살펴본 바와 같이 SQL과 유사하며 기본적으로 “WHERE”절에서 RDF문의 트리플 구조인(<자원>, <특성>, <문>|“문”)의 조건을 통해 ‘SELECT’ 질의 변수를 추출하는 구조이다. 질의를 처리하기 위해서는 “WHERE” 절의 조건을 RDF 저장 시스템에 적합한 SQL로 변환 작업이 필수라 할 수 있다.

추출의 조건이 되는 RDF는 object에 해당하는 문이 <자원>이거나 “리터럴”일 수 있으며 이 트리플 statement는 RDF 저장 시스템에 rdfstatement 테이블에 존재한다. 따라서 RDQL의 질의를 데이터베이스 질의 언어로 처리하기 위해서는 object가 <자원>인 경우와 “리터럴”인 경우로 나누어 질의를 처리한다.

case 1. (<자원>, <특성>, <문>) 예제 1, 그림 6,7 참조

RDQL 질의 :

```
SELECT ?name
WHERE (<http://oopsla.snu.ac.kr/~hyjung>,
<http://www.w3.org/2001/vcard-rdf/3.0#N>, ?name”)
```

name

```
=====
<anon:763f5d:f9420a303c:-7fff>
```

SQL 질의 : RDF의 네임 스페이스를 ‘OOPSLA’라 하면

```
SELECT r1.roname
FROM rdfnamespace n,
     rdfresource r0,
     rdfpredicate p0,
     rdfresource r1,
     rdfstatement s0
WHERE n.id = r0.ns and
      n.id = r1. ns and
      r0.id <> r1.id and
      n. nsname = ‘OOPSLA’ and
      r0.id = s0.subject and
      r0.roname = ‘http://oopsla.snu.ac.kr/
~hyjung’ and
      p0.id = s0.predicate and
      p0.predicate = ‘http://www.w3.org/
2001/vcard-rdf/3.0#N’ and
      r1.id = s0.objresource and
      s0.res = 1;
```

RONAME

```
-----
anon:763f5d:f9420a303c:-7fff
```

case 2. (<자원>, <특성>, “문”) 예제 2, 그림 6,7 참조
RDQL 질의 :

```
SELECT ?x
WHERE (?x, <http://www.w3.org/2001/vcard-rdf/
3.0#FN>, “Jung Hoyoung”)
```

x

```
=====
<http://oopsla.snu.ac.kr/~hyjung>
```

SQL 질의 : RDF의 네임 스페이스를 ‘OOPSLA’ 라 하면

```
SELECT r0.roname
FROM rdfnamespace n,
     rdfresource r0,
     rdfpredicate p0,
     rdfliteral l0,
     rdfstatement s0
WHERE n.id = l0.ns and
      n.nsname = ‘OOPSLA’ and
      r0.id = s0.subject and
      p0.id = s0.predicate and
      p0.predicate = ‘http://www.w3.org/2001/
vcard-rdf/3.0#FN’ and
      l0.id = s0.objliteral and
      l0.val = ‘Jung Hoyoung’ and
      s0.res = 2;
```

RONAME

```
-----
http://oopsla.snu.ac.kr/~hyjung
```

5. 결론 및 향후 연구 과제

본 논문에서는 웹 자원에 대한 지식구조를 기술하는 RDF에 대한 관계형 데이터베이스 저장 모델과 데이터베이스를 이용하여 RDQL 질의 처리를 수행할 수 있는 시스템에 제안하였다. 이는 시멘틱 웹이 발전함에 따라 증가할 수 있는 메타 데이터를 효율적으로 저장, 관리할 수 있으며 특히 다량의 지식기반 문서관리에 유용하다. RDF에 대한 관리 및 질의를 메모리 객체가 아닌 데이터베이스에서 지원을 함에 따라 시멘틱 웹에서 사용할 수 있는 웹 자원, 전자 문서, 오프라인 서적에 이르는 특히 대용량의 콘텐츠 관리 시스템에 대해 효율적

인 질의 처리에 도움이 될 것으로 보인다. RDQL은 RDF문서에 대해 구조적인 질의에 제한되는데 이는 질의를 할 때 RDFS를 사용하기 힘들다는데 있다. RDF(S)에 대해 범용적인 저장 시스템을 설계한다면 RQL에 대한 지원이 가능하며 이는 향후 연구과제로 해결할 문제이다.

참 고 문 헌

- [1] Bebee, B.R.; Mack, G.A., "Distributed meta data objects using RDF," Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999. (WETICE '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] NilsKlarlund, Anders Moller and Michael I. Schwartzbach, "DSD: a schema language for XML," Proceedings of FMSP'00 The third workshop on Formal methods in software practice, 2000.
- [5] S.Alexaki, V.Christophides, G.Karvounarakis, D.Pleousakis &K.Tolle, "On Storing Voluminous RDF Description: The case of Web Portal Catalogs," In Proc. of WebDB2001 in conjunction with ACM SIGMOD'01 Conference, 2001.
- [6] D.Florescu and D.Kossmann, "A performance evaluation of alternative Ontology Editor for the Semantic Web," In Proceeding of KI2001.
- [7] Steve Pepper ; Graham Moore ; XML Topic Maps (XTM) 1.0, TopicMaps.org.
- [8] <http://www.hpl.hp.com/semweb/>
- [9] James Hendler. "Agents and the Semantic Web," IEEE Intelligent Systems Journal, March/April 2001.
- [10] 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.



김 정 민

1992년 홍익대학교 전자계산학과 졸업(학사). 1994년 홍익대학교 전자계산학과 졸업(석사). 2002년 서울대학교 전기, 컴퓨터공학부 박사과정 수료. 관심분야는 Semantic Web, Ontology, IR, Database



정 준 원

2000년 동국대학교 컴퓨터공학과(학사) 2003년 서울대학교 전기.컴퓨터공학부(석사). 2003년~현재 서울대학교 전기.컴퓨터공학부 박사과정 재학 중. 관심분야는 시멘틱웹, 온톨로지, XML, 데이터베이스



김 중 남

2003년 서울대학교 컴퓨터공학부(학사) 005년 서울대학교 전기.컴퓨터공학부(석사). 심분야는 데이터베이스, 시멘틱웹, 온톨로지, XML



임 동 혁

2003년 고려대학교 컴퓨터교육과(학사) 2005년 서울대학교 전기.컴퓨터공학부(석사). 2005년~현재 서울대학교 전기.컴퓨터공학부 박사과정 재학 중. 관심분야는 데이터베이스, XML, 시멘틱웹, 온톨로지

김 형 주

정보과학회논문지 : 데이터베이스
제 33 권 제 1 호 참조



정 호 영

2000년 한국외국어대학교 컴퓨터공학과(학사). 2002년 한국외국어대학교 컴퓨터공학과(석사). 2003년~현재 서울대학교 전기.컴퓨터공학부 박사과정 재학 중. 관심분야는 시멘틱웹, 온톨로지, XML, 데이터베이스