

# RDF 기반의 온톨로지 처리시스템

## (A RDF based Ontology Management System)

정준원<sup>†</sup>   정호영<sup>†</sup>   김종남<sup>†</sup>   임동혁<sup>†</sup>   김형주<sup>\*\*</sup>  
 (Jun-Won Jung) (Ho-Young Jung) (Jong-Nam Kim) (Dong-Hyuk Lim) (Hyoung-Joo Kim)

**요약** 수많은 종류의 데이터가 발생하는 현재와 같은 상황에서 데이터의 처리속도가 발전함에 따라 원하는 정보를 얻기는 더욱 어려워지고 있다. 이와 같은 상황에서는 데이터를 얼마나 빠르게 처리하는지 보다 얼마나 의미 있는 정보를 얻을 수 있느냐가 중요한 문제이다. 시맨틱웹은 의미적인 정보와 관계를 기술하여 지능적인 정보 처리를 지원하며 이를 통해 온톨로지를 구축할 수 있다. 하지만 현재 실질적인 서비스를 고려한 온톨로지 관리 시스템은 부족한 상황이다. 본 논문에서는 온톨로지를 저장, 관리, 검색하는 기본적인 기능은 물론 실질적인 온톨로지 서비스를 수행할 수 있도록 효율적인 데이터 처리에 중점을 두고 온톨로지 관리 시스템을 구현하였다. 먼저 RDF 데이터 모델이 저장장치에 독립적인 인터페이스를 가지도록 시스템을 설계하고 RDF 데이터를 데이터베이스에 효율적으로 저장, 처리하는 기법을 제안하였다. 연관된 데이터의 빠른 처리를 지원하기 위한 스키마와 시맨틱 정보 기반의 캐시기법을 제안하였으며 사용자 중심의 온톨로지 검색 인터페이스를 제안하였다. 그리고 실제 온톨로지 데이터를 시스템에 적용해 봄으로써 구축된 시스템이 효율적으로 동작함을 확인하였다.

**키워드** : 온톨로지, 시맨틱웹, 온톨로지 처리시스템

**Abstract** Computing has been making a lot of progress in the quantity of data, today. It's going to be more difficult to get appropriate information as the number of data increases dramatically. Therefore it's more important to get meaningful information than to focus on the speed of processing. Semantic web enables an intelligent processing by adding semantic information to data, and it is useful to make ontology system. In this paper, we implemented ontology processing system which support function for ontology and efficient processing for practical service. We proposed system design for independent from storage, storing technique for RDB, caching technique by schema information and useful user interface.

**Key words** : Ontology, SemanticWeb, Ontology processing system

### 1. 서론

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 DB와 같이 데이터 자체를 구조화시켜 저장하거나 XML 처럼 메타데이터를 통해 데이터의 구조를 부여하

고 이를 통해 효율적으로 데이터를 처리하는 방식들이 있다. 이와 같이 데이터에 구조를 부여해서 효율적으로 처리하는 것과는 또 다른 접근방법으로서 정보에 대한 의미적인 메타데이터를 기술함으로써 효과적인 처리를 도모하고자 하는 시맨틱웹과 같은 방법이 있다[1]. 시맨틱웹은 원래 W3C(World Wide Web Consortium)에서 웹 데이터에 대한 의미 확장을 위해 개발이 시작되었으나 최근에는 다양한 응용에 사용될 수 있는 온톨로지로서 함께 연구되고 있다. 시맨틱웹은 데이터에 의미 정보와 연관관계를 기술함으로써 지능적인 처리나 고급 정보의 처리가 가능하다. 그리고 이와 같은 정보 기술능력 은 온톨로지를 구축하는데 적합하다[2,3].

온톨로지란 초기에 철학분야에서 개념을 표현하기 위해 사용되는 여러 용어의 집합을 의미하거나 특정 분야에서 사용되는 용어의 집합을 의미하기도 하였다. 일반적인 의미에서의 온톨로지란 작게는 용어의 집합에서부

· 본 연구는 BK21 정보기술 사업단과 정보통신부ITRC(e-Business기술 연구센터)에 의해 지원되었음

<sup>†</sup> 학생회원 : 서울대학교 전기.컴퓨터공학부  
 jwjung@oopsla.snu.ac.kr  
 hyjung@oopsla.snu.ac.kr  
 jnkim@oopsla.snu.ac.kr  
 dhlim@oopsla.snu.ac.kr

<sup>\*\*</sup> 종신회원 : 서울대학교 전기.컴퓨터공학부 교수  
 hjk@oopsla.snu.ac.kr

논문접수 : 2004년 11월 10일  
 심사완료 : 2005년 3월 28일

터 크케는 용어에 대한 부가정보나 관계, 색인 등을 표현한 정보라고 할 수 있다. 결국 시멘틱웹도 온톨로지에 속하며 최근에 W3C에서 시멘틱웹의 일종인 RDF(Resource Description Framework)[4]를 확장한 OWL(Web Ontology Language)[5]과 ISO(International Organization for Standardization)에서 TopicMap[6] 같은 온톨로지를 기술하는 표준을 제정함으로써 데이터의 의미나 관계를 이용해 처리하는 활발한 연구가 진행되고 있다. 본 논문에서는 온톨로지를 기술하는데 있어 가장 기본적인 언어인 RDF를 사용하며 이에 대한 자세한 내용은 2장의 배경지식에서 자세히 살펴보겠다.

온톨로지 정보를 처리하는데 있어 가장 기본적인 것은 바로 온톨로지를 생성, 편집, 저장, 검색할 수 있는 온톨로지 처리시스템이다[2]. 이와 관련된 시스템이 있기는 하지만 대부분의 경우 대용량의 데이터를 실제 서비스 환경에서 처리할 수 있는 정도의 처리능력을 지원하지 못한다. 이것은 온톨로지의 표준은 잘 구현하고 있지만 아직 온톨로지의 효율적인 데이터 처리에 대한 측면은 고려하지 못하기 때문이다. 따라서 본 논문에서는 효율적인 실제 서비스를 고려한 온톨로지 처리시스템을 설계하고 각 부분에 대한 효율적인 처리 기법을 제안한다. 먼저 RDF 데이터 처리, 질의처리, 캐쉬, 저장 및 외부 연결과의 독립을 지원하기 위한 인터페이스와 같이 온톨로지 처리를 위한 기능별로 시스템의 각 컴포넌트를 설계한다. 이와 같이 구조가 정해진 다음에 처리의 속도나 효율에 가장 연관이 많은 부분은 RDF 데이터 모델을 처리하고 저장하는 방법이다. RDF 데이터에 대한 가장 일반적인 정보 검색은 데이터의 구조를 통해 접근되는 것이 아니라 연관정보를 통한 검색 혹은 의미적 관계에 기반한 검색이다. 따라서 이 부분의 효율을 높이기 위해 온톨로지의 데이터에 대한 접근이 의미적 정보와 관계가 많고 이는 곧 스키마의 구조와 연관이 많다는 점에 착안한 캐쉬기법을 제안하였다. 또한 RDF의 데이터모델인 트리플모델은 메모리나 관계형데이터베이스에 저장하는데 있어서 연관 정보를 검색할 경우 반복된 처리나 조인이 빈번히 발생할 수 있다[7,8]. 특히 관계형데이터베이스에서 연관 정보 검색을 위한 데이터를 처리할 경우 빈번한 조인은 큰 성능 저하를 발생시킨다. 따라서 RDF를 관계형데이터베이스에 저장할 때 정보의 중복을 최소화하고 확장성을 가지도록 저장소를 설계한 후 조인으로 인한 효율성의 저하를 캐쉬를 통해 극복하도록 하였다. 그리고 마지막으로 사용자가 온톨로지를 검색하는데 있어 필요에 따라 유용한 검색 방법이 다를 수 있다. 따라서 다양한 검색을 지원할 수 있도록 그래프, 트리, 키워드 질의를 혼합한 인터페이스를 제안하며 RDF로 제공되는 부가정보를 처리한 통계기능을

제공하도록 한다. 이를 통해 온톨로지에서 기술되는 정보들이 의미적인 검색에서뿐만 아니라 부가정보를 통한 고급 지식을 획득할 수 있음을 확인해 본다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 다른 온톨로지 관리시스템이나 온톨로지 처리와 관련된 연구들에 대해서 살펴본다. 그리고 3장에서는 본 연구에서 온톨로지 기술언어로 사용하는 RDF의 데이터 모델에 대해 설명하며 4장에서는 본 논문에서 구현한 온톨로지 관리시스템의 구조와 각 컴포넌트들에 대한 설명, 그리고 제안된 기법에 대해서 설명한다. 5장에서는 이와 같은 과정을 통해 구현된 시스템의 구현 결과를 살펴보고 다른 시스템과 비교해 본다. 마지막으로 6장에서는 결론을 정리하고 향후 연구에 대해 소개한다.

## 2. 관련 연구

온톨로지를 기술하기 위한 언어로서 W3C에서 제정한 시멘틱웹의 일종인 RDF와, RDF의 확장으로서 웹 온톨로지 구축을 위한 OWL, ISO에서 제정한 TopicMap, DARPA(Defense Advanced Research Projects Agency)에서 제정한 DAML+OIL[9] 등이 있다. 이와 같은 온톨로지 기술 언어들은 표현 방법은 다르지만 모델이나 표현력에서 많은 유사성을 가지고 있다. 특히 RDF와 DAML+OIL의 주요 특성들이 OWL에 적용되었다. 본 논문에서는 RDF로 온톨로지를 구축하는데 그 이유는 RDF가 시멘틱웹이라는 이름으로 제정되었지만 웹 데이터뿐만 아니라 일반적인 온톨로지 구축에 있어 주요한 기능들로 이루어져 있기 때문이다. 특히 RDF는 OWL이나 DAML+OIL에서도 기본 골격을 이루고 있으며, 따라서 추후 OWL과 같이 더욱 다양한 내용을 기술하는 온톨로지로서 확장이 용이하다.

RDF데이터를 지원하는 대표적인 온톨로지 관리시스템으로서 Ontoprise사의 OntoEdit[10], FZI(Forschungszentrum informatik)의 KAON(Karlsruhe Ontology)[11], Stanford Medical Informatics의 Protégé-2000[12]이 있다. 이 시스템들은 온톨로지를 구축하고 편집, 및 검색을 지원하는 시스템들이다. 각 시스템 별 특징을 살펴보면 OntoEdit는 온톨로지를 데이터를 정의하고 구축하는 방법론인 온톨로지 공학의 각 단계를 구현한 시스템으로서 온톨로지를 설계, 구축하는 것에 적합한 시스템이다. 따라서 그래프 관계를 표현하는 사용자 인터페이스를 지원하지만 이는 설계를 지원하기 위함이며, 인터페이스가 매우 복잡하기 때문에 일반 사용자를 위한 서비스로서는 부적절하다. KAON은 비교적 잘 모듈화 된 시스템으로서 상위 어플리케이션을 지원하기 위한 상위 API, 저장과 검색을 위한 하위 API를 지원하고 있다. 하지만 RDF의 저장기법에 대해서는 깊

이 고려되지 않고 있으며 제공되는 검색 인터페이스는 온톨로지 편집에 중점을 두어 구조를 살펴보는 기본적인 브라우저만 지원된다. Protégé-2000은 의학정보에 대한 온톨로지를 구축하기 위해 시작된 시스템으로서 온톨로지를 구축하고 검색하는 것에 중점을 둔 시스템이다. 따라서 그래프적인 편집이나 검색은 잘 지원하지만 처리의 효율성 보다는 온톨로지 정보의 획득에 중점을 둔 시스템이기 때문에 효율적 처리에 대한 고려가 부족하다. 또한 시스템이라기보다는 플러그 인으로 확장되는 프레임워크에 가깝다.

위의 시스템들은 RDF 규격에 기반하여 온톨로지 생성 및 편집, 검색과 같은 기본적인 온톨로지 관리기능을 지원하지만 효율적 처리에 있어서는 많이 부족한 상태이다[13]. 예를 들어 데이터를 저장하는데 있어 XML이나 파일 형태의 자료만을 지원한다면 관계형데이터베이스와 같은 데이터베이스에 저장하는 경우에도 RDF 데이터의 트리플 모델의 특징으로 인한 연관 관계 검색에서의 빈번한 조인에 대해 잘 고려하지 못해 성능이 저하되거나 캐쉬에 대한 부재로 빈번한 DB접근이 발생한다. 또한 각 시스템이 실질적인 서비스보다는 온톨로지 생성이나 관리 중심이어서 사용자를 위한 인터페이스로서는 부족하다. 또한 시스템 구조에 있어서 일부 시스템이 확장을 고려한 인터페이스를 지원하기도 하지만 로컬서비스, 웹서비스와 같은 다양한 서비스를 고려하지 않거나 저장소 독립적 인터페이스와 같은 모듈화가 미미한 경우도 있다.

Sesame[14]는 RDF를 데이터베이스에 저장하고 질의 및 추론을 수행하는 기능을 제공하는 프레임워크로서 Aduna사에서개발하여 오픈소스로 제공하고 있다. Sesame는 각 기능별로 모듈화가 잘 되어있고 저장장치 독립성을 지원하며 데이터베이스 지원, RQL, RDQL과 같은 질의 및 각 기능에 대한 API 를 지원하고 있기 때문에 앞서 소개한 다른 시스템에 비해 많은 기능을 제공하지만 관계형데이터베이스에 RDF를 저장하고 관리하는데 있어 의미적인 연결관계를 고려한 처리는 고려하지 않고 있다. 또한 저장소의 설계에 있어 RDF와 RDF 스키마만을 지원하고 OWL에 대한 확장을 고려하지 않았으며 데이터 검색을 위한 사용자 인터페이스가 구현되어있지 않다.

RDF의 의미적인 캐쉬와 관련해서는 직접적으로 연구된 바가 아직 없다. 비교가 될 수 있는 연구로서 DB분야의 시멘틱 캐쉬기법[15]은 DB정보를 전송할 때 관계를 고려해서 차집합만을 제공하는 기법으로 데이터 자체의 의미정보를 고려하는 것과는 거리가 있다. 또한 웹 캐쉬기법에서 페이지와 링크를 노드와 간선으로 보고 그래프로 모델링 한 뒤 캐쉬하는 기법[16]이 있으나 이

것은 그래프 구조만 고려한 기법이기 때문에 마찬가지로 데이터의 의미정보를 고려하지 못하고 있다.

RDF 저장기법과 관련한 연구에 의하면 크게 수평모델과 수직모델로 나눌 수 있다. 수평모델은 RDF트리플의 요소를 통합테이블에 저장하므로 모든 트리플이 하나의 레코드로 저장된다. 이 경우 모델은 간단하지만 데이터의 중복이 심하고 테이블의 크기가 커지게 된다. 수직모델의 경우는 트리플의 요소가 각기 다른 테이블에 저장되며 잘못 설계할 경우 많은 조인이 발생하게 된다[7].

인터페이스와 관련해서는 그래프와 트리형태의 탐색, 그리고 텍스트 기반의 탐색에 대한연구가 있다[17]. 그래프 기반의 탐색은 정보의 연관성은 잘 표현하지만 대부분의 경우 실제 정보를 표현할 경우 수 많은 노드로 인해 내용을 파악하기가 힘들다. 트리 형태의 탐색은 계층 관계를 잘 분류해 주지만 연관 관계를 파악하기 힘들거나 트리의 깊이가 깊어질 경우 탐색이 어렵다. 키워드 검색의 경우 목적에 맞는 정보를 빠르게 찾을 수 있으나 검색된 정보간의 관계를 알기 어렵다. 본 논문에서는 이와 같은 부분들을 모두 고려한 시스템을 구현하고 있다.

### 3. RDF 데이터 모델

본 논문에서는 RDF를 사용하여 온톨로지를 구축한다. RDF는 W3C에서 제정한 것으로서 기술하고자 하는 대상에 대한 부가정보, 데이터간의 상하 및 연관 관계 등을 기술하는 능력을 가진다. 즉 데이터를 정의하고 그 데이터에 대한 설명이나 관계를 기술함으로써 온톨로지를 구축할 수 있는 방법을 제공한다.

RDF는 기본적으로 주어(subject), 서술(predicate), 목적(object)의 트리플(triple)모델로 기술된다. 주어란 표현하고자 하는 데이터를 의미하며, 서술은 주어에 대해 기술하거나 주어와 목적의 관계를 의미한다. 목적이란 서술에 대한 내용이나 값을 의미한다. 또한 각 내용들에 대해서 URI를 통해 기술할 수 있다. 예를 들어 "JACK이 Ontology Design이라는 제목의 논문을 작성했다."는 내용을 기술한다면 각 내용은 다음 표 1과 같이 대응된다.

표 1 RDF 트리플 표현

주어(subject)	http://www.sample.com/JACK
서술(predicate)	http://www.sample.com/term/creation
목적(object)	"Ontology Design"

위의 표 1의 모델은 다시 RDF 모델을 기술하는 '트리플 그래프모델'로 표현할 수 있다. 그래프 모델에서 주어는 원으로, 서술은 간선으로, 목적은 URI일 경우 원, 리터럴(literal)일 경우 사각형으로 표시된다. 다음 그림 1

은 표 1의 내용을 그래프로 나타내고 있다.

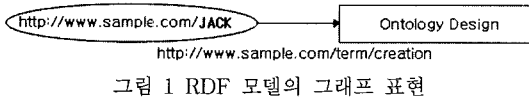


그림 1 RDF 모델의 그래프 표현

RDF는 이와 같은 트리플을 연속적으로 기술함으로써 정보를 표현한다. RDF에서는 이와 같이 실제 데이터를 기술하는 것 뿐만 아니라 데이터에서 사용되는 용어의 종류와 용어간의 관계를 기술하는 스키마인 RDF 스키마(RDF Schema)를 지원한다. 다음 그림 2의 윗부분은 RDF 스키마를, 아래 부분은 이에 대한 인스턴스인 RDF를 나타낸다. 아래 그림 2에서는 표현을 간단히 하기 위해 URI는 제외하고 나타내었다.

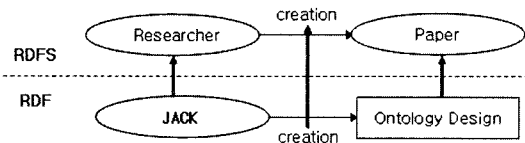


그림 2 RDF와 RDF스키마

위의 그림에서 보면 그림 1이 예제의 내용에 RDF 스키마가 추가되어서 JACK이라는 사람이 'researcher'라는 타입을 가지고 'Ontology Design'이라는 논문이 'paper' 타입을 가진다. 또한 RDF의 서술 'creation'이 RDF 스키마에서 'researcher'와 'paper'의 관계에 있는 'creation'의 인스턴스임을 RDF스키마를 통해 나타내고 있다. 이와 같이 RDF는 RDF인스턴스와 RDF스키마의 구성요소를 이용해 정보들과 그 관계를 자유롭게 기술할 수 있다.

#### 4. 온톨로지 관리 시스템(Pinakes)의 설계와 구현

본 연구에서 구축하는 온톨로지 관리 시스템은 고대의 문헌목록인 'Pinakes'의 이름을 따와 근원적인 온톨로지라는 의미에서 Pinakes라고 명명하였다. 본 장에서는 본 시스템의 설계와 상세 설명, 그리고 성능향상을 위해 제안된 기법들에 대해서 설명하도록 한다.

##### 4.1 시스템 구조

Pinakes 시스템은 먼저 다양한 서비스와 확장을 고려하기 위해 각 기능에 대한 컴포넌트를 나누고 다양한 확장을 위한 인터페이스를 지원하도록 설계하였다. 다음 그림 3은 Pinakes시스템의 구조를 보여주고 있다.

시스템은 크게 웹 어플리케이션이나 사용자 서비스를 위한 서비스, 온톨로지 데이터를 처리하고 관리하는

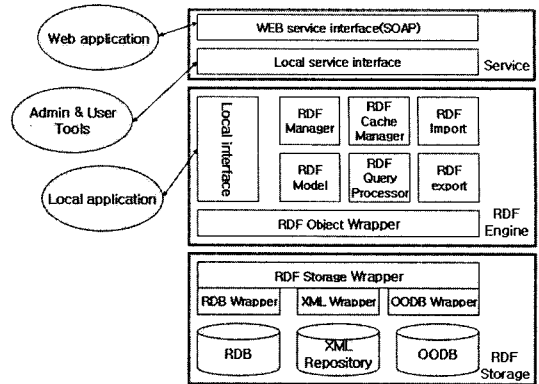


그림 3 Pinakes 시스템의 RDF 온톨로지를 위한 구조

RDF 엔진(RDF Engine), 영속적 저장소에 데이터 저장을 지원하는 RDF 저장소(RDF storage)로 나뉘어 있다. 이와 같은 구성은 서비스와, 처리, 저장을 독립시켜 다양한 확장과 응용을 지원하기 위함이다. 본 장에서는 검색의 편의성을 제공하기 위한 서비스, 처리에 대한 효율성을 고려하는 RDF 엔진, RDF 저장소로 나누어 각 부분을 살펴본다. 먼저 4.2절에서 시스템의 중심을 이루고 있는 RDF 엔진에 대해 설명하고, 4.3절에서는 RDF 엔진 중에서도 효율적인 처리를 위해 중요한 역할을 하는 RDF 캐쉬에 대해서 설명한다. RDF 캐쉬는 RDF 엔진에 포함되지만 큰 비중을 차지하므로 RDF 엔진과 별도의 절로 설명한다. 4.4절에서는 RDF 저장소에 대해 설명하며, 마지막으로 4.5절에서 이를 기반으로 처리한 결과를 사용자에게 제공하는 서비스에 대해서 설명한다.

##### 4.2 RDF 엔진

RDF 엔진은 RDF 데이터에 대한 처리와 메모리나 저장소에 있는 데이터를 다루는 기능을 수행한다. RDF Model은 RDF 트리플 모델의 데이터를 메모리상에서 다루기 위한 오브젝트와 메소드들을 의미한다. 처음 파일형태의 RDF 데이터가 메모리상에 RDF Model로 올라오는 과정을 통해 RDF 엔진의 컴포넌트들을 살펴볼 것이다. 먼저 RDF import는 XML형태의 RDF파일을 입력으로 받아 RDF Model로 생성하는 작업을 수행한다. 먼저 RDF파일이 입력으로 들어오면 SAX파싱 과정을 거쳐 구문적 검사를 수행한 후 이를 RDF파서 객체에게 전달한다. RDF파서는 RDF문서의 내용을 분석하여 주제, 서술, 목적과 같은 요소를 분석하고 RDFS를 참조해 각 용어와 관계를 분석 및 추출한다. 다음 이 정보를 RDF Model의 객체에 전달함으로써 RDF모델 형태의 데이터가 메모리에 상주하게 된다. RDF Manager는 RDF Model에서 지원하지 못하는 고급 처리를 담당하는 것으로서 특정 키워드를 가지는 요소에 대한 트리플

을 추출하거나 트리플의 연결을 따라 연관정보를 가져 온다든가, 두 트리플의 관계를 추출해 낸다든가 하는 기능을 수행한다. 이러한 동작은 스키마에 대해서도 함께 수행되며 이와 같이 고급기능들에 대한 처리들을 담은 것이 RDF Manager이다. 그리고 이와 같은 RDF model의 내용은 RDF export를 통해 XML형식의 RDF 파일로 저장할 수 있다. RDF Query Processor는 본 시스템에서 제공하는 키워드 기반 질의를 지원하기 위한 것으로서 RDQL형태의 질의를 처리하도록 한다. RDQL[18]이란 RDF 질의언어 중 하나로써 트리플 모델의 형식으로 질의를 수행한다. 예를 들어 "A라는 주어와 B라는 서술관계를 가지는 목적들을 찾아라"와 같은 식의질을 지원한다. Pinakes는 효율적인 처리를 위해 RDF cache manager를 통해 캐쉬를 지원하고 있는데 이에 관한 자세한 내용은 다음 절에서 살펴보도록 한다. Local interface는 각 컴포넌트가 제공하는 기능을 다양한 어플리케이션에서 사용할 수 있도록 제공하는 인터페이스이다. 따라서 Pinakes는 다양한 어플리케이션에 사용될 수 있다. 마지막으로 RDF 오브젝트 래퍼는 RDF 모델의 구현과 독립적으로 시멘틱웹을 처리하기 위한 일관된 인터페이스를 제공한다. 따라서 추후 OWL과 같은 확장이나 RDF 표준에 완전히 일치하지 않는 형식의 온톨로지를 지원할 수 있다.

4.3 RDF 캐쉬

본 논문에서 구현하는 온톨로지 관리시스템인 Pinakes는 메모리에 모든 데이터를 상주시키고 처리하는 것이 아니라 관계형데이터베이스에 데이터를 저장한 후 처리한다. 이때 트리플 데이터의 연결에 따라 데이터를 접근하고 처리할 경우 각 트리플 요소를 데이터베이스에서 가져오는 것은 비효율적이다. 또한 RDF 데이터에 대한 접근은 대부분 서술과 관련된 연관 관계에 의한 접근이나 의미적 관계에 의한 접근으로 이루어진다. 따라서 RDF가 가지고 있는 메타데이터의 내용이나 연관관계를 통해 의미적 관계를 가지는 캐쉬를 생성함으로써 효율적인 캐쉬를 지원하고자 하였다. 다른 분야에서의 데이터 접근이 시간적 지역성에 기반하므로 캐쉬가주로 접근빈도를 기반으로 구축되는 반면 RDF 데이터에 대한 참조는 시간적 지역성 보다는 연관관계에 의한 접근이 더 주를 이루기 때문이다. 다음의 그림 4는 RDF의 의미적 정보에 기반한 캐쉬의 필요성과 방법을 설명하기 위한 간단한 예제이다.

그림에서 'JACK'이라는 학생은 'Ontology Design'이라는 논문을 작성하였고, 'BOB'이라는 교수는 'Ontology caching'이라는 논문을 작성했다는 내용이 RDF로 표현되어 있다. 이때 RDF 데이터만 관찰했을 경우 두 개의 트리플 사이에는 연관 관계가 없다. 하지만 RDF스키마를

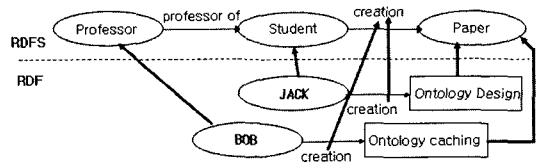


그림 4 RDF와 연관 관계들

함께 관찰하게 되면 'BOB'이 RDF 스키마의 'Professor'에 대한 인스턴스이고, 'JACK'이 'Student'의 인스턴스이므로 'BOB'이 'JACK'의 지도교수라는 연관관계를 알 수 있고 "교수 BOB의 지도학생들이 작성한 모든 논문"에 대한 검색이 가능하게 되는 것이다. 물론 경우에 따라 "Ontology Design의 저자들"을 검색하고 싶을 경우 RDF에서 'Ontology Design'이라는 리터럴 목적과, 'creation' 서술관계를 가지는주체를 검색하는 연산도 필요하다. 이와 같이 의미적인 검색이나 연관 검색을 지원하기 위해 RDF와 RDFS에서 표현되는 메타데이터의 내용에 기반한 캐쉬가 필요하다. 또한 예제에서 기술한 RDF는 한 개의 트리플로 정보를 표현하고 있지만 실제 정보를 기술하는 데에는 원하는 정보를 기술하기 위해 많은 트리플의 연결과 공백노드(blank node)들이 필요하다.

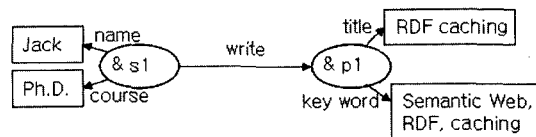


그림 5 공백노드(blank node)를 통한 표현

예를 들어 'Jack'이라는 사람이 가지는 여러 정보들을 표현하기 위해서는 공백노드를 넣고 이것에 대한 정보를 나타내는 트리플 관계를 기술해야 한다. 그림 5는 &s1인 공백 노드가 'Jack'이라는 대상을, &p1 공백노드가 'RDF caching'이라는 대상을 나타내며, 전체 트리플은 "박사과정의 Jack이 Semantic Web, RDF, Caching을 키워드로 가지는 RDF caching이라는 논문을 작성했다."라는 내용을 표현하고 있다. 트리플을 관계형데이터베이스에 저장하기 위해서는, 설계에 따라 다르지만 그림 6의 (a)와 같이 주어, 서술, 목적 세 요소를 한 개의 테이블에 표현 할 경우 데이터의 중복이 발생하고 트리플의 연관 검색시 마다 테이블에 자신에대한 조인이 발생한다. 예를 들어 그림 6의 'Jack'이 'Ph.D.'과정이라는 것을 알기 위해서는 조인이 발생한다. 그림 6의 (b)와 같이 주어, 서술, 목적 세 요소를 각 테이블에 표현하고 트리플을 구성하는 테이블이 이를 참조하도록 하면 중

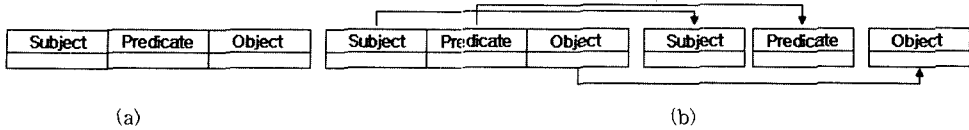


그림 6 트리플의 스키마 사상

복을 줄일 수 있지만 이 경우 하나의 트리플을 얻기 위해서도 조인이 발생한다. 어느경우든 시멘틱웹 데이터의 처리에서 연관된 정보를 가져오는 것은 조인을 필요로 하고 특히 공백노드의 연결을 따라서 값을 가져오는 것은 더 많은 조인을 발생시킨다. 정보의 기술이 복잡해 질수록 공백 노드와 연관성이 증가할 가능성이 높기 때문에 캐쉬의 필요성은 더욱 증대된다. 이와 같은 의미와 연관관계에 대한 캐쉬를 지원하기 위해 Pinakes의 캐쉬 관리자는 RDF와 RDF스키마의 모델을 그래프로 간주하고 특정 데이터에 대한 요구가 발생했을 경우 해당 node를 중심으로 세 가지의 집합을 만든다.

첫 번째 집합은 해당 노드 중심으로 거리 6(거리란 한 노드와 다른 노드간에 최단거리 간선의 수를 의미한다, 단 스키마와 관련된 간선은 계산하지 않는다)까지의 데이터로 한 집합을 생성한다. 이것은 트리플의 연결적 접근을 효과적으로 지원하기 위함이다. 그림 7은 교수, 학생, 연구 결과물인 논문과, 저서, 논문에 게재된 논문지, 책을 출판한 출판사에 대한 정보를 나타내는 RDF 스키마와 RDF 문서의 일부를 나타내고 있다. 여기서 'Ph.D.' 과정인 'Jack'의 'name' 서술에 대한 목적노드가 탐색의 현재 요구가 발생한 노드일 경우 집합을 생성하

면 공백 노드인 &s1, &p1, &j1, &s2, &p2, &j2와 각 공백노드에 바로 인접한 리터럴노드 들이 집합 Set1에 포함된다. 즉 'Jack'뿐만 아니라 'Jack'이 관리자인 같은 팀에 있는 'Andy' 의 논문과 게재정보까지도 포함된다. 두 번째 집합은 해당 노드의 바로 인접 노드의 타입 혹은 스키마에 해당하는 값을 보고 그 스키마에 종속되는 인스턴스들로 집합을 생성한다. 이것은 의미적으로 같은 스키마에 종속될 경우 탐색될 확률이 높기 때문이다. 예를 들어 논문에 대한 공백노드 &p1이 요구가 발생한 노드일 경우 다른 논문들도 검색될 확률이 높기 때문에 논문 같은타입인 다른 논문들이 두번째 집합 Set2에 포함된다.(그림 7에서는 개념적으로 표현하기 위해 인스턴스가 적게 표현되어 첫 번째 집합에서 모든 관련 논문이 포함되는 것처럼 보인다.) 세 번째 집합은 요구된 노드의 스키마를 보고 스키마상에서 거리 4안에 있는 연관된 스키마 노드의 인스턴스들로 집합을 생성한다. 이것은 의미적으로 연관이 있는 정보들을 포함하기 위해서이다. 예를 들어 그림 7에서 'Andy'의 'name'에 해당하는 목적이 요구된 노드인 경우 인스턴스 상에서는 교수 'Bob'과 아무 연관이 없지만 스키마를 통해 연관되어 있기 때문에 스키마를 통해 관련된 노드들을 집합 Set3

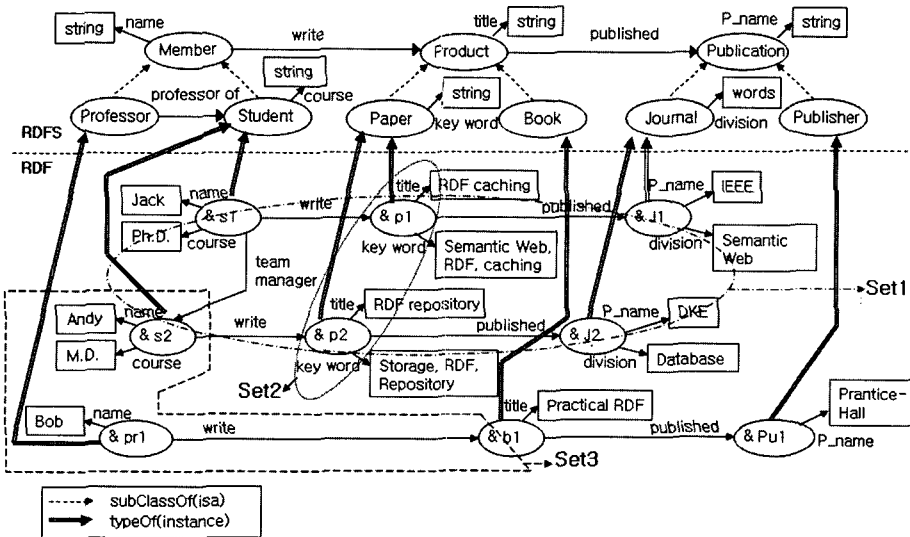


그림 7 RDF와 RDF스키마

에 포함시킨다.

이와 같이 생성한 세 개의 집합에 대해 그래프 구조를 생성하여 캐쉬집합을 생성한다. 만약 이 집합이 정해진 캐쉬 집합의 크기보다 클 경우 최 외각 노드부터 삭제하여 크기를 맞춘다. 캐쉬 관리자는 새로운 요구가 들어왔을 때 위와 같은 알고리즘을 통해 의미와 연관관계를 고려한 캐쉬를 생성한다. 그리고 캐쉬를 벗어나는 요구가 발생할 시 새로운 요구 노드를 중심으로 캐쉬를 생성한 후 기존 캐쉬와 그래프 관계를 따져 결합한 후 새로운캐쉬를 생성한다. 그리고 마찬가지로캐쉬의 용량에 맞추어 그래프 외각부터 제거해 조정한다. 캐쉬 관리자는 RDF 엔진에 접근하는 각 응용이나 사용자 별로 캐쉬를 생성하여 관리한다. 이것은 사용자가 요구하는 노드를 중심으로 캐쉬 집합을 생성하기 때문이며, 이와 같이 요구 별로 다르게 생성하는 이유는 접근 빈도보다는 사용자가 연관관계에 따라 접근하는 노드가 달라지며, 각 노드마다 탐색될 확률이 달라지기 때문이다. 본 기법에서는 캐쉬를 생성하는 집합을 구축할 때 정적으로 거리를 제한하고 있다. 그 이유는 그래프 관계를 다 고려하는 것은 많은 처리시간을 필요로 하기 때문이다. 이와 같은 거리를 결정하는데 있어서 고려해야 할 것은 먼저데이터의 구조에 따라서 연결이 긴, 즉 연관관계가 많은 경우와 한 노드에 인접 노드가 많은 경우에 따라 각 집합을 생성하는데 있어 처리의 효율과 정보의 완전성을 고려해야 한다. 또한 캐쉬의 크기와 캐쉬 생성에 소모되는 시간을 고려해야 한다. 이와 같은 것들은 스키마 설계자와 데이터 구축에 종속적이기 때문에 본 연구에서는 본 시스템과 데이터의 환경에 적절하도록 집합을 생성하기 위한 거리를 실험적으로 정하였다. 마지막으로 응용 별로 캐쉬를 유지하는 것은 앞에서 설명한 캐쉬 교체기법에서 중복부분을 제거하기 위해 현

재 캐쉬를 알고 있어야 하기 때문이다. Pinakes는 이와 같은캐쉬 기법으로 RDF모델의 연관관계와 의미를 고려한 효율적인 처리를 지원한다.

4.4 RDF 저장소

RDF 스토리지 래퍼는 RDF엔진의 RDF 오브젝트 래퍼를 상속받아 각 저장소에 해당하는 래퍼를 구축한 것으로서 RDF 데이터를 연속적 저장소인 데이터베이스에 입출력하는 기능을 수행한다. 저장소에 대해 이와 같이 래퍼를 두고 각 래퍼가 다른 종류의 저장소를 지원할 수있게 하는 인터페이스를 구축함으로써 저장장치 독립성과 다양한 저장소를 지원할 수 있게 하였다. RDF 저장소로는 관계형데이터베이스, XML 저장소, OODB가 사용될 수 있다. XML 저장소의 경우 XML로 표현된 RDF문서를 가공 없이 바로저장 할 수 있다는 장점을 가진다. 하지만 이것은 XML모델로 표현된 데이터므로 RDF 처리를 위해서는 다시 RDF모델로 파싱하는 과정이 필요하다. 객체지향 데이터베이스에 저장하는 경우에는 RDF모델이 객체지향 모델과 유사하기 때문에 쉽게 매핑할 수 있다는 장점을 가지지만 관계형데이터베이스에 비해 대용량의 데이터와 효율적인 처리가 부족하다. 따라서 본 논문에서는 관계형데이터베이스를 중심으로 저장소를구축하였다. 관계형데이터베이스의 경우 용량과 안정성, 성능에 있어 이미 검증이 되었을 뿐만 아니라 실제 환경에서 가장보편적으로 사용된다는 장점을 가진다. 이와 같은 이유로 본 연구에서는 관계형데이터베이스를 기반으로 시스템을구축하였다.

이 경우 RDF 모델을 관계형 모델로 매핑하는 것이 중요하다. 즉 RDF모델을 관계형 모델로 변환하는 것이 쉽지 않을뿐더러 설계에 따라 처리 효율과 표현력에 많은 차이를 가져온다. 특히 RDF의 경우 기본 요소인 트리플과 그에 대한 연관검색이 주를 이루기 때문에 조인

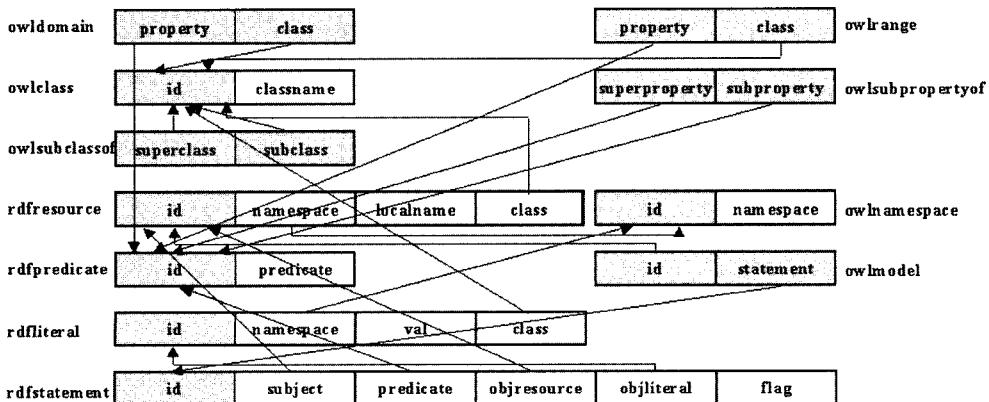


그림 8 RDF 모델을 지원하기 위한 관계형데이터베이스 스키마

의 발생을 최소화 하는것이 중요하다. 본 논문에서는 관계형데이터베이스를 저장소로 구현하였으며 그림 8과 같은 스키마를 설계하였다.

위의 구조는 RDF 데이터모델 뿐만 아니라 추후 OWL확장을 고려해서 설계하였다. 먼저 RDF에 대한 트리플 요소에 대해서 주어와 목적은 rdfsresource 테이블에 서술은 rdfsredicate에 리터럴은 rdfsliteral에 담고 rfsstatement 테이블이 이를 참조하도록 구성하여 중복을 최소화 하는데 중점을 두었다. 또한 OWL을 기술하기 위해 OWL의 표준안에 명시된 요소들에 대한 필드들을 owlDomain, owlRange, owlClass, owlSubPropertyOf, owlSubClassOf, owlNamespace, owlModel 테이블에 기술하고 각 정보에따라 참조하도록 하였다. 이 경우 각 요소들이 분리되어 저장되고 참조를 통해 연결되기 때문에 4.3절에서 설명한 바와 같이 정보의 중복은 없지만 조인이 발생한다. 이와 같은 문제들은 XML 저장소의 질의처리 문제와 유사성을 가지는데 이 경우 1-index, 2-index, T-index[19] 및 반구조적 데이터를 위한 인덱싱 기법[20]이 제안되어 있으며, 이러한 기법들은 RDF에 적용해 성능을 높이는 것을 고려해 볼 수 있을 것이다. Pinakes는 이와 같이 저장소 독립적 설계와 표현에 대한 확장성 및 인덱싱에 용이하도록 관계형데이터베이스 기반의 저장소를 구현하고 있다. 이와 같은 설계와 더불어 Storage 래퍼는 RDF모델에 대한 접근을 그림 8과 같은 스키마에 대한 입출력으로 변환하는 기능을 담당한다.

#### 4.5 서비스

서비스부는 사용자에게 편리한검색을 지원하기 위한 사용자 인터페이스와 이를 웹서비스로 제공하기 위한 웹서비스 인터페이스로 구성된다.

##### 4.5.1 사용자 인터페이스

Pinakes는 실제 서비스 환경을 고려하고 있기 때문에 온톨로지 탐색에 있어 사용자에게 다양한 방식의 탐색 기법을 제공하고 있다. 관련연구에서 설명한것과 같이 일반적인 시스템에서 지원하고 있는 한, 두 가지 방식의 탐색기법을 지원하는 것은 사용에 있어 용이하지 않다. 따라서 Pinakes에서는 그래프, 트리, 키워드기반의 탐색을 복합적으로 지원할 뿐만 아니라 각 기능에 있어서도 효율적인 검색을 지원하는 방법을 제안하였다. 먼저 트리 구조를 통한 계층 관계를 따르는 검색이 가능하다. 트리 구조는 정보의 수직적 계층구조를 기반으로 트리를 생성해서 검색하는 방식으로 정보의 계층적 구조를 파악하는데 효율적이다. 그래프 구조의 표현은 정보간의 관계를 쉽게 파악할 수 있다는 장점을 가지고 있지만 많은 데이터에 대해서는 그래프가 복잡해져서 탐색이 어렵다는 단점을 가지고 있다. 따라서 본 논문에서는 그

래프를 표현하는데 있어 거리 1의 인접 노드만을 표현하는 방법을 제안하였다. 이 경우 선택되는 노드에 따라서 동적으로 관련된 정보가 표현되어야 하며 한 노드에 많은 에트리뷰트가 존재하는 경우(많은 데이터가 연관된 경우)에도 유용하다. 키워드 검색의 경우 트리플 모델을 기반으로 트리플 요소를 선택하고 해당 요소에 대한 키워드를 입력함으로써 해당 요소에 대한 결과를 트리와 그래프로 함께 나타내도록 하였다. 예를 들어 트리플 요소중에서 주제를 'JACK'이라고 키워드를 입력하면 JACK을 중심으로한 그래프가 동적으로 생성되며, 트리에서는 JACK까지의 내용이 열려서 표현되게 된다. 이와 같은 세가지 방식의 연계를 통해 키워드 검색의 구조 파악의 어려움, 트리 검색 시 분류를 모르거나 깊이가 깊어질 경우의 어려움, 그래프 검색 시 표현이나 탐색에 대한 어려움을 극복할 수 있게 된다. 또한 다른 시스템에 없는 독창적인 기능으로 메타 데이터 처리를 통한 통계 및 결과 그래프를 지원한다. 따라서 온톨로지를 통한 고급 정보 제공이라는 측면에도 부합되도록 하였다. 실제 모습에 대해서는 다음장의 구현 결과에서 살펴 보도록 하겠다.

##### 4.5.2 웹서비스 인터페이스

Pinakes의 최상위 부분에는 서비스 인터페이스가 위치한다. 이것은 관리자나 사용자를 위한 온톨로지 서비스를 제공하기 위한 서비스 인터페이스와 이를 다시 SOAP(Simple Object Access Protocol)을 지원하도록 구현함으로써 웹서비스를 지원하도록 한 웹서비스 인터페이스로 구성된다. 따라서 온톨로지에 대한 접근을 웹서비스로 구현할 수 있는 기능을 제공한다. 이와 같은 구성을 통해 일반적인 관리자, 사용자 서비스뿐만 아니라 웹서비스까지 포함하는 다양한 서비스를 지원할 수 있도록 하였다.

## 5. 결 과

### 5.1 구현 및 실험환경

Pinakes는 java와 JSP로 구현되었으며 웹브라우저를 통해 탐색하게 된다. 시스템 환경은 AMD 1.2Ghz CPU, 512MB RAM의 컴퓨터와 Linux 8.0 OS, J2EE 1.4, Tomcat 4.1, SOAP 2.3.1 이다. 데이터는 연구실의 구성원과연구활동, 프로젝트, 논문 등 연구실의 정보에 대한 내용을 RDF로 구축한 2MB의 문서를 사용하였다.

### 5.2 구현 결과

다음의 그림 9는 본 논문에서 구현한 pinakes시스템의 실행화면이다.

위의 그림은 특정 논문에 대해서 검색한 내용으로서 해당 논문에 대한 온톨로지 상에서의 정보를 나타내고 있다. 만약 이와 같은정보를 온톨로지가 아니라 웹상에



서 홈페이지를 통해 검색한다고 해보자. 사용자는 웹사이트에 들어가서 먼저 해당 논문을 여러 링크를 통해 검색하고 그 논문과 같은 곳에 게재된 논문이나 해당 저자가 쓴 다른 논문을 찾고 싶을 경우 다시 여러 번의 링크 검색과정을 거쳐야 한다. 또한 해당 저자가 몇 년도에 어떤 종류의 논문을 몇편이나 작성했는지 알고 싶을 경우 다시 여러 링크를 검색하면서 수많은 리스트 중에서 일일이 찾아 세어야 할 것이다. 하지만 온톨로지는 연관 관계를 보여주고 이를 통한 검색을 지원함으로써 효율적인 정보검색이 가능하며, 부가 정보의 처리를 통해 고급 정보의 제공도 가능하다. 또한 위에서 보듯이 본 논문에서 구현한 시스템이 매우 간결하고 직관적인 인터페이스를 제공함으로써 관리자뿐만 아니라 일반 사용자를 위한 서비스에 잘 부합되는 것을 알 수 있다. 이와 같이 본 논문에서 구축한 온톨로지 관리시스템이 정보 검색과 서비스에 있어 효과적임을 확인할 수 있다.

5.3 타 시스템과의 비교

표 2는 RDF를 지원하는 타 시스템과 각 기능에 대한 비교를 나타내고 있다. 다른 시스템에 비해 Pinakes는 컴포넌트를 독립시키고 인터페이스를 제공하여 다양한 확장을 지원하며 그로 인해 다양한 저장소를 지원할 수 있다. 또한 캐쉬와 관계형데이터베이스 래퍼를 통한 효율적인 데이터 처리를 고려하고 있으며 실제 서비스를 고려한 서비스기능 강화로 지역 서비스 및 웹 서비스를 지원한다. 또한 사용자 인터페이스에 있어 관리자 및 일반 사용자가 편리하게 사용할 수 있도록 그래프, 키워드, 트리 인터페이스가 유기적으로 동작할 수 있도록 하였으

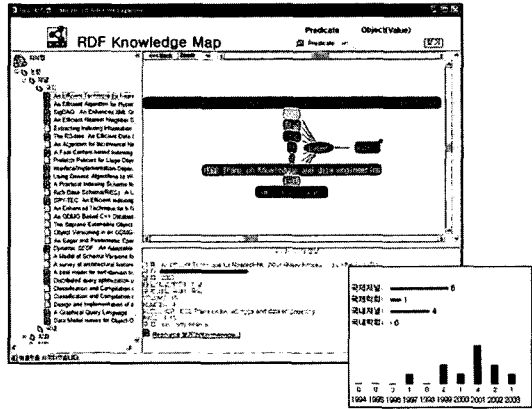


그림 9 온톨로지 관리 시스템(Pinakes)의 실행화면

며 메타데이터에 대한 통계 기능을 추가함으로써 온톨로지를 통한 고급정보를 얻을 수 있다.

표 2의 시스템들은 저장부터 처리 및 사용자 인터페이스를 지원한다는 점에서 Pinakes와 유사하지만 지원하는 질의언어가 다르거나 RDB저장을 지원하지 못하는 등 차이점이 많아 응답속도 보다는 기능 위주로 비교를 하였다. Protégé-2000의 경우 시스템이라기보다 플러그인으로 구축되는 프레임워크이고 플러그인에 따라 결과가 결정되기 때문에 비교하지 않았다.

응답속도 비교를 통한 처리 성능을 살펴보는 것에는 Pinakes와 동일하게 RDQL을 지원하고 RDB를 지원하는 Sesame 1.1에 같은 데이터와 같은 질의를 사용해서 비교하였다. 특히Sesame는 위의 시스템들과 달리 직접

표 2 타 시스템과의 비교

	Pinakes	KAON	OntoEdit
저장소	In-memory, RDB, File, OODB, XML-Storage	In-memory, File	자체 포맷
검색방식	그래프, 트리, 키워드	트리, 검색기능 미약	그래프, 트리, 키워드
질의	RDQL	F-Logic	없음
캐쉬	지원	없음	없음
서비스	웹서비스, 로컬서비스, API 지원	API 지원	없음
통계기능	지원	없음	없음
외부파일 출력	RDF	없음	RDF
추론	없음	지원	지원
데이터 모델	RDF	RDF, DAML	RDF

표 3 실험에 사용된 질의

질의	유형	예제
Q1	한 요소의 인접 정보를 탐색하는 너비 검색에 해당	'x'라는 연구원과 연관된 모든 관계들과 자료를 찾으시오
Q2	연관 정보따라서 탐색하는 깊이 검색에 해당	'x'라는 연구원이 'y'년도에 작성한 결과물을 찾으시오
Q3	깊이, 너비 검색 및 스카마 정보가 관련되는 검색에 해당	'x'의 논문들의 공동 저자들 중에서 'team manager'인 연구원들의 연구분야를 찾으시오

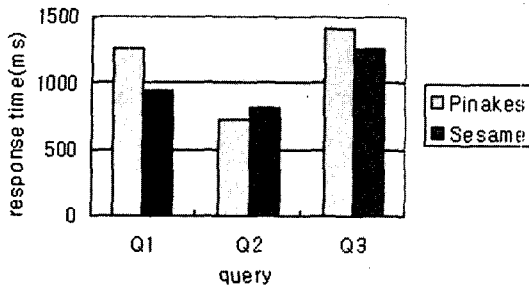


그림 10 실험 결과 1

적으로 기능을제공하지 않고 RDF를 다루기 위한 API 들을 지원하므로 위와 같은 기능비교보다는 성능 비교에 적합하다. 실험환경은 앞에서 언급하였으며 Pinakes의 캐쉬 크기는 400KB, 질의문은 유형에 따라 표 3과 같이 분류하였다. 그리고 각 질의 유형별로 50개의 질의를 만들어서 실험을 수행하였으며 응답시간은 평균 응답시간을 의미한다. 그 결과는 그림 10과 같다.

위의 결과에서 Q1과 Q2에서는 sesame보다 성능이 떨어지며 Q2에서 조금 앞서고 있다. 이것은 RDF 엔진에서 데이터를 처리하는데 있어 아직 개선의 여지가 있다는 것을 알 수 있다. 하지만 연관 데이터의 접근인 Q2 질의에 대한 결과에서 보듯 캐쉬를 통해 효율적인 처리가 가능함을 확인할 수 있었고, 실제 온톨로지에 대한 접근이 주로 데이터의 연관성에 의해 이루어진다는 점을 고려할 때 본 논문에서 제안한 기법이 매우 유용하게 작용할 것이다.

다음으로 캐쉬의 효율을 살펴보기 위해 Pinakes에 캐쉬를 사용하지 않은 경우와 LRU 캐쉬를 사용한 경우에 대한 캐쉬의 hit ratio와 캐쉬를 사용하지 않는 경우를 포함한 경우의 응답시간을 측정해 보았다.

먼저 그림 11의 왼쪽은 LRU기법과 비교한 캐쉬의 hit ratio를 나타내고 있다. LRU기법과 비교할 때 Q1 유형의 질의에 대해서 좋은 결과를 보이지만 차이가 크지

는 않다. 이것은 너비검색 유형의 질의가 시간적 지역성에 의해서도 효과를 볼 수 있기 때문으로 보인다. 하지만 Q2와 같이 연관성을 중심으로 검색하는 경우 LRU와 비교해 우수함을 명확히 확인할 수 있다. 복합적인 형태 및 스키마 정보를 사용하는 Q3는 넓은 검색범위를 가지기 때문에 나머지 두질의에 비해 전체적인 성능이 낮지만 여기에서도 본 논문에서 제안한 기법이 우수함을 확인할 수 있다. 그림 11의 오른쪽, 응답시간에 있어 Q1질의에서는 본 논문의 캐쉬기법이 hit ratio가 뛰어난데도 두 캐쉬기법이 큰 응답시간의 차이를 보이지 못한다. 그 이유는 캐쉬 집합을 생성하기 위해 여러 노드를 탐색하는 것이 LRU보다 비용이 크기 때문이다. 하지만 hit ratio 차이가 큰 Q2 질의에서는 명확한 응답시간의 차이를 보이고 있으며 깊이와 너비검색이 복합적이라고 볼 수 있는 Q3에서도 LRU보다 우수한 효과를 보이고 있다. 본 결과를 통해 의미나 연관성 기반의 캐쉬기법이 기존에 시간적인 접근의 지역성에 기반한 캐쉬기법보다 온톨로지에 효과적임을 확인할 수 있다.

### 6. 결론 및 향후 연구방향

온톨로지란 기술하고자 하는 데이터에 대한 부가정보 및 의미적 정보, 그리고 정보간의 관계를 기술함으로써 유기적으로 정보를조직함은 물론 소프트웨어 에이전트로 하여금 정보의 의미를 이해하고 처리할 수있는 기반을 제공한다. 최근 온톨로지와 관련된 다양한 표준이 제정되고 있는데 대표적인 시멘틱웹기술어인 RDF는 OWL, DAML+OIL과 같은 온톨로지 언어의 기반을 제공하며 온톨로지의 기본 요소를 형성한다. 온톨로지에 있어 가장 중요하고 기본적인 연구는 바로 온톨로지 생성, 편집, 저장 및 검색등을 지원하는 온톨로지 처리 시스템이다. 하지만 현재 구축되어있는대부분의 시스템은 관리기능은 잘 구현하고 있지만 효율적인 처리나 실제 서비스에 대한 부분에 대한 고려가 부족하다. 따라서 본 논문에서는 RDF를 이용하여 실질적인 서비스와 효율적

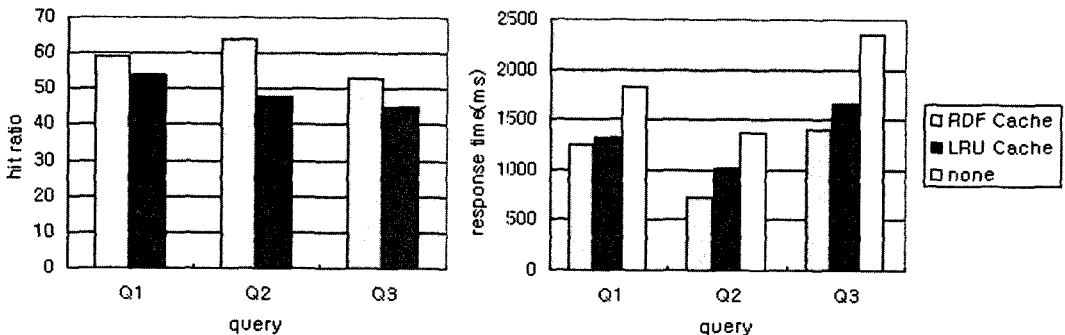


그림 11 실험 결과 2

인 처리를 지원하는 온톨로지 처리 시스템을 구축하고 관련 기법에 대해서 제안하였다.

먼저 시스템의 각 기능에 따라 분류하고 독립적인 인터페이스를 구축함으로써 다양한 확장과 API를 통한 응용을 가능하게 하였으며 온톨로지의 접근이 연관성이나 의미적인 관계에 따라 이루어지는 것에 기반하여 RDF와 RDF 스키마의 연관정보를 통해 캐쉬를 생성하는 캐쉬기법을 제안하였다. 또한 저장장치에 있어 저장소와 처리를 수행하는 RDF 엔진 사이에 인터페이스를 구축함으로써 저장 장치독립성을 이룩하고 이를 통해 다양한 데이터베이스를 지원할 수 있게 되었다. 특히 관계형 데이터베이스 기반의 저장소를 구축하는데 있어 RDF의 트리플 모델과 트리플의 연속적 검색을 잘 지원할 수 있는 관계형 데이터베이스 스키마를 제안하였다. 이와 같이 캐쉬와 저장소를 통해 효율적 데이터 처리기법을 고려함은 물론 지역서비스 및 웹서비스를 고려한 인터페이스를 제공함으로써 실제 서비스 환경을 잘 지원할 수 있도록 하였다. 또한 사용자 인터페이스에 대해서도 트리, 그래프, 키워드를 조합시켜 각 기법의 단점을 극복한 사용자 중심의 사용환경을 지원함으로써 효율적인 검색을 지원하고 관리자뿐만 아니라 일반 유저들에게도 유용하도록 하였다. 특히 메타데이터를 처리한 통계기능을 추가함으로써 온톨로지를 통해 고급정보를 얻을 수 있도록 하였다. 그리고 이와 같이 구현한 시스템을 다른 온톨로지 시스템과 기능을 비교해봄으로써 유용함을 확인해 볼 수 있었다.

하지만 아직 여러 부분에서 더 많은 연구가 필요하다. 먼저 효율적인 처리에 대한 부분으로서 캐쉬 기법에 있어 좀더 체계적으로 접근 패턴을 분석하고 이를 기반으로 한 의미적 캐쉬에 대한 더 깊은 연구가 필요하며, 그 외 데이터를 처리하는 RDF 엔진부의 개선이 필요하다. 저장에 있어서는 관계형 데이터베이스 저장에 있어 이를 효과적으로 지원하기 위한 인덱싱 기법에 대한 연구도 필요하다. 이와 같은 효율적 처리 외에도 이질적인 구조나 동의어를 가지는 온톨로지를 통합하거나 변환하는 기법, 그리고 온톨로지의 정보에 대한 추론으로 더욱 고급 정보를 추출하는 기술과 다양한 응용분야에 적용하기 위한 연구가 이루어져야 할 것이다.

## 참고 문헌

- [1] T. Berners-Lee, J. Hendler, O. Lassila. "The Semantic Web," Scientific American, May 2001.
- [2] S. Staab, H.-P. Schnurr, R. Studer, & Y. Sure. "Knowledge processes and ontologies," IEEE Intelligent Systems, 16(1), January/February 2001.
- [3] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, "The Semantic Web: the roles of XML and RDF," IEEE Internet Computing, Volume:5, Issue:4 Sept.-Oct. 2000.
- [4] Brian McBride, Hewlett-Packard Laboratories, "RDF Primer," W3C Recommendation 2004, <http://www.w3c.org/TR/rdf-primer/>
- [5] Deborah L. McGuinness, Frank van Harmelen. "OWL Web Ontology Language Overview," W3C Recommendation 2004, <http://www.w3c.org/TR/owl-features/>
- [6] Graham Moore, "Topic Map technology—the state of the art," XML 2000 Conference & Exposition, Washington, USA, December 2000.
- [7] James A. Hendler. "Agent and the semantic web," IEEE Intelligent Systems Journal, March/April 2001.
- [8] Prasenjit Mitra, Gio Wiederhold, Martin L. Kersten, "A Graph-Oriented Model for Articulation of Ontology Interdependencies," EDBT 2000.
- [9] Horrocks, I., et. al. "The DAML+OIL Language Specification," <http://www.daml.org/2001/03/daml+oil-daml>
- [10] Y. Sure, S. Staab, J. Angele, D. Wenke, and A. Maedche. "OntoEdit: Guiding ontology development by methodology and inferencing," Prestigious Applications of intelligent Systems (PAIS), in conjunction with ECAI 2002, Lion, France, July 2002.
- [11] Alexander Maedche, Boris Motik and Raphael Volz. "KAON-A Framework for semantics-based E-Services," Institute AIFB, University of Karlsruhe
- [12] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson, & M. A. Musen. Creating "Semantic Web Contents with Protege-2000." IEEE Intelligent Systems 2001. 16(2):60-71
- [13] Almilia Magkanaraki, Grigoris Karvounarakis, Ta Tuan Anh, Vassilis Christophides, Dimitris Plexousakis, "Ontology Storage and Querying," Technical Report No 308., ICS-FORTH, 2002.
- [14] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema." ISWC 2002.
- [15] Shaul Dar, Michael J. Franklin, Bjorn T. Jonsson, Divesh Srivastava, Michael Tan, "Semantic Data Caching and Replacement," VLDB, 1996.
- [16] Jia Wang, "A Survey of Web Caching Schemes for the Internet," ACM SIGCOMM 1999.
- [17] Paul Mutton, Jennifer Golbeck. "Visualization of Semantic Metadata and Ontologies," In Seventh International Conference on Information Visualization (IV03), IEEE, July 2003.
- [18] Andy Seaborne. "RDQL A Query Language for RDF," W3c member submission, Hewlett Packard, January 2004, <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>

- [19] Tova Milo, Dan Suciu, "Index Structures for Path Expressions," ICDT, 1999.
- [20] Roy Goldman, Jennifer Widom, "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases", VLDB, 1997.



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



정 호 영

2000년 한국외국어대학교 컴퓨터공학과(학사). 2002년 한국외국어대학교 컴퓨터공학과(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 : 컴퓨팅의 실제  
제 11 권 제 3 호 참조