시멘틱 검색 엔진 설계 및 구현

허선영*, 김은경
한국기술교육대학교 대학원 전기전자공학과*
한국기술교육대학교 정보기술공학부
hsysj119@naver.com*, egkim@kut.ac.kr

A Design and Implementation of the Senamtic Search Engine

Sun-Young Heo*, Eun-Gyung Kim
Korea University of Technology and Education, Graduate School,
Dept. of Electricity & Electronic eng*.
Korea University of Technology and Education,
School of Information Technology Eng.

요 약

시맨틱 웹은 정보의 의미를 개념으로 정의하고 개념들 간의 관계성을 표현함으로써, 문서들 간의 단순 연결이 아닌 의미 연결을 통해서 보다 정확하고 효율적인 정보 검색이 가능하게 된다. 이러한 시맨틱 웹의 비전이 구체화되기 위해서는 웹 온톨로지(Web Ontology)를 기반으로 의미 정보로 구성된 시맨틱 문서들에 대한 추론을 통해서 웹상에 존재하는 엄청난 정보들 간의 관련성을 파악하고 사용자가 요구하는 정보를 보다 효율적으로 검색할 수 있는 시스템이 필수적이다. W3C에서 제안한 OWL은 대표적인 온톨로지 언어이다. 시맨틱 웹 상에서 OWL 데이타를 효율적으로 검색하기 위해서는 잘 구성되어진 저장 스키마를 구축해야 한다. 본 논문에서는 Jena2의 경우, 단일 테이블에 문서의 정보를 저장하기 때문에 단순 선택 연산 (Simple Selection), 조인 연산이 요구되는 질의에 대한 성능이 저하되고 대용량의 OWL데이터의 처리에 있어 성능이 저하되는 문제를 해결하기 위하여 본 논문에서는 OWL 문서의 의미를 Class, Property, Individual로 분류하여 각각의 데이터 정보들을 테이블에 저장하기 위한 다중 변환기와 OWL 변환기 기능을 가진 시멘텍 검색 엔진을 설계 및 구현하였다. 본 검색 엔진을 테스트한 결과, 단순정보검색 질의 시 Jena2에서 비정규화된 테이블 구조로 저장할 때보다 질의 응답 속도를 향상 시킬 수 있었고, 조인 연산 시 두 테이블의 크기로 인한 조인비용이 발생하는 문제점을 해결함으로써 빠른 검색 및 질의 속도를 보장할 수 있었다.

1. 서 론

웹(Web)이 우리 일상의 한 부분이 된지 이미 오래되었고, 현대 사회의 모든 분야에서 정보 제공 및 정보 교환의 필수적인 요소가 되었다. 그러나 현재의 웹은 문서들 간의 단순 연결을 통해서 정보를 제공하며, 이러한 연결을 통해서 상호 관련된 정보를 쉽게 찾을 수 있다는 장점이 있지만, 수집 및 검색된 정보는 사람에 의해서 해석되고 정제되어야 한다는 점에서 정보 저장소 이상의 기능을 제공하지는 못하고 있는 실정이다. 따라서 이러한 기존 웹의 한계점을 극복하기 위해서 World Wide Web Consortium(W3C)표준으로 규정된 시맨틱 웹 온톨로지 언어(RDF, RDFS, OWL 등) 및 관련 기술에 대한 연구가 활발히 진행되고 있다. W3C가 OWL을 표준 온톨로지 언어로 지정함에 따라 추후 대부분의 온톨로지 데이터들은 OWL로 기술될 전망이다. 이로 인해, 최근 RDF[3, 4], RDFS, OWL, SPARQL[1] 등의 언어를 위한 프로그램 환경에서 규칙 기반 추론 엔진을 포함하고 있는 Java 프레임워크인 Jena2[2]에 대한 관심이 높아지고 있다.

Jena2는 OWL 데이터에 대한 저장모델 자동 생성 시 물리적 스키마 구조가 단순하고, 이로 인해 많은 OWL 관련 시스템 개발에 이용되고 있다. 하지만 비정규화된 단일구조로서 단일테이블에 정보들이 저장됨에 따라, 대용량의 OWL 데이터에 대한 저장 및 질의 시, 성능이 저하되는 문제점을 가지고 있다[5, 10]. 본 논문에서는 단순 선택 연산 (Simple Selection)은 물론 조인 연산이 요구되는 질의에 대한 성능이 저하되는 문제를 해결

하고, 기본적인 Jena2 API를 이용하여 검색엔진을 구현하였다. Jena2 API를 이용하여 구현한 본 검색엔진의 데이터 저장 및 질의처리를 위한 성능을 향상시키기 위하여 다중 변환기와 OWL 변환기를 설계 및 구현하였다. 본 논문에서는 데이터 저장 및 질의처리 성능의 향상을 위하여 구현한 다중 변환기와OWL 변환기의 설계 및 구현에 대한 내용을 다루고 있다

2. Jena2의 개요

2.1 Jena2 구조

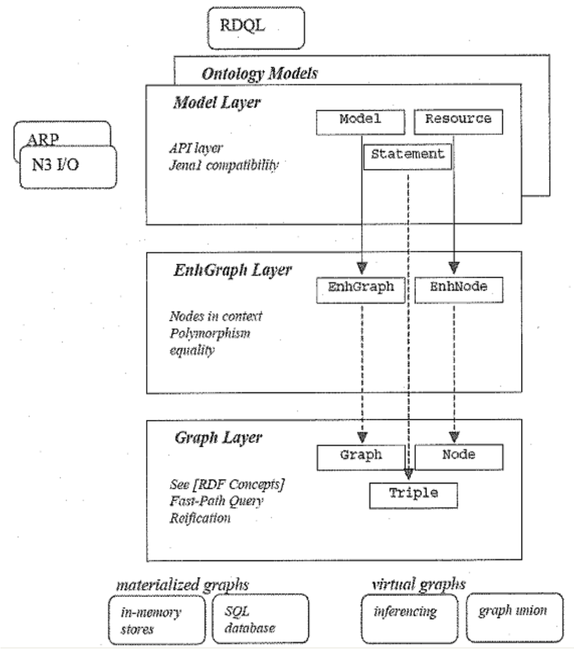

그림 1 Jena2 구조

Jena2의 구조는 그림1과 같다. Jena2는 시멘틱 웹 응용프로그램을 구축하기 위한 자바 프레임워크로서RDF, RDFS, DAML+OIL[6], OWL을 포함하고 룰 기반 추론 엔진과 질의를 위한 RDQL[7], SPARQL 등을 포함하는 응용프로그램 개발API를 제공한다. Jena2는 오픈 소스로서 HP 연구소에서 개발 한 툴이며, 현재 2.5 버전까지 개발되어 있다.

그림 2는 특정 OWL파일을 로드한 후 Jena 관계형 데이터베이스에 데이터를 저장하는 스키마를 나타낸다 Jena2 프레임워크를 통해서 OWL파일을 저장할 때 총 7개의 테이블만이 생성된다 OWL 문서에 대한 실제 데이터 정보는 subject, property, object의 컬럼으로 구성된 “jena_gntn_stmt”라는 테이블에 분류되어 저장된다

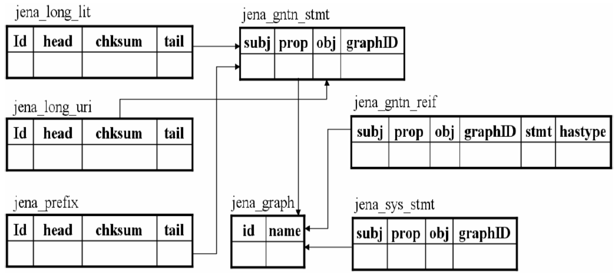

그림 2. Jena2 관계형 데이터베이스 저장 스키마

2.2 Jena2의 문제점

Jena1는 트리플 데이터를 저장하기 위해 정규화된 데이터베이스 스키마를 사용하였기 때문에 저장 크기 면에서는 효율적인 장점이 있으나 질의 시 Statement 테이블과 리터럴 테이블 그리고 리소스 테이블 간에 많은 횟수의 조인연산을 필요로 해야 하는 단점이 있다 Jena2의 경우, Jena1의 문제점을 해결하기 위해 의도적으로 비정규화를 시킴으로써 Jena1과 비교하여 데이터의 검색 시간과 조인 연산의 횟수는 줄일 수 있는 장점이 있지만, 하나의 테이블 공간이 커짐으로써 단순 선택연산 시 성능이 저하되는 문제점을 지닌다 또한 비정규화가 높아 조인 연산 시 불필요한 정보를 액세스하게 되고 이에 따른 성능 저하를 가져온다 본 논문에서는 이러한 Jena2의 문제점을 보완하기 위하여 OWL 문서의 의미론적 입장에서 Class, Property, Individual로 구조화하여 관계형 데이터베이스에 저장하였다

3. 다중 변환기와 OWL 변환기

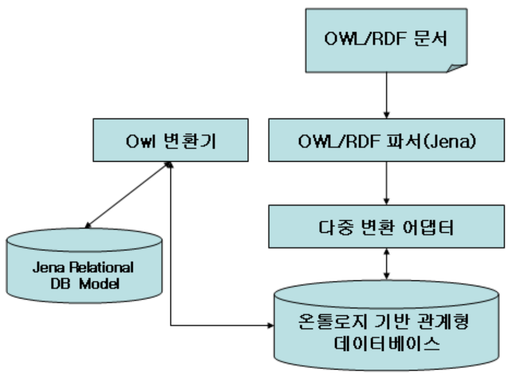

그림 3. 다중 변환기와 OWL 변환기

본 논문에서 구현한 다중 변환기와OWL 변환기의 구조 및 기능은 그림3과 같다. 기존 Jena2 시스템의 문제점을 보완하기 위하여 OWL 문서의 의미를 Class, Property, Individual로 분류하여 관계형 데이터베이스에 저장하기 위한 다중 변환기와 OWL 변환기를 설계하였다.

그림 3에서 알 수 있듯이, 다중 변환기는 Jena2 의 OWL/RDF 파서를 이용하여 OWL 문서를 파싱하고 파싱한 결과를 다중 변환기에게 전달한다. 다중 변환기에서는 전달받은 결과를 새로운 개념의 온톨로지의 의미 기반 정보 형태로 변환하여 관계형 데이터베이스에 저장된다. OWL 변환기는 Jena2의 저장소에 이미 저장된 데이터를 마이그레이션(Migration) 하기 위하여 온톨로지 기반 관계형 데이터베이스로 저장한다. 본 논문에서는 기존의 Jena2 프레임워크에서 단일테이블로 생성되었던 구조를 OWL 문서의 의미론적 입장에서 Class, Property, Individual로 분류하여 구조화했다. OWL 문서의 의미를 Class, Property, Individual로 분류하여 각각의 데이터 정보들을 테이블에 저장한다.

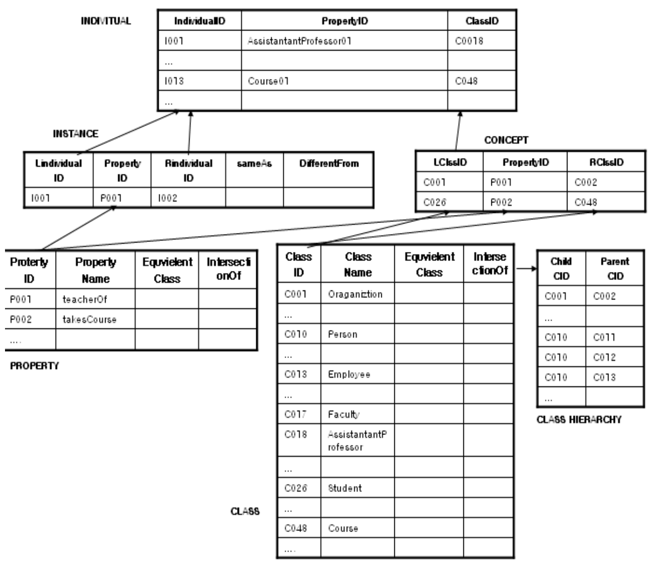

그림 4. 온톨로지 관계형 데이터베이스 스키마

그림 4에서 보는 바와 같이, 본 논문에서는 OWL 문서의 내용을 Property와 Class의 해당 정보에 대한 각 테이블은 주 키 값인 ID와 데이터 값인 Name 필드 이외에도 추가 적인 정보를 저장하기 위한 필드로 구성된다. 또한 각 클래스 간의 부모에 해당하는Class와 부모에게 포함되어 있는 자식관계에 대해서는 CLASS HIERARCHY를 통하여 표현할 수 있다. 이는 질의에 포함되어 있는 특정 클래스의 하위 클래스까지 추론하여 보다 정확한 질의 결과를 생성할 때 빠른 연산을 가능하게 한다. Individual-Property-Individual 형태의 Instance와 Class-Property-Class형태의 Concept 테이블과 Class와 Individual의 관계를 매핑 시킬 수 있는 테이블로 구성된다.

3.1 다중 변환기

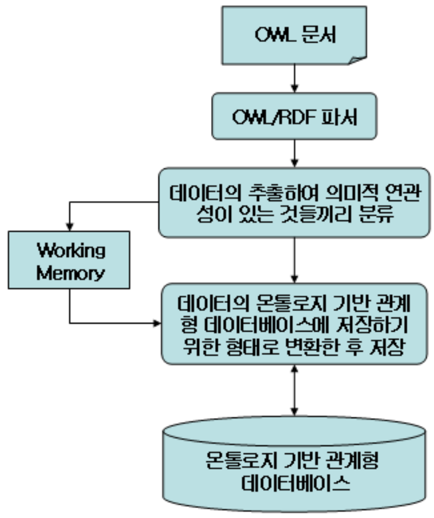

그림 5. 다중 변환기의 처리 과정

그림 5는 다중 변환기의 변환 과정을 도식화하여 보여주고 있다. 다중 변환기는 OWL문서를 Jena2의 RDF 파서를 이용하여 파싱한다. 이 파싱한 결과는 OWL문서의 의미적 연관성을 고려하여 데이터를 추출하고 변환한다. 변환 후 온톨로지의 구조를 기반으로 테이블을 생성한 다음 변환된 데이터들을 온톨로지 기반 관계형 데이터베이스에 저장한다.

다중 변환기는 OWL 데이터 정보에 대해 추출하고 분류하는 부분과 데이터를 임시 저장할 수 있는 작업 메모리 부분, 그리고 변환하여 관계형 데이터베이스에 저장할 수 있도록 지원하는 부분 등 세 부분으로 구성되어 있다. OWL 데이터 정보에 대해서 추출하고 분류하는 부분에서는 Jena2의 파서를 이용하여 읽어 들인 OWL 문서의 의미를 분석한 후 요구하는 정보들을 그림 2의 온톨로지를 기반으로 한 테이블들에 저장될 수 있도록 정보를 분류한다. 분류된 데이터 정보들을 임시로 저장할 경우 Working Memory에 데이터를 보관한다. 온톨로지를 기반으로 데이터를 생성하고 테이블을 생성하는 과정을 거쳐 분류된 데이터들을 테이블에 저장한다. 이와 같이 OWL 문서를 다중 변환기의 저장 프로세스를 통해 최적화된 온톨로지 기반 관계형 데이터베이스에 저장함으로써 기존 Jena2의 단일화로 구성된 저장 모델의 단점을 보완할 수 있다.

3.2 OWL 변환기

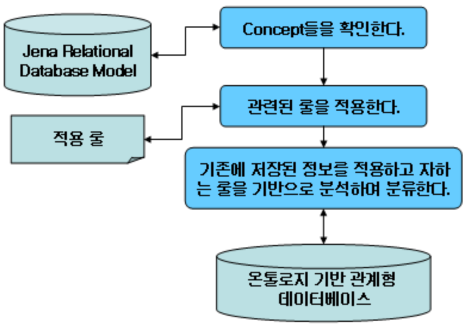

그림 6. OWL 변환기의 처리 과정

그림 6은 그림 3에서 Jena2에 이미 저장되어 있는 데이터를 온톨로지 기반 관계형 데이터베이스에 저장하기 위한 데이터를 생성하기 위한 모듈이다 이미 저장된 Jena2 관계형 데이터베이스에 저장된 구조와 데이터들을 분석하여 각 구성의 관계를 확인한 후 Jena2 관계형 데이터베이스의 데이터들을 최적화하여 온톨로지 기반 관계형 데이터베이스에 저장하기 위한 형태로 재구성 될지에 대한 세부적인 정보들을 기존JRD에 저장되어 있는 구조와 OWL 문서를 기반으로 만들어진 룰을 통하여 매핑한다. 매핑 후, 온톨로지 기반 관계형 데이터베이스에 저장하기 위한 형태로 변환하여 저장한다

4. 분석 및 비교 평가

본 논문에서는 Jena2와 본 엔진 간의 수행 속도를 정량적으로 비교해 보기 위하여 Jena2의 SPARQL 엔진에서 제공하는 university ontology를 이용하였다. 그림 7은 질의 수행 결과를 비교한 그래프이며 표1은 성능 테스트에 사용된 질의 리스트이다

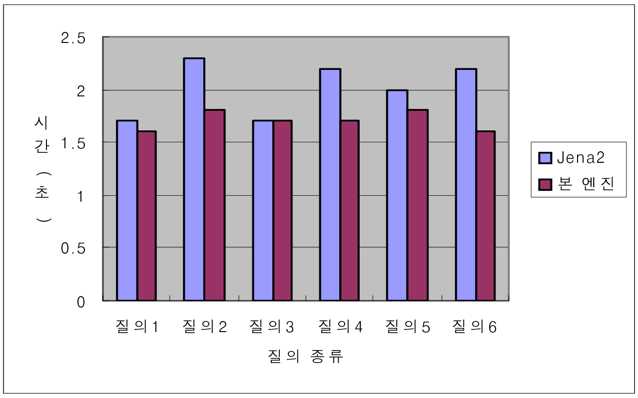

그림 7. university ontology에 질의를 수행한 결과

표 1. 테스트에 사용된 질의 유형

	OWL-QL 유형
질의1	(type ?x UndergraduateStudent) (takesCourse ?x http://www.Department0.University0.edu/UndergraduateStudent3)
질의2	(type ?x Student) (type ?y Course) (teacherOf http://www.Department0.University0.edu/FullProfessor0 ?y)
질의3	(type ?x GraduateStudent) (takesCourse ?x http://www.Department0.University0.edu/GraduateCourse3) (takesCourse ?x http://www.Department0.University0.edu/GraduateCourse49) (name ?x ?y)
질의4	(type ?x Professor) (worksFor ?x http://www.Department0.University0.edu) (name ?x ?y) (emaiAddress ?x ?z) (telephone ?x ?w)
질의5	(type ?x Professor) (worksFor ?x http://www.Department0.University0.edu) (name ?x ?y)
질의6	(type ?x UndergraduateStudent) (takesCourse ?x http://www.Department0.University0.edu/UndergraduateStudent3) (type ?y Department) (subOrganizationOf ?y http://www.University0.edu)

그림 7에서 알 수 있듯이, 수행한 모든 질의에 대해서 본 논문에서 구현한 엔진이 더 좋은 성능을 보이는 것은 아니지만 대개는 약 83%정도가 성능이 좋게 나타났다. 성능 결과를 자세히 살펴보면 질의2, 질의4, 질의6에 대한 성능이 특히 좋다. 질의2와 질의6은 여러 개의 클래스와 여러 개의 프로퍼티가 주어진 질의이고 질의4의 경우는 하나의 클래스에 대하여 질의가 주어진 경우이다.

본 논문에서는 OWL에서 정의한 Class, Individual, Property의 개념을 이용하여 OWL 문서의 데이터를 관계형 데이터베이스에 저장함으로써 단순정보검색 질의 시 Jena2에서 비정규화된 테이블 구조로 저장할 때보다 데이터의 중복성을 최소화 시키고OWL 문서의 구조에 최적화시켜 저장함으로써 질의 응답 속도를 향상 시킬 수 있었다. 또한 조인 연산 시 두 테이블의 크기로 인하여 조인비용이 발생하는 문제점을 해결함으로써 빠른 검색 및 질의 속도를 보장할 수 있었다. 마지막으로 본 엔진의 데이터 저장 형태는 rdfs:subClassOf 조건을 별개의 테이블에 저장함으로써 계층 구조에 대한 조건 판단이 요구되는 질의에 대한 처리 속도를 향상시킬 수

있다. 표2는 Jena2의 문제점을 보완하여 성능을 테스트한 결과를 보여주고 있다.

표 2. Jena2의 성능을 개선한 결과

비교 항목	Jena2	본 논문에서 성능을 개선한 결과
질의 처리 속도	느림	빠름
데이터 중복성	높음	낮음
질의 모델링의 편의성	낮음	높음
모델 변환의 용이성	낮음	높음

5. 결론 및 향후 과제

이 논문에서는 Jena2의 데이터 저장 및 질의처리 성능을 보완하기 위한 기능을 설계 및 구현하였으며 Jena2의 SPARQL 엔진에서 제공하는 university ontology를 가지고 성능을 테스트하였다. 본 논문에서는 시맨틱 웹 온톨로지 언어로 기술된 OWL 문서의 저장, 관리, 질의처리 기법을 높이는 측면에서 많은 장점을 제공한다.

현재, 본 논문에서는 rdfs:subClassOf 조건만을 다루고 있다. rdfs:subClassOf 외에도 OWL에는 정의하고 있는 다양한 제약조건(Restrictions, Axioms)들이 있다. 이러한 제약 조건을 기초로 한 추론뿐만 아니라, 관리자의 노하우를 사용자 정의 규칙으로 표현해서 추론할 수 있는 기능을 추가하여, 사용자 입장에서 보다 쉽고 효율적인 방법으로 자신의 의도에 보다 가까운 검색이 가능하도록 확장할 계획이다.

참고문헌

[1] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query, Mar. 2007.

[2] Jena -A Semantic Web Framework for Java, http://jena.sourceforge.net/

[3] RDF/XML Syntax Specification, http://www.w3.org/TR/rdf-syntax-grammar, Feb. 2004.

[4] RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema, Feb. 2004.

[5] OWL Web Ontology Language Reference,http://www.w3.org/TR/OWL-ref, Feb. 2004.

[6] DAML+OIL Reference Description W3C Note, http://www.w3.org/TR/daml+oil-reference, Dec. 2001.

[7] RDQL - A Query Language for RDF, http://www.w3.org/Submission/RDQL, Jan. 2004.

[8] Jeen Broekstra, Arjohn Kampman, Frank van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," Lecture Notes In Computer Science (LNCS), Vol. 2342, pp. 54-68, Jun. 2002.

[9] Hak Soo Kim, Hyun Seok Cha, Jungsun Kim, and Jin Hyun Son, "Development of the Efficient OWL Document Management System for the Embedded Applications," Lecture Notes In Computer Science (LNCS), Vol. 3597, pp.75-84, Jul. 2005.

[10] Myung-Jae Park and Chin-Wan Chung, "Property Based OWL Storage Schema in Relational Databases," in Technical Report, CS/TR-2005-247, Div. of Computer Science, KAIST, Dec. 2005.

[11] C. Zaniolo. 'The logical data language (idi) : An integrated approach to logic and database", MCC Technical report STP-LD-328-91, 1991.

[12] Fikes, Richard, Jessica Jenkins, and Qing Zhou."Including Domain- Specific Reasoners with Reusable Ontologies," Proceedings of the 2003 International Conference on Information and KnOWLedge Engineering. Las Vegas, Nevada, USA. June 23-26, 2003.

[13] 최종원, "온톨로지 계층 정보를 이용한 웹 서비스 발견 알고지즘", 한양대학교 컴퓨터공학과 석사학위 논문, 2004

[14] 오우택, "지식 검색과 공유를 위한 온톨로지 기반 지식 관리 시스템 구축, 한양대학교 컴퓨터공학과 석사학위 논문, 2004