

온톨로지 추론을 위한 온톨로지 컨트롤 구현

*박재훈, 전양승, 최종옥, 정석태, 정영식, 한성국
원광대학교 컴퓨터공학과

e-mail : pjh98@wonkwang.ac.kr, globaljeon@wonkwang.ac.kr, cjomail@wonkwang.ac.kr,
stjoung@wonkwang.ac.kr, ysjeong@wonkwang.ac.kr, skhan@wonkwang.ac.kr

Implementation of Ontology Control for Ontology Inference

*Jae-Hun Park, Yang-Seung Jeon, Jong-Ok Choi,
Suck-Tae Joung, Young-Sik Jeong, Sung-Kook Han
Dept of Computer Engineering, Wonkwang University

Abstract

시맨틱 웹에서 지능형 검색은 잘 구축된 온톨로지의 추론에 의해 가능하다. 온톨로지는 간단히 표현하면 단어와 관계들로 구성된 사전으로서 어느 특정 도메인에 관련된 단어들을 계층적 구조로 표현한다. 이런 구조는 다양한 방법으로 표현이 가능하며 추론에 적합한 구조인 N-TRIPLE로의 변환을 위해 JENA API를 사용한다.

I. 서론

현재의 웹은 사람이 보고 잘 이해할 수 있도록 하기 위한 브라우저의 디스플레이 또는 레이아웃 기술에 초점을 맞추고 있다. HTML을 이용한 이러한 표현방식은 문서의 내용과 의미를 나타내는 시맨틱 정보를 표현하기가 어려우며, 따라서 사람이 아닌 프로그램 또는 소프트웨어 에이전트가 자동으로 문서로부터 의미를 추출하기가 어렵다.

이러한 문제점을 해결하기 위해서 1990년대 말에 W3C(World Wide Web Consortium)에서 시맨틱 웹(Semantic Web)을 제안하였다.

Tim Berners Lee에 의해 제안된 시맨틱 웹은 컴퓨터가 웹상의 정보를 이해하고, 정보를 창출할 수 있는 웹 환경으로서, 정보의 탐색과 의사결정이 인간이 아닌 컴퓨터가 할 수 있도록 만들어진 Web 환경이다. 즉, 메타데이터의 개념을 통하여 웹 문서에 시맨틱 정보를 덧붙이고, 이를 이용하여 에이전트가 의미 정보를 자동으로 추출할 수 있는 패러다임을 조성하는 것이다. 부수적으로 의미 정보의 자동 추출뿐 아니라 정보의 확장이나 공유 등도 가능하다.

시맨틱 웹의 목적은 웹에 있는 정보를 컴퓨터가 쉽게 이해할 수 있도록 도와주는 표준과 기술을 개발하여 시맨틱 검색, 데이터 통합, 네비게이션, 업무의 자동화 등을 지원하는 것이다. 시맨틱 웹에서 이러한 기능을 지원하기 위해서는 컴퓨터의 지능적인 정보처리가 가능토록 웹 문서 내에 지식 표현을 위한 온톨로지를 삽입하고, 지식 간의 관계를 설정하며 추론 규칙을 포함 시켜야한다. 이를 통해서 사용자가 원하는 정보를 정확하게 전달해 줄 수가 있다.

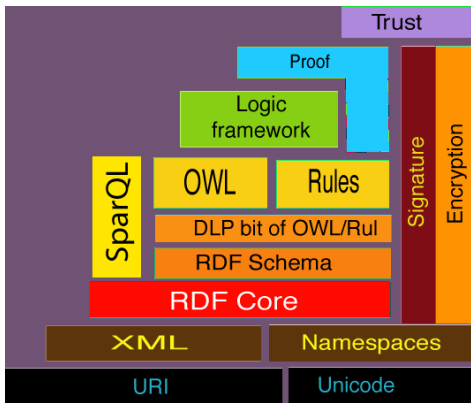


그림 1. 2005년판 시맨틱웹 구조

II. 관련 연구

2.1 온톨로지 개념

시맨틱 웹에서의 검색을 위해 온톨로지는 가장 중요한 요소이다. 온톨로지의 개념을 이해하고 다양한 온톨로지 언어를 이용해 온톨로지의 설계 및 구축 작업이 선행되어야 시맨틱 웹 검색이 가능하다.[4]

온톨로지는 간단히 표현하면 단어와 관계들로 구성된 사전으로서 어느 특정 도메인에 관련된 단어들을 계층적 구조로 표현하고 추가적으로 이를 확장할 수 있는 추론 규칙을 포함한다. 그리고 웹 기반의 지식 처리나 응용 프로그램 사이의 지식 공유, 재사용을 가능하게 하는 중요한 요소이다. 온톨로지에는 계층분류와 추론규칙에 대한 정의가 포함된다. 계층분류는 객체의 클래스와 서브클래스, 그들 간의 관계를 정의한다. 그리고 온톨로지를 표현하기 위해 스키마와 구문구조 등을 정의한 언어가 온톨로지 언어이다[2][4][5].

2.2 온톨로지 언어

2.2.1 RDF

특정 자원에 대한 메타 데이터를 기술하는 XML 기반의 프레임워크이다. RDF는 레코드(record)를 하나의 기술 단위로 취급해온 기존의 방식과는 달리 자원, 속성, 속성 값을 하나의 단위로 취급하는 이른바 “Triple” 개념이 그 핵심이다. 자원 속성 표현의 세분화로 인해 자원에 대한 좀 더 정교한 기술이 가능해지고, 자원들 간의 관계 설정이 속성(Predicate)를 통해 무한으로

가능하게 되어진다. 각각의 자원들은 URI를 통해 고유 식별자를 가지게 된다. 그리고 자원을 기술하는 속성 명 또한 고유한 URI를 통해 XML Namespace에 정의되어진 속성을 사용함으로써 상호간 의미 충돌을 막는다. 속성의 값으로는 다른 URI가 지정될 수 있으며, 속성 값으로 지정된 자원 역시 다시 기술의 대상이 되기 때문에 그 자원에 대한 속성과 속성 값을 다시 부과할 수 있다[1].

| | |
|----------------------|--------------------------------|
| Subject (Resource) | http://www.w3.org/Home/Lassila |
| Predicate (Property) | Creator |
| Object (literal) | "Ora Lassila" |



그림 2. RDF의 Triple 구조

2.2.2 TopicMap

TopicMap은 ISO를 중심으로 한 Semantic Web 기술로 ISO/IEC 13250 표준으로 지식 표현 기술(Knowledge Representation)의 표준이다. RDF와 마찬가지로 XML 기반의 표준 기술 언어인 XTM(XML Topic Maps)라는 언어를 사용하여 정보와 지식의 분산 관리를 지원한다. 이는 정보 자원의 구성, 추출, 네비게이션에 관계하는 새로운 패러다임으로, 정보와 지식 관리를 위해 최적화된 표현 양식을 제공하고 있다. Topic Maps는 지식층과 정보층의 이중 구조를 나타내는데, 지식층은 상위 계층으로 토픽(topic)과 토픽 간의 연계(Association)로 구성된다. 토픽은 특정 주제를 나타내는 표현이고, 연계는 주제들 간의 관계를 나타내는 표현이다. 정보층은 디지털 콘텐츠를 나타내며, 이들 지식층과 정보층은 어커런스(Occurrence)를 통해 상호 연결이 이루어진다.

2.2.3 OWL

OWL은 표현력에서는 우수하지만 처음 접근하는 이용자, 개발자 및 개발도구 지원 등에 용이하지 않은 측면이 있어 이를 개선하여 보편적인 이용자를 확보하기 위해 좀더 간결하고 사용하기 쉬운 언어가 필요하게 되었다. 이러한 목적으로

개발된 DAML+OIL의 새로운 개정 언어이다. 또한 OWL의 서브셋으로 제안된 OWL Lite는 OWL과 DAML+OIL의 공통적이고 유용한 부분들을 간추려 만들어졌다. 기능적인 면에서 볼 때 웹 응용프로그램을 지원하기 위해 간단하면서도 RDFS에 비해 풍부한 표현력을 가지는 언어이다.

OWL에서 제안하고 있는 유형은 OWL Lite, OWL DL, OWL Full이 있는데 OWL Lite는 시소러스에 접근이 용이하고 단순한 면을 강조하여 웹 응용에 강점을 갖고 있으며 OWL DL은 Lite보다 좀더 논리적인 표현을 위한 온톨로지이다. 마지막으로 OWL Full은 표현력에 있어서 가장 풍부하며 RDF의 자유로운 구문을 모두 허용하고 있다.

즉, OWL Full은 DL과 Lite의 모든 기능을 포함하는 관계이며 유효성 및 호환성에 있어 가장 완벽하지만 웹 온톨로지 언어로써 이용하기에 Lite가 용이하며 시소러스의 개념에 접근하기 쉽다는 점에서 웹 응용에 많이 사용된다.

```
<owl:Ontology rdf:about="" />
<owl:Class rdf:ID="Family">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Movie" />
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    가족 영화에 대한 정보를 표현한다.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Animation">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      애니메이션에 대한 정보를 표현한다.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Movie" />
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="OST">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Music" />
    </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
```

그림 3. OWL 문서

2.3 온톨로지의 표현

온톨로지는 RDF/XML, N3, N-TRIPLE 등 다양한 방법으로 표현이 가능하며 추론을 위해 OWL-Lite로 작성된 온톨로지는 Subject, Predicate, Object로 구성되는 Triple 형태로 변환해야 한다. 변환 방법은 수작업이 아닌 온톨로지를 액세스하고 컨트롤 할 수 있는 JENA API를 통해 가능하다.

2.4 온톨로지 추론

현재 시맨틱 웹에 적용하려고 시도하고 있는

추론 기술은 논리기반(Logic based)의 추론과 룰 기반(Rule based)의 추론이 있다. 사실들과 일반적인 규칙에 근거하여 추론하는 논리기반의 추론 엔진을 시맨틱 웹에 적용하려면, 예를 들면 FOL 기반 추론은 단조 추론(monotonic reasoning)만 가능하다거나, true/false 값만 대상으로 다루어야 한다거나, 또는 학습(learning) 기능이 없다는 것과 같은 몇 가지 단점들이 있다. 또한 룰 기반의 추론은 똑같은 문제가 다시 주어져도 똑같은 양의 작업(추론)을 하여 같은 결론을 얻거나, 룰 시스템에서 제공하는 지식의 영역을 벗어나는 문제에 대해서는 전혀 도움을 줄 수 없다는가, 또한 룰 시스템을 구축하고 유지하는데 많은 시간과 노력이 든다는 단점이 있다.

III. 시스템 개발 환경

3.1 개발환경

본 논문의 구현은 온톨로지의 추론을 위한 온톨로지 컨트롤이다. 현재 온톨로지를 액세스하고 컨트롤 할 수 있는 API로 HP에서 만든 JENA API를 활용했다.

표 1. 개발환경

| 온톨로지 컨트롤 구현 개발 환경 | |
|-------------------|-----------------|
| 개발언어 | Java (JDK1.4.2) |
| 개발도구 | Eclipse 3.1 |
| API | Jena 2.2 |
| Web Server | Apache 2.0 |
| Servlet | Tomcat 5.0 |

기본적인 실행결과와는 이클립스의 커맨드 윈도우를 통해 확인할 수 있지만 구현 결과의 명확성을 위해 간단한 JSP 파일을 제작했다.

3.2 개발 방법

온톨로지는 추론을 위해 N-TRIPLE 형태로 변형해야 한다. JENA API에서 제공하는 OntModel은 온톨로지 파일을 N-TRIPLE 형태로 변형이 가능하다. 변형된 온톨로지는 StmtIterator 타입으로 반환해 Triple 요소를 추출해 출력한다.

IV. 구현

구현에 사용한 온톨로지 파일은 camera이다. getModelNT 함수는 온톨로지의 URI와 요청타입의 파라미터를 이용해 N-TRIPLE로 변환된 온톨로지를 StmtIterator로 반환한다[7].

```

public class StmtTest {
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        String ONT_URI = "http://www.xfront.com/owl/ontologies/camera/#";
        String requestType = "N-TRIPLE";

        readOntModel rom = new readOntModel();

        // N-Triple 변환 Method 호출 (ReturnVar : StmtIterator)
        StmtIterator iter = rom.getModelNT(ONT_URI, requestType);

        System.out.println("#####");

        // ReturnVar인 iter의 값을 화면에 출력
        while (iter.hasNext()) {
            Statement stmt = iter.nextStatement(); // get next Statement
            Resource subject = stmt.getSubject(); // get the subject
            Property predicate = stmt.getPredicate(); // get the predicate
            RDFNode object = stmt.getObject(); // get the object

            System.out.print(subject.toString());
            System.out.print(" " + predicate.toString() + " ");

            if (object instanceof Resource) {
                System.out.print(object.toString());
            } else {
                // object is a literal
                System.out.print(" \"" + object.toString() + "\"");
            }
            System.out.println(" .");
        }
    }
}
    
```

그림 4. N-TRIPLE 출력 소스

Iterator 타입으로 반환된 온톨로지는 Triple 요소인 subject, predicate, object로 나누어 커맨트 윈도우에 출력한다.

```

1.18201 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18202 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18203 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18204 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18205 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18206 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18207 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18208 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18209 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18210 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18211 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18212 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18213 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18214 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18215 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18216 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18217 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18218 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18219 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
1.18220 http://www.owl-ontologies.com/unnamed.owl#chessboard http://www.w3.org/2000/01/rdf-schema#Property http://www.owl-ontologies.com/unnamed.owl#
    
```

그림 5. N-TRIPLE 출력 화면

V. 결론 및 향후 연구 방향

현재 사용한 온톨로지는 비교적 작은 규모의 온톨로지이다. JENA API는 대형 온톨로지에 대

한 처리 부분에서 퍼포먼스가 좋지 않은 편이다.

이러한 단점을 보완하기 위해 온톨로지를 파일 처리가 아닌 데이터베이스에 온톨로지를 임포트한 후 JENA API가 데이터베이스의 온톨로지를 접근하도록 하는 방법을 활용했다. 아직 제대로 구축된 온톨로지가 없어 비교 테스트를 할 수 없었다는 점이 아쉬웠지만 데이터베이스를 활용한 구현이 더욱 좋은 퍼포먼스를 보였다.

감사의 글

본 연구는 학술진흥재단 기초과학연구사업 중 지방 연구중심대학 육성사업인 헬스케어 기술개발사업단의 지원에 의해 수행되었으며 이에 감사드립니다.

참고문헌

- [1] 이재호, “시맨틱 웹의 온톨로지 언어”, 정보과학회지, 제21권, 제3호, pp. 18-27, 2003
- [2] 양정진, “시맨틱 웹에서의 온톨로지 공학”, 정보과학회지, 제21권, 제3호, pp.28-35, 2003
- [3] 장명길 외, “의미기반 정보검색”, 정보과학회지, 제19권, 제10호, pp. 7-18, 2001
- [4] 옥철영, “한국어정보처리와 온톨로지”, 2004 한국어정보처리연구회 동계 튜토리얼 자료집
- [5] 최호섭, 옥철영, 김창환, 왕지현, 장명길, “질의응답시스템을 위한 백과사전 기반 지식베이스와 온톨로지”, 제15회 한글 및 한국어 정보처리학술대회 자료집, pp. 177-183, 2003
- [6] 김현희, 안태경, “온톨로지를 이용한 인터넷웹 검색에 관한 실험적 연구”, 정보관리학회지, 제20권, 제1호, pp. 417-455, 2003
- [7] 정도현, “시맨틱웹을 위한 온톨로지 언어와 구현사례 연구”, 정보관리연구, 제34권, 제3호, pp. 87-109, 2003
- [8] Berners-Lee, T., Hendler, J., Lassila, O., “The Semantic Web”, Scientific American, 2001
- [9] <http://jena.sourceforge.net>