

논문 2008-45CI-6-23

온톨로지 저작 도구 개발에 관한 연구

(A Study on the Development of Ontology Management Tool)

김 원 필*, 김 정 호*

(Won-Pil Kim and Jeong-Ho Kim)

요 약

오늘날, 웹 데이터의 의미적 처리를 위한 시맨틱 웹에 대한 많은 연구가 진행 중이다. 시맨틱 웹 환경을 실현하기 위해서 가장 중요한 작업은 객체들의 개념과 개념들 사이의 관계를 정의하는 온톨로지의 구축이다. 온톨로지 구축을 위해 OWL, RDF(S), DAML+OIL 등과 같은 언어들에 기반하여 온톨로지를 구축하기 위한 Protege, KAON, OIEd 등의 온톨로지 저작 도구들이 개발되었다. 이처럼 많은 온톨로지 구축 언어와 도구가 개발되었지만, 온톨로지 이론에 대한 이해부족, 구축 언어 어휘 사용의 어려움 그리고 저작 도구 사용의 어려움 등의 이유로 온톨로지 저작 도구의 활용은 온톨로지 전문가 및 연구자들로 제한되어 있다. 그리고 워드넷과 같은 범용의 대형 온톨로지를 재사용하기 위한 연구 또한 부족한 실정이다. 그러므로 본 연구에서는 온톨로지 전문가들을 비롯하여 온톨로지 구축을 원하는 일반 사용자들도 쉽게 온톨로지를 구축할 수 있는 OWL 온톨로지 저작 도구를 설계하고 구현하였다. 본 논문에서는 온톨로지 저작 도구의 개발에 필요한 주요 모듈에 대한 내용들을 소개함으로써 저작 도구 개발에 도움이 되고, 일반인들의 온톨로지에 대한 이해를 도와줌으로써, 본 시스템을 사용하여 쉽게 온톨로지를 구축할 수 있기를 기대한다.

Abstract

Nowadays, the study on the semantic web has been actively progressing for processing the web data semantically. For actualizing the semantic web environment, the core task is to build the ontology that defines the concepts and relations between concepts about the all things. Many ontology languages such as OWL, RDF(S), DAML+OIL were developed for building the ontology. And the many ontology tools were also implemented based on them. Although, many language and tools were researched, the practical use of the ontology tools is limited to the experts and researchers about the ontology because of the difficulty of the vocabulary, weak understanding about the ontology theory and the difficulty of the use of the ontology tools. And there are no studies on the reuse of constructed huge ontology.

Therefore, in our study we design and implement the OWL ontology management tool that both the ontology experts and general users who want to build the ontologies are able to construct the ontology easily. In this paper, we introduce the main modules used in our tool and features of our tool.

Keywords: 시맨틱 웹, 온톨로지, 온톨로지 에디터

I. 서 론

컴퓨터가 객체들의 정의와 객체간의 관계를 이해하고, 그 속에서 의미를 파악하여 처리하는 시맨틱 웹에 대한 연구가 활발히 진행되고 있다. 시맨틱 웹에서 핵심이 되는 연구는 컴퓨터가 세상의 모든 개념들을 어떻게 처리하느냐는 것이다. 온톨로지는 개념들을 명확하게 정의하고, 개념들 사이의 관계를 정형화함으로써, 사람과 기계

가 함께 이해할 수 있고, 새로운 의미를 도출할 수 있다. 그렇지만 각각 소규모의 도메인 개념들에 대해서만 정의 하더라도 온톨로지의 크기는 비대해지며, 온톨로지 언어의 어휘 사용의 복잡성과 온톨로지 구축의 어려움 등의 이유로 시맨틱 웹의 실현이 늦어지고 있다.^[8]

RDF(S)와 OWL은 2004년 2월에 W3C에 의해 국제 표준으로 제정되어 현재 가장 널리 사용되고 있는 온톨로지 언어이다. 그리고 이 언어들에 기반한 온톨로지 저작 도구 역시 Protege, OIEd, KAON등 50여 가지 이상 개발되었다. 이와 같이 많은 온톨로지 구축 언어와 저작 도구가 개발되었지만, 온톨로지 언어의 어휘

* 정회원, 조선이공대학교 U-사이버보안과 교수
(chosun college of science & technology)

접수일자: 2008년6월5일, 수정완료일: 2008년10월23일

사용의 어려움과 온톨로지 이론에 대한 이해 부족 그리고 온톨로지 저작 도구 사용의 어려움 등의 이유로 온톨로지 저작 도구의 활용은 온톨로지 전문가 및 연구자들로 제한되어져 왔다. 그러므로 본 연구에서는 온톨로지 전문가들을 비롯하여 온톨로지를 구축하고자 하는 일반 사용자들도 쉽게 온톨로지를 구축할 수 있는 OWL 온톨로지 저작 도구를 설계하고 구현하였다.

II. 관련연구

온톨로지 저작 도구 개발에 필요한 기술을 파악하고, 향상된 인터페이스 구성을 위하여 기존의 국내/외 온톨로지 저작 도구들을 조사하였다.

2.1 KAON

독일의 KARlsruhe 대학과 FZI 연구회에서 공동으로 제작한 KAON은 웹을 기반으로 구축되어졌으며, 서버에 온톨로지들을 두고 여러 사용자가 동시에 접속하여 온톨로지를 생성, 편집할 수 있다.^[9]

2.2 Protege 3.1

미국의 Stanford 대학에서 제작한 Protege는 현재 온톨로지 구축에서 가장 널리 활용되고 있다. 트리 구조를 이용하여 자원들의 상/하위 계층을 표현하여, 쉽게 자원의 정보를 확인하고 수정할 수 있다.^[10]

2.3 OILED

OILED는 영국의 Manchester 대학에서 DAML+OIL 언어에 기반하여 제작한 단순한 온톨로지 편집기이고, 다른 온톨로지 도구에 비해 쉬운 인터페이스로 구성되어 있다. OILED의 주요 기능은 FaCT 추론기를 이용하여 온톨로지들을 분석할 수 있고, DAML+OIL을 SHIQ Description Logic으로 변환하여 온톨로지 일관성을 체크할 수 있다.^[11]

2.4 OntoEdit

독일의 Ontoprise에서 개발한 OntoEdit는 OilEd와 같이 DAML+OIL언어에 기반하고 있으며, 사용자 인터페이스를 제공하지만 사용하기에 복잡하다. 트리로 구성된 계층 내에서 Copy and Paste가 가능하여 개념들을 재조직 할 수 있다.^[12]

2.5 OWL Editor

국내 기업인 씨컴테크사에서 개발된 OWL Editor는 순수 OWL파일을 처리하고 있으며, 텍스트 편집창 인터페이스를 제공한다. 하지만 자원의 추가, 수정 및 삭제 등의 편집은 다이얼로그를 띄워 이루어진다.^[13]

2.6 OWL 그래픽 편집기

OWL 그래픽 편집기도 국내 기업인 케이테크사에서 개발하였다. OWL 그래픽 편집기는 그래프 표현 방식을 이용하여 온톨로지를 모델링하고, DAML+OIL과 OWL파일을 시각화하여 다룰 수 있다. 그리고 RDF(S), DAML+OIL, OWL로 작성된 문서의 파싱과 Serialization을 지원한다.^[14]

앞에서 소개한 국내/외에서 개발된 온톨로지 저작 도구들은 모두 온톨로지에 대한 전문 지식과 온톨로지 구축 언어의 어휘를 모두 알고 있는 전문가와 연구자들을 대상으로 제작된 것이다. 또한 모든 온톨로지 구축에 있어서 사람에 의해 처음부터 끝까지 제작되어야 하는 단점이 있다.

본 연구에서는 기존 도구들의 주요특징들을 최대한 활용하고 또 기존 도구의 한계를 해결하기 위하여 온톨로지 저작을 위한 새로운 시스템을 설계하고 개발하였다.

III. 본 시스템 구성

3.1 전체 시스템 구성

본 시스템의 구축목적은 온톨로지 전문가들과 온톨로지 구축을 필요로 하는 일반 사용자들에게도 편의성을 제공하는 일관된 온톨로지를 구축하고 관리하는 것이다. 그래서 사용자가 다루기 쉬운 인터페이스로 설계하였으며, 핵심 기능은 온톨로지 구축과 편집, OWL 온톨로지 문서 파싱, OWL 온톨로지 문서로의 표현, 사용자에게 온톨로지 언어에 대한 사용자 지침을 제공 그리고 기존에 구축되어진 대형의 온톨로지(WordNet)에 기반한 자동으로 도메인 온톨로지를 구축할 수 있는 기능이다. 본 시스템의 전체적인 구조는 그림 1과 같다.

사용자 인터페이스는 기존의 온톨로지 도구들에서 느낄 수 있는 복잡함과 온톨로지를 구축하는데 어려움을 최소화하기 위해, 기본적으로 필요한 버튼, Tree View와 편집창으로 구성하였다. 온톨로지를 새롭게 작성할 수도 있지만, OWL 파서가 미리 작성된 OWL 온톨로지를 파싱하여 편집할 수 있으며, 이를 다시

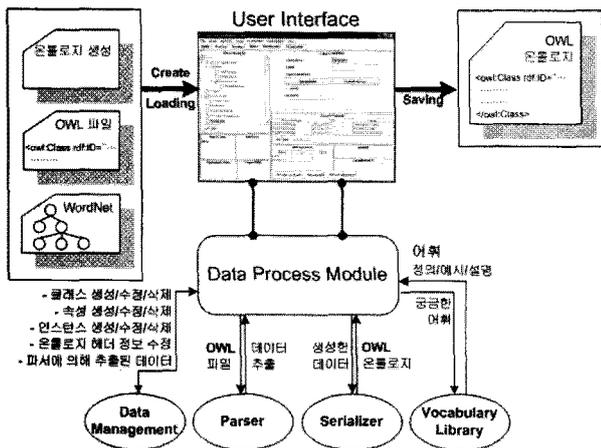


그림 1. 시스템 구조
Fig. 1. Architecture of system.

표 1. 시스템의 특징
Table 1. Feature of System.

1. 트리(Tree)를 이용한 개념들 사이의 계층도 : 개발자들은 원하는 자원으로 바로 이동이 가능하며, 쉽게 관리와 조작이 가능
2. 클래스와 속성의 정의를 통해, 인스턴스를 생성할 때 자동으로 속성의 값들을 정의 할 수 있는 테이블(Table) 구성
3. 자원 정의와 자원들 사이의 관계 정의를 간편하게 하기 위한 폼(Form) 구성
4. 자원을 정의할 때, OWL 어휘 사용의 어려움을 극복하기 위한 OWL 어휘 라이브러리(Library) 제공
5. 워드넷(WordNet)에 기반한 특정 도메인에 대한 자동 온톨로지 구성
6. 데이터들의 효과적인 관리와 빠른 구문 분석에 의한 신속한 파싱 모듈 개발
7. 독립적인 플랫폼(JDK 1.5 이상)

Serializer는 OWL 온톨로지 문서로 저장할 수 있는 구조로 되어 있다. 표 1은 도구의 주요 특징들을 나타낸 것이다.

온톨로지 저작 도구 개발에 필요한 전반적인 기능들은 JENA(A Semantic Web Framework for Java)^[1], SOFA(Simple Ontology Framework API)^[2]등의 라이브러리 형태로 제공되고 있고, 이미 개발된 다수의 온톨로지 저작 도구들은 이러한 라이브러리들을 채택하여 개발되어졌다.

그렇지만 본 시스템은 독자적인 기술력을 확보하여, 온톨로지 저작 도구를 최대한 활용하기 위해 다른 외부의 어떠한 라이브러리도 사용하지 않고 제작하는데 큰 의미가 있다.

3.1.1 사용자 인터페이스(User Interface)

본 시스템은 4개의 Tab View(Class, Property, Instance, Serialization)로 구성되어 있다. Class, Property와 Instance Tab에는 자원들의 전체적인 상/하

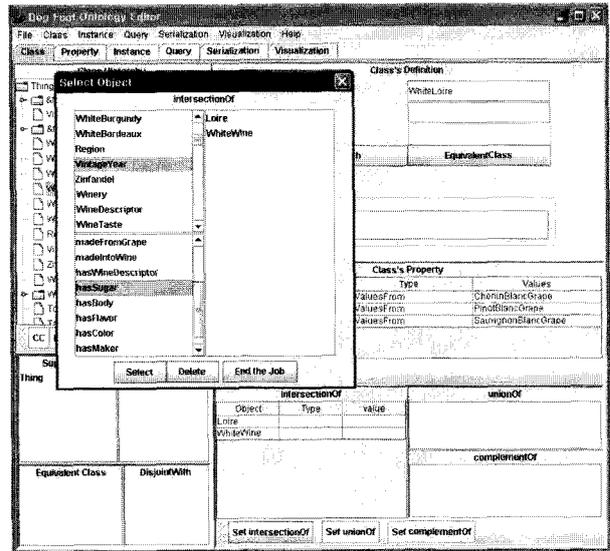


그림 2. 사용자 인터페이스
Fig. 2. User Interface.

위 관계를 한눈에 알아볼 수 있게 트리형식으로 구성하여 계층을 표현하였다. 특히 Class Tab은 선택된 자원과 직접 연결된 상/하위 자원, equivalentClass와 disjointWith 관계를 바로 파악 할 수 있도록 설계하였고, 클래스 정의, 지역 속성(Local Restriction), intersectionOf, unionOf와 complementOf를 쉽게 설정할 수 있도록 간단한 버튼들과 설정창으로 구성하였다. Property와 Instance Tab 또한 사용자의 편의 위주로 설계하고 구축하였다. 그림 2는 특정 클래스의 intersectionOf를 다이얼로그를 기반으로 설정할 수 있는 인터페이스를 보여주고 있다.

Serialization Tab에서는 버튼을 클릭 함으로써 작성된 온톨로지에 대한 OWL 구문으로 표현함과 동시에 OWL파일로 저장할 수 있도록 하였다.

3.1.2 자원(Resource : Ontology Header, Class, Property, Instance) 관리

온톨로지 도구에서 가장 핵심이 되는 부분이 효율적이고 체계적인 자원관리이다. 이를 위해 각 자원의 데이터 셋(data set)을 구성하여야 하고, 이 데이터들을 일관적으로 관리하기 위한 모듈을 따로 두어야 한다. 본 시스템에서는 Ontology Header, Class, Property, Instance로 나누어 데이터 셋을 두었다. Ontology Header는 전반적인 온톨로지에 대한 정보를 저장하는 부분이고, Class 데이터 셋은 상위클래스, 클래스 이름, 라벨(label), 지역속성(Local Restriction)등의 클래스 정의에 필요한 것들을 비롯하여, 클래스를 통해 생성된

표 2. Class 데이터 셋 정의 부분
Table 2. Definition part of Class Data Set.

```

class DataClass // Class 데이터 셋
{
    String superClass = new String();
    String nameClass = new String();
    String ClassLabel = new String();
    String ClassComment = new String();
    Vector ClassDisjointWith = new Vector();
    Vector ClasshasInstance = new Vector();
    boolean oneOf = false;
    Vector equivalentClass = new Vector();
    String DeprecatedClass = new String();
    Vector intersectionOf = new Vector();
    Vector unionOf = new Vector();
    Vector complementOf = new Vector();
    Vector ClasshasProperty = new Vector();
}

class DataIntersectionOf // intersectionOf
{
    String object = new String();
    String typeOfIntersectionOf = new String();
    String valueOfIntersectionOf = new String();
}

class LocalRestrictionProperty // 지역 속성
{
    String PropertyName = new String();
    String hasValueType = new String();
    String hasValues = new String();
}
    
```

표 3. Class 데이터 셋 관리 모듈
Table 3. Management module of Class Data Set.

함수	기능
addObject()	클래스 생성
addObjectComment()	comment 추가
addDeprecatedClass()	deprecatedClass 설정
addObjectOneOf()	oneOf 옵션 설정
addObjectLabel()	label 설정
addDisjointWith()	disjointWith 클래스 설정
addEquivalentClass()	equivalentClass 설정
addIntersectionOf()	intersectionOf 설정
addComplementOf()	complementOf 설정
addUnionOf()	unionOf 설정
addProperty()	지역속성 추가
addInstance()	인스턴스 생성
modifyInstance()	인스턴스 이름 수정
modifyProperty()	지역속성 수정
modifyObject()	클래스 이름 수정
deleteIntersectionOf()	intersectionOf 중 특정 객체 삭제
deleteUnionOf()	unionOf 중 특정 객체 삭제
deleteComplementOf()	complementOf 중 특정 객체 삭제
deleteInstance()	인스턴스 삭제
deleteProperty()	지역속성 삭제
removeObject()	클래스 삭제
getObject()	특정 클래스의 모든 정보 반환
getTotClassName()	모든 클래스의 이름 반환
getSubObject()	특정 클래스의 하위 클래스 이름 반환
getClassCount()	모든 클래스의 갯수 반환
getInstance()	특정 클래스가 갖는 인스턴스 이름 반환
compareOverlap()	생성된 클래스들 중 겹침 감지

인스턴스, 미리 정의된 속성과 관계를 설정하는 부분으로 데이터를 구성하였다. Property와 Instance 데이터 셋 또한 각각 속성과 인스턴스 정의 부분과 클래스와 관계 설정 부분으로 구성되어 있다. 표 2는 본 시스템에서 정의하고 있는 Class 데이터 셋 부분이다.

본 시스템은 모든 데이터를 메모리에 로드시켜 처리한다. 그러므로 메모리의 낭비를 최소화하기 위해 Class 데이터 셋에서는 모든 클래스에 존재하지 않을 intersectionOf와 지역 속성을 따로 두어 관리하고, Class 데이터 관리 모듈의 접근을 줄여 거대한 온톨로지를 제작할 경우에도 시스템에 걸릴 과부하 없이 데이터 처리 속도를 빠르게 하였다.

위와 같이 정의된 Ontology Header, Class, Property, Instance의 데이터 셋은 일관적으로 관리되며, 본 시스템의 인터페이스에 의해 접근된다. 데이터 셋을 관리하기 위해서는 각각의 관리 모듈이 필요하다. 각 모듈에는 각각의 데이터에 대한 추가, 수정, 삭제로 이루어져 있고, 각 데이터 셋에 대한 정보를 얻을 수 있는 부분이 포함되어 있다. 개발자가 본 시스템의 인터페이스를 이용하여 데이터를 조작하는 것은 표 3과 같은 각각의 해당 모듈을 통해 관리되도록 하였다. 표 3은 Class 데이터 셋을 관리하는 모듈에 대한 함수와 기능들을 나타낸 것이다.

3.1.3 Serialization(UnParsing)

Serialization은 시스템을 통해 구성된 자원들과 자원들 사이의 관계들을 형식적인 온톨로지 구문으로 작성

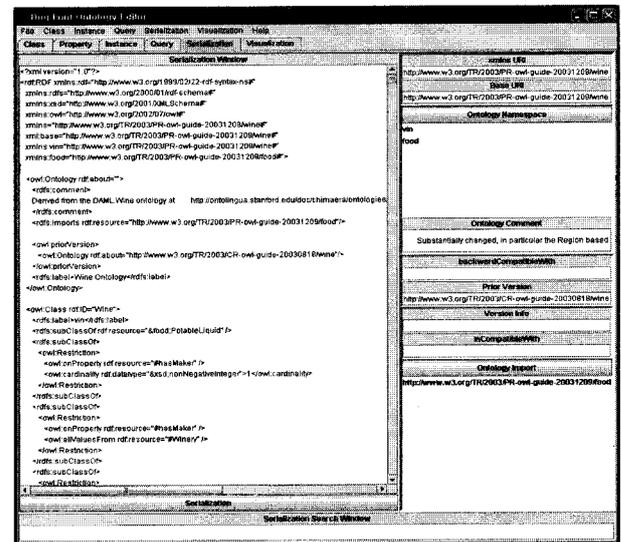


그림 3. 시스템의 Serialization 이미지
Fig. 3. Serialization image of system.

하는 것을 말한다. 본 시스템의 Serializer는 OWL 구문으로 작성되도록 제작하였으며, Serialization Tab에서 온톨로지 헤더 정보를 수정할 수 있게 하였고, 버튼 하나로 Serialization과 동시에 OWL 파일로 저장할 수 있게 하였으며, Serialization된 OWL 구문에서 특정 객체를 검색할 수 있는 기능을 구현하였다.

그림 3은 온톨로지에서 가장 많이 활용되고 있는 Wine 온톨로지를 본 시스템을 통해 Serialization한 것이다. 그림 3과 같이, 화면의 오른쪽에서 온톨로지 헤더의 정보를 기입하고 수정할 수 있으며, 하단의 검색 바를 이용하여 객체 검색을 할 수 있게 구성하였다.

3.1.4 Parsing

파싱은 Serialization과 반대의 기능으로, 기존에 작성된 온톨로지 구문을 분석하는 것을 말한다. 본 시스템은 현재 OWL로 작성된 구문의 분석이 가능하며, CPU의 작업량을 줄여 빠른 분석이 가능하도록 2단계의 과정을 거치도록 구성하였다. 1단계에서는 OWL 구문을 토큰(token) 단위로 나누는 작업을 한다. 그리고 2단계에서는 1단계에서 나뉘어진 토큰을 모아 어휘를 분석한 후 자원들을 각 데이터 셋으로 저장을 한다. 현재 대부분의 온톨로지 저작 도구들의 파싱은 자신의 프로젝트 파일을 생성하여 다루기 때문에, 특정 도구에서 생성된 온톨로지는 해당 도구에서만 다루어지는 단점이 있다. 그리하여 동일한 온톨로지 언어로 작성되어 있더라도 통일성이 없다. 그렇지만 본 시스템은 일반적인 OWL 온톨로지를 다루고 있고, 다른 도구에서 Serialization된 OWL 문서라 할지라도 본 도구를 통하여 일관된 OWL 온톨로지를 제작할 수 있다.

3.1.5 구축 자동화

본 시스템에서는 온톨로지 구축에 있어서 기존의 개념 정의 및 데이터베이스 지식 표현의 내용에 대하여 재 사용성을 높혀 온톨로지를 구축할 수 있는 방법론을 적용하였다. 본 시스템에서는 범용의 대형 온톨로지인 워드넷을 입력형식으로 받아서 이를 OWL 온톨로지 형식으로 표현하고자 하는 방법론을 제안하고 작성하였다. 워드넷은 데이터베이스 형식의 대형 온톨로지로서 현재까지 가장 널리 사용되고 있는 범용의 온톨로지이다. 워드넷 내에서는 개념간의 관계를 표현하기 위하여 다양한 관계를 기호로 표현하고 있다. 이 중에서 본 시스템은 워드넷 내의 명사에 대한 부분만을 처리하도록 한다. 명사는 !(반의), @(상위), ~(하위), synset number

표 4. OWL 어휘 매칭

Table 4. Matching of OWL vocabulary.

워드넷	OWL 어휘
! @ ~ 동일 synset ID	owl:disjointWith owl:superClassOf owl:subClassOf owl:equivalentClass

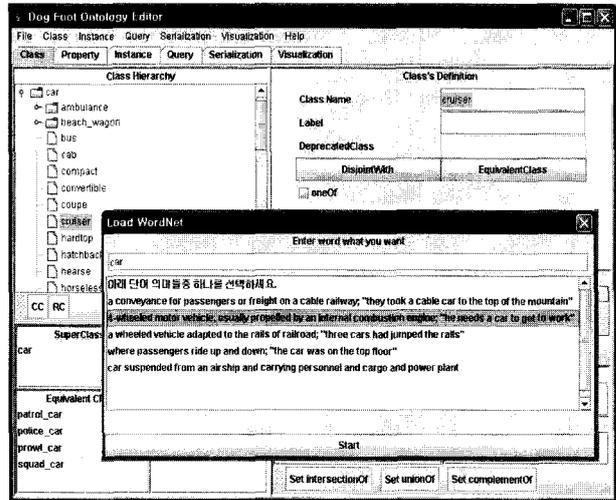


그림 4. 워드넷 자동입력 인터페이스

Fig. 4. Interface of WordNet automatic-input.

(동의)의 관계들로 개념들의 관계를 표현되고 있다. 이들 각 관계들은 OWL 어휘들과 표 4과 같이 매칭될 수 있다.

이러한 매칭을 이용하여 데이터베이스 형식의 워드넷을 OWL 온톨로지로 변환하여 쉽게 온톨로지를 구축하고, 더 다양한 개념을 온톨로지 저작 도구를 사용하여 추가, 관리 할 수 있게 된다. 그림 4는 본 시스템에서 워드넷을 OWL 온톨로지로 변환하여 표현한 내용이다.

3.1.6 RDF/OWL 어휘 라이브러리

OWL 온톨로지 정의에 사용되는 어휘에는 [5]에 의하면 rdfs:Class의 하위 클래스들만 17가지, rdf:Property의 하위 속성들로 23가지가 존재한다. 그리고 OWL 어휘의 근간이 되는 RDF(S)의 어휘^[6] 29가지까지 하면 모두 69가지가 된다.

일반 사용자가 이 어휘들의 정확한 기능까지 모두 숙지하기란 쉽지가 않다. 게다가 이들 어휘에 대한 설명이 영문으로 되어 있고, 대부분 전문 용어로 구성되어 있어 온톨로지 언어 어휘에 대한 전문가를 제외하고는 이를 참고하기란 힘든 일이다. 그래서 이들 어휘에 대한 내용을 본 연구에서는 국내 표준으로 제정

표 5. 온톨로지 저작 도구 비교
Table 5. Comparison of Ontology management tool.

저작 도구	사용자 인터페이스	편집 방식	파일 형태	플랫폼독립성	파싱 속도	어휘라이브러리	구축 자동화
KAON	O	Graph	kwsProject File	O	느림	×	×
Protege 3.0	O	Tree & Form	pprj ProjectFile	O	느림	×	×
OilEd	O	Tree & Form	DAML+OIL,OWL	O	보통	×	×
OntoEdit	O	Tree & Graph	.	O	.	×	×
WebOnto	O	Graph	.	O	.	×	×
OWL Editor (씨컴테크)	O	Text	OWL	×	빠름	×	×
OWL 그래픽 편집기 (케이테크)	O	Graph	.	O	.	×	×
본 시스템	O	Tree & Form	OWL	O	아주 빠름	O	O

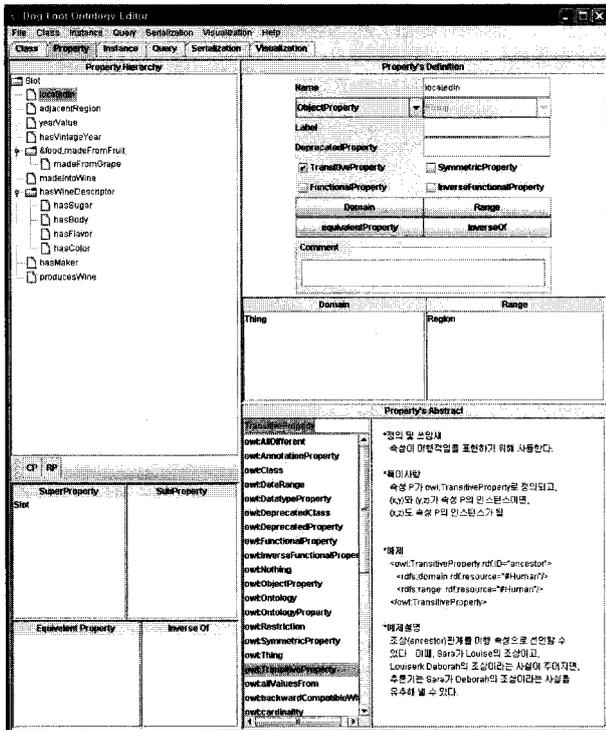


그림 5. RDF/OWL 어휘 라이브러리
Fig. 5. Vocabulary library of RDF/OWL.

하고자 하는 RDF/OWL 어휘의 국내 사용자지침 표준안을 작성하고 본 시스템에 접목시켜 개발자가 즉시 어휘를 검색할 수 있도록 하였다. 이를 통해, 어휘의 구체적인 정의와 어휘의 사용에 대한 예제를 통해 온톨로지 구축시 정확한 어휘사용을 가능하게 한다. 그림 5는 본 시스템에 적용한 RDF/OWL 어휘 라이브러리 적용한 화면이다.

IV. 비교평가

본 시스템의 주요 목적은 온톨로지의 구축 및 관리이

다. 개발자가 온톨로지를 다루기 쉬운 사용자 인터페이스 방식인지, 편집은 어떤 방식으로 이루어지는지, 플랫폼(platform)에 독립적이어서 컴퓨터의 운영 환경에 지장을 받지 않는지, 개발자에게 OWL 온톨로지에 관련한 직접적인 도움을 줄 수 있는지(어휘 도움말) 등의 항목으로 평가하였다. 그리고 파일 접근 방식이 범용적인 온톨로지 개발에 도움이 되는지(파일 형태)와 파싱의 속도 등도 항목으로 추가하였다. 표 5는 널리 사용되는 국외 온톨로지 저작 도구(5)와 국내에서 개발된 온톨로지 저작 도구(2)를 본 시스템과 비교한 내용이다.

모든 시스템은 사용자 중심으로 설계가 되어 있으나 인터페이스 구성 방식은 모두 다르다. 편집방식은 크게 그래프방식과 트리와 폼을 이용한 방식으로 나뉘어지는데, 그래프 방식은 클래스에 설정된 관계와 클래스로부터 생성된 인스턴스들을 그래프로 보여주어 온톨로지 전체를 쉽게 파악할 수 있게 시각화한 것이다. 하지만 온톨로지가 비대할 경우 수많은 클래스들과 클래스 사이에 설정된 복잡한 관계들로 인해 더욱 혼잡하게 되고, 시스템 자체의 리소스(메모리)를 많이 소모하여 시스템에 과부하가 걸리기 쉽다. 그러한 이유로 자원에 직접 접근이 용이하고 데이터 처리가 쉬운 트리와 폼 형식을 많이 사용한다. 그리고 온톨로지를 다룰 때 프로젝트 파일을 이용하는 경우에는 다른 도구를 이용하여 작성된 온톨로지를 다루기 힘든 단점이 있기 때문에, 본 도구에서는 OWL 파일을 직접 사용하여 OWL 언어로 작성된 온톨로지는 모두 다룰 수 있는 범용성을 높였다. 컴퓨터의 운영환경에 영향을 받지 않게 하였으며, 파싱 속도 또한 다른 도구들 보다 빠른 결과(wine.owl 파싱 : Protege 56초, 씨컴테크의 OWL Editor 4초, 본 시스템 2초)를 보였다. 특히, 본 시스템에서는 기존의 도구들이 제공하지 않는 OWL 어휘 라

이브리리를 제공함으로써 사용자가 온톨로지 구축을 쉽게 할 수 있도록 하였다. 또한 워드넷에 기반하여 특정 도메인에 대한 온톨로지를 자동으로 구축이 가능하게 함으로써, 온톨로지를 구축하는데 소요되는 노력과 시간을 줄일 수 있게 하였다.

V. 결 론

본 OWL 온톨로지 저작 도구는 온톨로지 전문가뿐만 아니라 온톨로지 구축을 희망하는 일반 사용자들까지 고려하여 사용하기 편리하게 설계하고 개발한 도구이다. 인간과 컴퓨터의 정확한 커뮤니케이션을 위한 시맨틱 웹에서 객체들 사이의 관계를 의미적으로 연결하는 온톨로지는 필수이다. 이러한 시맨틱 웹 실현을 앞당기기 위해서는 전문가들을 비롯한 일반 사용자 누구나 쉽게 온톨로지를 구축하고, 관리하는 것이 필요하기 때문에 본 온톨로지 저작 도구를 개발하게 되었다. 이 도구가 시맨틱 웹 실현뿐만 아니라 온톨로지를 이용하는 모든 분야의 빠른 실현에 도움이 될 것으로 기대된다.

참 고 문 헌

- [1] <http://jena.sourceforge.net/>
- [2] <http://sofa.projects.semwebcentral.org/index.html>
- [3] <http://www.knowledgetech.co.kr>
- [4] 김민수, 송세현, 서재우, 김민구, “클래스 시각화를 통한 OWL 온톨로지 편집기 개발”, 정보과학회 발표지, VOL.31 NO.02 pp.109~111, 2004. 10.
- [5] <http://www.w3.org/TR/owl-ref/>, OWL Web Ontology Language Reference
- [6] <http://www.w3.org/TR/rdf-schema/>, RDF Vocabulary Description Language 1.0:RDF Schema
- [7] OntoWeb, Deliverable 1.3: A survey on ontology tools, IST-2000-29243, 2002. 5
- [8] York Sure, Jurgen Angele, Steffen Staab, “OntoEdit: Guiding Ontology Development by Methodology and Inferencing”, LNCS Vol. 2519, Pages: 1205-1222, 2002.
- [9] <http://kaon.semanticweb.org/>
- [10] <http://protege.stanford.edu/index.html>
- [11] Robert Stevens, Chris Wroe, “Building Ontologies in DAML + OIL”, Comparative and Functional Genomics, Volume 4(2003), Pages 133-141
- [12] <http://www.ontoknowledge.org/tools/ontoedit.shtml>
- [13] KIPONTO 지식정보처리와 온톨로지 워크숍
- [14] http://www.knowledgetech.co.kr/product/product_3.htm

저 자 소 개



김 원 필(정회원)
1994년 호남대학교 전산통계학과 학사 졸업.
1999년 호남대학교 컴퓨터공학과 석사 졸업.
2004년 조선대학교 전자계산학과 박사 졸업.

2008년~현재 조선이공대학 U-사이버보안과 교수

<주관심분야 : 유비쿼터스, 정보검색, 정보보안>



김 정 호(정회원)
1976년 조선대학교 전자공학과 학사 졸업.
1983년 조선대학교 전자공학과 석사 졸업.
2001년 조선대학교 전자공학과 박사 졸업.

1980년~현재 조선이공대학 U-사이버보안과 교수

<주관심분야 : 유비쿼터스, 정보검색, 정보보안>