

# 시맨틱 웹에서 온톨로지를 위한 OWL 저작도구

안병규\*, 이무훈\*, 이민희\*, 김동혁\*, 백주현\*\*,  
고병오\*\*\*, 조현규\*\*\*\*, 송병렬\*\*\*\*, 최의인\*

## OWL Authoring Tools for Ontology in Semantic Web

Byung-Gyu Ahn, Moo-Hoon Lee, Min-Hee Lee,  
Dong-Hyuk Kim, Joo-Hyun Baek, Byung-Oh Goh,  
Hyun-Kyu Cho, Byoung-Youl Song, Eui-In Choi

### 요 약

현재의 웹 검색은 단순히 키워드 매칭만을 수행하기 때문에 검색 결과가 사용자가 원하는 정보와는 의미적으로 상이한 결과들을 많이 포함하고 있다. 사용자가 원하는 정보와 의미적으로 정확히 일치하는 정보들을 추출하기 위해서는 웹 자원에 대한 정확한 의미 부여와 함께, 이를 파악하고 선택적으로 획득, 통합, 가공할 수 있는 온톨로지와 같은 시맨틱 웹 환경 요소들이 필요하다. 웹 기술 표준화 단체인 W3C에서는 이와 같은 웹 자원에 대한 의미표현 기술로 OWL(Web Ontology Language)이라는 웹 온톨로지 표현 언어를 발표하였으나 아직 이를 효과적으로 저작, 편집할 수 있는 도구는 전무한 실정이다. 따라서 본 논문은 OWL을 기반으로 하여 온톨로지의 생성 및 편집을 효과적으로 제공할 수 있는 통합된 저작도구를 설계 및 구현하였다.

**Key Word** : Semantic Web, Ontology, OWL

---

본 연구는 한국전자통신연구원의 "시맨틱 비즈니스 문서 편집기 개발"의 연구 결과임  
\* 한남대학교 컴퓨터공학과  
\*\* 국방과학연구원  
\*\*\* 공주교육대학교 컴퓨터교육과  
\*\*\*\* 한국전자통신연구원

## 1. 서론

웹상에 존재하는 정보의 양은 예측할 수 없을 정도로 거대하며, 웹의 확산으로 인하여 이러한 정보의 양은 기하급수적으로 늘어나고 있다. 정보의 양이 증가하고 웹 이용자의 수준과 요구사항이 다양하고 복잡해짐에 따라 정확하고 적절한 정보를 제공하는 것이 중요한 문제가 되고 있다. 따라서 필요한 정보만을 효율적으로 추출하기 위해서 웹 정보들 간의 관계와 정보 자원들 사이의 의미적 연관성에 대한 메타정보와 컴퓨터가 이해할 수 있는 지식 표현 수단이 가능한 “시맨틱 웹”의 개념이 대두되었다 [1].

시맨틱 웹은 컴퓨터 스스로가 웹에 연결된 정보의 의미를 인식하고 사용자가 필요로 하는 정보를 검색하며 검색된 정보에서 지식을 유추할 수 있는 기능을 제공한다. 즉, 시맨틱 웹은 사람이 웹 정보에 대하여 의미를 파악하고 의미에 따라 필요한 정보를 선택하듯이 컴퓨터가 웹 정보의 의미를 이해하고 의미에 따라 선택적으로 정보를 획득, 통합, 가공할 수 있는 웹 환경을 제공하는 것이다. 시맨틱 웹에서 이러한 기능 및 수단을 제공할 수 있도록 하기 위한 기술 요소들로는 메타데이터, XML, RDF, 온톨로지 등이 있다[10, 11].

온톨로지는 초기에 지식의 공유와 재사용을 목적으로 인공지능 분야에서 연구되고 개발되었으나 최근 시맨틱 웹의 개념이 등장하면서 시맨틱 웹의 핵심 요소로 부각되고 있다. 컴퓨터가 XML 문서에 포함된 태그의 의미를 정확하게 인식하거나 에이전트 또는 응용 프로그램에 의해 지식을 처리하기 위해서는 지식을 표현하고 절차적 추론

과정을 수행할 수 있는 기반인 온톨로지가 필요하다. 이러한 필요에 따라 온톨로지 언어와 이를 활용하기 위한 온톨로지 저작 및 관리 도구에 관한 연구가 활발히 이루어지고 있다. 대표적인 온톨로지 언어로는 XOL, SHOE, OML, DAML+OIL 등이 있으며, 이를 활용하기 위한 저작 도구로는 OntoEdit, OilEd, Protégé-2000 등이 있다 [3, 6, 7, 9]. 하지만 이러한 온톨로지 저작 도구는 각기 다른 온톨로지 언어를 기반으로 하여 각기 다른 방식으로 개발되어 왔기 때문에 온톨로지를 저작하고 탐색함에 있어서 연동할 수 없고 전체적인 온톨로지 자원의 분석이 비효율적으로 처리된다는 문제점이 있다.

이러한 이유로 W3C를 중심으로 온톨로지 언어에 대한 표준화 작업을 수행하고 있다. 그 결과로 새로운 온톨로지 언어인 OWL을 발표되었으며, 현재 최종 권고안 상태이다. 본 논문에서는 W3C에서 표준으로 개발중인 온톨로지 언어인 OWL을 기반으로 온톨로지를 저작 및 편집할 수 있는 저작 도구를 설계하고 구현한 결과를 제시한다[2, 4, 5, 8].

## 2. 관련연구

### 2.1 Semantic Web

시맨틱 웹은 기존의 웹과 완전히 구별되는 새로운 개념이 아니라 현재의 웹을 확장하여 웹의 정보에 잘 정의된 의미를 부여하고 이를 통해 컴퓨터와 사람이 협동적으로 작업을 수행할 수 있도록 하는 패러다임이다.

시맨틱 웹에 대한 연구는 현재 크게 언어(Language), 기반구조(Infrastructure), 온톨로지(Ontology), 휴먼 인터페이스(Human interface) 등의 세부 주제로 나누어 이루어진다. 특히, 시맨틱 웹 언어는 온톨로지 언어와 같은 의미로서 시맨틱 웹의 내용을 표현하는데 반드시 필요하다. 잘 정의된 언어가 존재해야 시맨틱 웹의 주요 이슈인 상호운용성이 성취될 수 있으므로 언어에 대한 연구 결과는 시맨틱 웹의 다른 연구 분야에 대해서도 많은 영향을 미친다. 이미 RDF, RDF Schema, DAML+OIL, OWL 등의 시맨틱 웹 언어에 대한 표준들이 제안된 상태이다.

## 2. 2 RDF(S)

기존의 웹은 HTML로 이루어져 있어 데이터에 대한 표현력이 부족하기 때문에, 이에 대한 보완책으로 XML이 제안되었다. XML은 문서 구조에 자유로운 표현을 사용할 수 있는 보다 향상된 마크업 언어라고 할 수 있다. 하지만 서로 다른 메타데이터 도메인 간의 의미적인 매핑을 수행하는데 있어서 XML 스키마만으로는 메타데이터의 요소와 다른 요소들과의 관계를 기계가독형으로 표현할 수 없다. 이러한 한계를 극복하기 위하여 W3C 주도로 웹 자원에 대한 기술을 위한 RDF가 제정되었다. 하지만 자원의 속성과 자원들 간의 관계를 기술하기 위해 설계되었기 때문에 속성과 클래스의 명확한 정의, 클래스와 클래스 간의 관계, 속성과 속성간의 관계를 기술하기 위한 방법을 제공하지 않는다. 이를 보완하기 위해서 제정한 RDF 스키마는 메타데이터의 속성에 관한 정의, 속성에 대한 제어방법, 클

래스간의 관계 등을 사람과 기계가 동일하게 이해하고 처리할 수 있는 형태를 표현하는 것이 가능하도록 설계되었다.

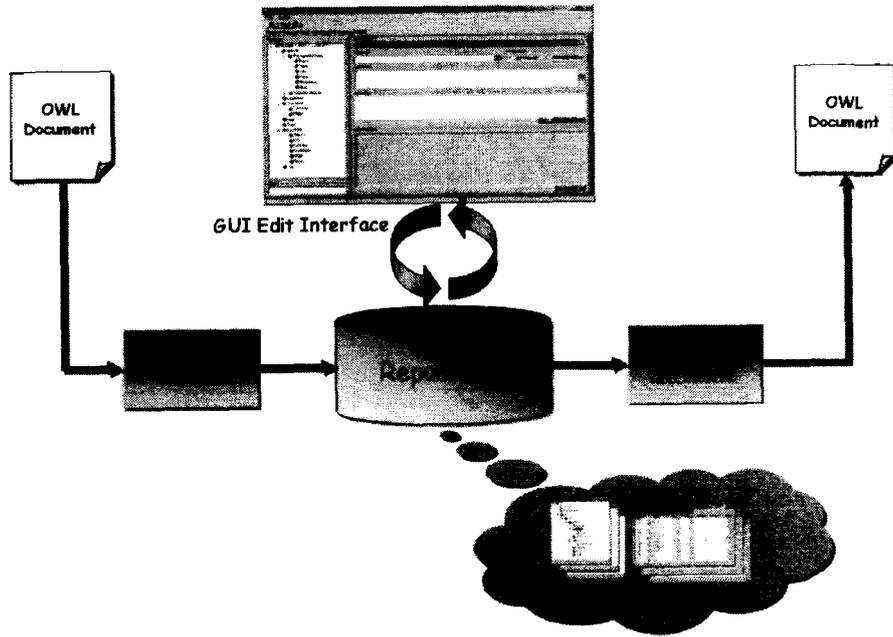
## 2.3 OWL

W3C의 주도로 웹 온톨로지 언어가 개발되기 시작한 이유는 RDF와 RDF 스키마가 표현할 수 없었던 동의관계, 역관계, Union, Intersection 등의 주요 관계를 표현할 수 있도록 모델링 요소를 확장, 강화할 필요가 있었기 때문이다. 그 결과 나타난 것이 DAML+OIL이다. OWL은 이 DAML+OIL에 기반을 둔 온톨로지 구축 경험을 토대로 개념의 일관성을 확보하여 클래스와 속성의 개념 및 그들 사이의 관계가 보다 명료하게 정의되도록 정리한 온톨로지 언어이다. 즉, OWL은 시맨틱 웹 형성의 가저인 온톨로지 개발에 필요한 언어 구조를 재정립한 결과이다.

OWL은 웹 온톨로지와 연관된 지식창고(knowledge bases), 즉 추론시스템에 축적된 논리적 명제(assertions)들을 정의하는 언어로서, 이러한 명제는 하나의 온톨로지에 근거하거나 또는 OWL이 명시하는 방식에 따라 수집한 다수의 분산형 온톨로지에 근거할 수 있다. OWL 명제는 클래스의 구성원들에 관한 사실과 구성원들 간의 관계를 기술하며, 이러한 명제의 집합으로 이루어진 온톨로지는 구문적으로 정의되지 않은 사실의 논리적 유추를 가능하게 한다.

## 3. OWL 저작도구

### 3.1 시스템 구조



<그림 1> 전체 시스템 구조

<그림 1>은 본 연구에서 구현한 OWL 저작도구의 전체 시스템 구조로서, OWL Parser, Repository, GUI Edit Interface, OWL Serializer로 구성되어 있다.

### 3.2 OWL Parser

OWL Parser는 사용자가 문서를 Load하거나 Import할 때, OWL 문서에서 요소들을 추출하여 Repository 객체에서 사용하는 내부 데이터 모델로 변환한다. Parsing은 두 단계로 수행되는데 요소들 간의 상호 참조가 가능하기 때문에, 먼저 pre-parsing 단계에서 고유한 URI를 갖는 Class, Property, Individual 들을 Repository 객체에 등록시킨다. post-parsing 단계에서는 모든 요소에 대해 실제 Contents를 Repository 내부에서 사용하는 데이터 모델로 변환하여 Repository에 반영시킨다.

### 3.3 Repository

Repository는 내부 데이터 모델로 변환된 OWL 요소들에 대한 정보를 통합 관리하는 객체로 Class, Property, Individual, Axiom, Header 정보 등 문서에 관한 모든 내용을 관리 한다. 또한 GUI 인터페이스에서 반복적으로 사용되는 Class와 Property의 Tree-Model도 Repository에서 통합 관리된다. 내부 데이터 모델의 설계는 W3C에서 발표한 OWL 명세서를 분석을 통해 각각의 요소들을 추출하여 설계하였다. <그림 2>는 Property 표현을 위한 Java Class 소스이다.

Repository는 GUI Interface와 연동을 통해 사용자가 OWL 온톨로지를 편집, 수정할 수 있도록 한다. 또한, OWL Serializer에서 내부 모델을 OWL 문서로 변환할 때에도 이용된다.

```

public class Property
    implements Comparable, owlClass {
    public static final int DATA_PROPERTY = 1;
    public static final int OBJECT_PROPERTY = 2;
    public static final int DEPRECATED_PROPERTY = 3;
    public static final int IMPORTED_PROPERTY = 4;

    public Identifier ID;
    public owlDocument doc = null;

    public int type=OBJECT_PROPERTY;

    public String comment = new String();
    public Vector vLabel = new Vector();

    public Vector subPropertyOf = new Vector();
    public Vector inverseOf = new Vector();
    public Property equivalentProperty;
    public Vector domain = new Vector();
    public Vector range = new Vector();

    public boolean Functional;
    public boolean inverseFunctional;
    public boolean Symmetric;
    public boolean Transitive;

    public Vector vTreeNodees = new Vector();
    public TreeSet tsTreeChild = new TreeSet();
    public Vector vReferencedObject = new Vector();
    .....(중략)

```

<그림2> Property의 내부 데이터 모델

Property	<ul style="list-style-type: none"> <li>■ ID 및 Comment 정의</li> <li>■ Super Property, inverse Property 설정</li> <li>■ Domain 및 Range 설정             <ul style="list-style-type: none"> <li>- named Class</li> <li>- anonymous Class</li> <li>- oneOf</li> <li>- Complex Class</li> </ul> </li> <li>■ Property 특성 정의             <ul style="list-style-type: none"> <li>- Functional</li> <li>- inverseFunctional</li> <li>- Symmetric</li> <li>- Transitive</li> </ul> </li> </ul>
Individual	<ul style="list-style-type: none"> <li>■ ID 및 Comment 정의</li> <li>■ Class 선택</li> <li>■ Relation 정의</li> </ul>
Axiom	<ul style="list-style-type: none"> <li>■ DisjointWith(Class) 정의</li> <li>■ DisjointWith(Individual) 정의</li> </ul>
Header	<ul style="list-style-type: none"> <li>■ versionInfo 정의</li> <li>■ priorVersion 정의</li> <li>■ backCompatibleWith 정의</li> <li>■ incompatibleWith 정의</li> <li>■ import 정의</li> </ul>
Namespace	<ul style="list-style-type: none"> <li>■ Namespace 정의</li> </ul>

<그림 3> GUI 모듈별 기능

<그림 4>는 GUI Interface의 예로 Class 편집을 위한 화면구성을 보여주고 있다.



<그림 4> Class 편집 화면

### 3.4 GUI Interface

OWL은 온톨로지 저작을 위한 풍부한 표현력을 제공하고 있으며, OWL 저작도구에서는 OWL 명세서 분석을 통하여 이를 효과적으로 편집할 수 있도록 6개의 모듈로 나누어 설계 및 구현하였다. <그림 3>은 각 모듈별 기능을 나타낸다.

모듈	기능
Class	<ul style="list-style-type: none"> <li>● ID 및 Comment 정의</li> <li>● Namespace 선택</li> <li>● Super Class 설정             <ul style="list-style-type: none"> <li>- named Class</li> <li>- anonymous Class</li> <li>- oneOf</li> <li>- Complex Class                 <ul style="list-style-type: none"> <li>: unionOf, intersectionOf, complementOf</li> </ul> </li> </ul> </li> <li>● Equivalent Class 설정</li> <li>● Class Restriction 정의             <ul style="list-style-type: none"> <li>- Cardinality 설정                 <ul style="list-style-type: none"> <li>: cardinality, minCardinality, maxCardinality</li> </ul> </li> <li>- Property Restriction 설정                 <ul style="list-style-type: none"> <li>: allValuesFrom, someValuesFrom, hasValue</li> </ul> </li> </ul> </li> </ul>

### 3.5 OWL Serializer

Serializer는 Repository에서 관리되는 내부 데이터 모델을 OWL 문서로 변환하는 작업을 수행한다. 먼저 매핑 룰에 따라 내부 모델을 XML DOM 모델로 변환을 하고, DOM 모델을 다시 텍스트 형식의 OWL 문서로 변환을 한다.

#### 4. 결론

현재 국외에서 온톨로지를 처리하기 위해 많은 노력들이 진행 중에 있으며, 시맨틱 웹을 위한 많은 프로젝트와 제품을 개발 중이다. 하지만 W3C에서 온톨로지 표현언어로 제정하고 있는 OWL을 기반의 온톨로지를 효율적으로 작성하고 편집할 수 있는 저작도구는 아직까지 전무한 실정이다. 이에 본 연구에서는 OWL 기반의 온톨로지 문서를 시각적인 GUI 인터페이스를 통해서 효율적으로 생성 및 편집할 수 있는 저작 도구를 설계 및 구현하였다. 한편 여러 자원으로 분산 저장되어 있는 OWL 온톨로지들을 하나로 통합할 수 있는 Merge기능의 구현과, 다른 언어로 작성된 온톨로지를 OWL로 변환하여 통합할 수 있는 방안에 대한 연구가 필요하다.

#### 5. 참고문헌

[1] Berners-Lee, T., Hendler, J. and Lassila, O., "The Semantic Web", Scientific American, 2001.

[2] Deborah L. McGuinness and Frank van Harmelen, "OWL Web Ontology Language Overview," 2003, <http://www.w3.org/TR/owl-features/>.

[3] Frank van Harmelen, et al., "Reference description of the DAML+OIL ontology markup language," 2001, <http://www.daml.org/2001/03/reference>.

[4] Michael K. Smith, et al., "OWL Web Ontology Language Guide," 2003, <http://www.w3.org/TR/owl-guide/>.

[5] Mike Dean and Guus Schreiber, "OWL Web Ontology Language Reference," 2003, <http://www.w3.org/TR/owl-ref/>.

[6] OilEd, <http://oiled.man.ac.uk/>.

[7] OntoEdit, [http://www.ontoprise.de/home\\_en/](http://www.ontoprise.de/home_en/).

[8] Peter F. Patel-Schneider, et al., "OWL Web Ontology Language Semantics and Abstract Syntax," 2003, <http://www.w3.org/TR/owl-semantic/>.

[9] Protégé-2000, <http://protege.stanford.edu/index.html/>.

[10] W3C, Extensible Markup Language (XML), <http://www.w3c.org/XML/>.

[11] W3C, Resource Description Framework (RDF), <http://www.w3.org/RDF/>.

[12] W3C, RDF Vocabulary Description Language(RDFS), <http://www.w3.org/TR/rdf-schema/>.

- 저자약력

2002년           한남대학교 컴퓨터공학과(공학사)

2002년 - 현재   한남대학교 대학원 컴퓨터공학과 석사과정

- 공동저자

이무훈 : 한남대학교 대학원 컴퓨터공학과 석사과정

이민화 : 한남대학교 대학원 컴퓨터공학과 석사과정

김동혁 : 한남대학교 대학원 컴퓨터공학과 박사과정

백주현 : 국방과학연구소 연구원

고병오 : 공주교육대학교 컴퓨터교육과 조교수

최의인 : 한남대학교 컴퓨터공학과 부교수

조현규 : 한국전자통신연구원 팀장

송병렬 : 한국전자통신연구원 연구원

- 관심분야

XML Database, ebXML, Ontology, Semantic Web

- 연락처

주소 : 대전광역시 대덕구 오정동 한남대학교 컴퓨터공학과 데이터베이스 연구실

우편번호 : 306-791

전화 : 042)629-7864

e-mail : bgahn@dblab.hannam.ac.kr