

온톨로지 저작을 위한 OWL builder의 설계 및 구현

이무훈, 김동혁, 장창복, 조현규*, 송병열*, 최의인
한남대학교 컴퓨터공학과
한국전자통신연구원*

Design and Implementation of OWL Builder for Ontology Authoring

Moo-Hun Lee, Dong-Hyuk Kim, Chang-Bok Jang,
Hynu-Kyu Cho*, Byoung-Youl Song*, Eui-In Choi
Dept of Computer Engineering, Hannam University
Electronics and Telecommunications Research Institute*

요 약

웹의 확산과 더불어 웹 상에 존재하는 정보의 양은 예측할 수 없을 정도로 증가하였고, 웹 사용자의 이용수준과 요구사항도 매우 다양하고 복잡해졌다. 이러한 환경에서 사용자 요구에 맞는 적절한 정보를 제공하기 위해서는 정확하고 필요한 정보만을 효율적으로 추출하고, 웹 정보들 간의 관계와 정보 자원들 사이의 의미적 연관성을 기술하는 지식 표현 수단으로 온톨로지(Ontology)가 필요하게 되었다. 이를 위해 W3C는 다양한 온톨로지를 통합하여 시맨틱 웹에서 웹 정보들간의 의미적 표현을 기술할 수 있는 표준으로 OWL(Web Ontology Language)을 제안하였으나, 이를 활용하기 위한 저작도구가 전무한 실정이다. 따라서 본 논문은 W3C에서 표준으로 제안하고 있는 OWL의 속성들을 분석하여, OWL 구문을 직관적이고 효율적으로 생성, 저장할 수 있는 OWL builder를 설계 및 구현하였다.

1. 서론

시맨틱 웹(Semantic Web)은 웹 상의 정보에 잘 정의된 의미(Semantic)를 부여함으로써 사람뿐만 아니라 컴퓨터도 쉽게 문서의 의미를 해석할 수 있도록 하여 컴퓨터를 이용한 정보의 검색 및 해석, 통합 등의 업무를 자동화하기 위한 목적으로 제안되었다[1]. Tim Berners-Lee는 시맨틱 웹이 기존의 웹과 완전히 구별되는 새로운 웹의 개념이 아니라 현재 웹을 확장하여 웹에 올라오는 정보에 잘 정의된 의미를 부여하라고 이를 통해 컴퓨터와 사람이 협동적으로 작업을 수행할 수 있도록 하는 패러다임이라고 그 역할을 정의하였다[2, 3].

시맨틱 웹은 컴퓨터 스스로가 웹에 연결된 정보의 의미를 인식하고 사용자가 필요로 하는 정보를 검색하며 검색된 정보에서 지식을 유추할 수 있는 기능을 제공한다. 즉, 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. 이를 실현하기 위해서는 컴퓨터가

문서에 포함된 태그의 의미를 정확하게 인식하거나 에이전트 또는 관련 응용 프로그램에 의해 지식을 처리할 수 있어야 한다. 지식을 표현하고 절차적 추론과정을 수행할 수 있게 하기 위해서는 온톨로지에 관한 연구가 필요하다.

온톨로지는 용어 사이의 관계를 정의하고 있는 일종의 사전과 같은 것으로 어휘의 정의를 다른 어휘와의 논리적 관계뿐만 아니라 가장 기본적인 어휘부터 파악해 나가는 구조를 통해 나타낸다. 따라서 응용프로그램 사이에서 웹 기반 지식 처리, 공유, 재사용 하는 것을 가능하게 하는 역할과 함께 사람과 응용프로그램 시스템 사이에서 공통된 주제의 의사소통을 제공하여, 전자 상거래에서 구매자와 구입자간의 기계 기반의 의사소통을 가능하도록 만들어 준다. 이러한 온톨로지로는 XML 구문을 기반으로 만들어진 XOL(XML-based Ontology Exchange Language)과 OML(Ontology Markup Language), SHOE(Simple HTML Ontology Extensions)가 있으며 W3C에서 제정한 RDF(Resource Description Framework)와 RDFS(RDF Schema)를 기반으로 만들어진 OIL(Ontology Inference Layer), DAML+OIL(DARPA Agent Markup Language)이 있다[4, 5].

이와 같은 다양한 온톨로지 언어를 바탕으로 다양한 온톨로지 관련 브라우저, 편집기, 분석기, 추론 엔진 등의 소프트웨어에 대한 연구가 진행 중 이지만, 각기 다른 영역에서

본 연구는 한국전자통신 연구원의 "시맨틱 비즈니스 문서 편집기 개발"의 연구 결과임

서로 다른 온톨로지를 사용함으로써 상호운용성에 문제가 발생하게 되었다. 이러한 문제를 해결하고 웹을 비롯한 다른 분야에서도 온톨로지 문서를 처리하기 위해 W3C는 OWL을 온톨로지의 표준으로 제안하고 이를 널리 활용하기 위한 저작도구에 대한 개발도 진행중이다. 이러한 노력의 결과로 이미 OntoEdit, OilEd, Protégé-2000와 같은 저작도구가 개발되어 사용되고 있다. 하지만 이러한 저작도구들은 아직까지 온톨로지 저작의 필요사항(Requirement)을 완벽히 만족시키지는 못하고 있다.

따라서 본 논문에서는 표준화된 온톨로지 언어를 기반으로 하여 온톨로지를 효율적으로 생성, 저작 할 수 있는 OWL Builder를 설계 및 구현 하였다. 본 논문의 구성은 다음과 같다. 먼저, 2장에서는 OWL에 대한 개념 및 다른 온톨로지 저작도구에 대해서 기술하고, 3장에서는 효율적인 온톨로지 저작을 위한 OWL Builder를 제안한다. 그리고 4장의 결론에서는 향후 연구과제를 제시한다.

2. 관련연구

2.1 OWL

시맨틱 웹의 구현에 있어 RDF 바로 윗 계층에 필요한 것이 웹 문서에 포함된 용어의 의미를 형식적으로 기술할 수 있는 온톨로지 언어이다. 기계가 웹 문서를 대상으로 유용한 추론 기능을 수행하려면 RDF 스키마가 제공하는 기초적인 의미 표현력을 뛰어넘는 언어가 필요하다.

OWL은 문서에 포함된 정보를 어플리케이션을 이용하여 자동으로 처리하고자 할 때 활용하는 언어로써 임의의 어휘를 구성하는 용어(term)의 의미와 용어들 간의 관계를 명시적으로 표현할 수 있다. 이와 같이 용어와 용어들 간의 관계를 표현한 것을 온톨로지라 한다. OWL은 DAML+OIL로부터 파생된 언어로써 XML, RDF, RDFS 보다 더 많은 의미 표현 수단을 제공하므로, 웹 상에서 기계가 해석할 수 있는 콘텐츠를 작성하는데 있어 다른 온톨로지 언어보다 뛰어나다.

OWL은 표현력이 서로 다른 Lite, DL, Full의 하위 언어로 구성된다. OWL Lite는 클래스 분류 계층과 간단한 제약 사항 표현을 필요로 하는 곳에서 사용하기 위한 언어이다. OWL DL은 계산학적 완전성(Computational Completeness)과 결정가능성(Decidability)을 유지하면서 최대의 표현력을 활용하는 곳에 적합하다. OWL Full은 계산학적인 어떤 보장 없이 최대의 표현력과 RDF의 유연한 문법을 모두 활용하고자 할 때 사용되는 것이다. 올바른 표현의 범위 및 타당한 결론의 영역에 있어서 OWL DL은 OWL Lite의 확장이고 OWL Full은 OWL DL의 확장 형태로 사용된다[6].

2.2 온톨로지 저작도구

현재 개발된 온톨로지 저작도구로 가장 대표적인 것은 스탠포드 대학의 Protégé-2000, 맨체스터 대학의 OilEd, ontoprise® GmbH 사의 OntoEdit, AT&T의 DUET이 있다.

지금까지 온톨로지 구축에서 있어 가장 널리 활용되고 있는 저작도구는 Protégé-2000이다. Protégé-2000은 국립의료

도서관(National Library of Medicine), NSF(National Science Foundation), DARPA(Defense Advanced Research Projects Agency)의 후원을 받아 스탠포드 의과대학의 의료정보학과(Stanford University School of Medicine, Stanford Medical Informatics)에서 지식 기반의 구조를 작성하기 위한 도구로 15년간의 연구 기간을 거쳐 개발되었다. 최근에는 보다 광범위하게, 보다 많은 이용자가 사용할 수 있도록 응용프로그램의 범위를 넓히고 있으며 플러그인(Plug-In)을 통해 다양한 온톨로지의 저작을 지원하고 있다[7].

맨체스터 대학에서 개발된 OilEd는 DAML+OIL 기반으로 온톨로지를 작성하는 저작도구이다. OilEd는 OIL을 기반으로 하며 프리웨어 편집기로 지원하는 목표로 개발되었기 때문에 대규모 온톨로지 구축, 온톨로지 통합 및 마이그레이션(migration) 등의 전체적인 온톨로지 개발환경을 제공하지 않지만, 오픈소스 라이선스를 통해 사용자들이 소프트웨어를 자유롭게 공유함으로써 기술 보급은 물론 피드백을 통해 개발에 노력하고 있다[8].

OntoEdit은 독일의 ontoprise® GmbH 사에서 개발하였으며, 그래픽을 이용하여 온톨로지의 개발과 유지보수를 제공하는 온톨로지 공학 환경(Ontology Engineering Environment)을 구축하고 있다. OntoEdit은 강력한 내부 온톨로지 모델을 가지고 있으며 이 모델은 개념(concept), 관계(relation), 원칙(axioms)이 가능한 중립적 모델링 표현 언어를 제공하고 있다. 이 틀은 직관적인 인터페이스를 통해서 온톨로지에 대한 시각화 및 탐색을 지원한다[9].

DUET은 AT&T에서 개발되었고, DAML을 위한 UML(Unified Modeling Language) 시각화와 저작 환경을 제공한다. 핵심 DAML 개념은 DAML을 위해 UML 프로파일을 통하여 UML에 매핑시키며, 유효성 검증을 통해 유효한 UML 다이어그램을 DAML+OIL로 생성시켜준다[10].

3. OWL Builder

3.1 OWL Builder의 구조

본 논문에서 설계한 OWL Builder는 Parser, Internal Data Model, GUI Interface Visualizer, Serializer로 구성되어 있으며, 그림 1은 OWL Builder의 전체 시스템 구조를 보여주고 있다.

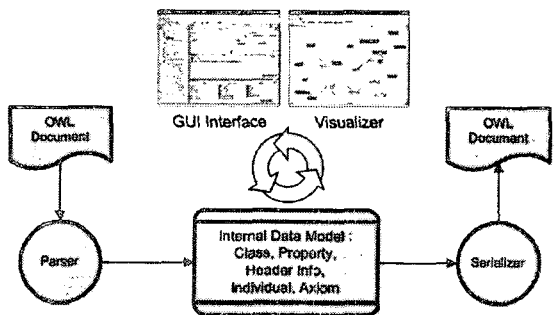


그림 1. OWL Builder의 시스템 구조

3.2 OWL 파서(parser)

OWL 파서는 OWL 문서를 입력 받아 OWL 문서에서 요소들을 추출하여 내부 데이터 모델의 객체로 변환한다. 파싱은 Pre-Parsing과 Post-Parsing의 두 단계를 거치게 되며, 먼저 pre-parsing 단계에서 Jena API를 통해 Jena Ontology 모델을 생성한다. post-parsing 단계에서 Jena Ontology Model을 참조하여 실제 툴에서 사용될 Internal Data Model로 변환한다. 변환 단계는 먼저 온톨로지의 헤더 정보를 확인하여 온톨로지서 사용된 Namespace를 내부 모델에 등록하고, named Class, Property, Individual에 대한 Dummy 객체를 생성한다. 생성된 Dummy 객체가 생성된 후에는 모든 객체에 대해 해당 객체가 갖고 있는 세부 의미 요소들을 추출하여 내부 모델에 반영하고 이들 간의 참조관계를 링크로 연결한다. 그림 2는 OWL 파서의 파싱 처리 루틴을 설명하고 있다.

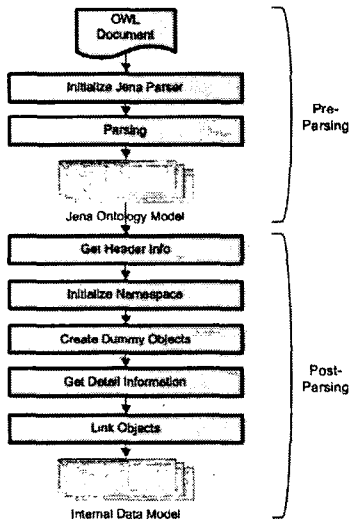


그림 2. 파싱 처리 루틴

3.3 내부 데이터 모델(Internal Data Model)

Parsing 단계에서 추출된 요소들은 내부 데이터 모델로 변환되어 메모리에 저장된다. 내부 데이터 모델의 설계는 OWL 스펙에서 제공하는 모든 의미 표현을 충실하게 제공할 수 있어야 한다. 또한 GUI 인터페이스를 통해 전달된 사용자의 요구에 따라 각각의 의미 요소들의 데이터를 쉽게 조작할 수 있도록 다양한 인터페이스가 제공되어야 한다. 본 논문에서는 OWL 스펙에서 요구하는 의미 표현을 충실하게 제공할 수 있도록 데이터 저장 및 처리를 위한 owlDocument, owlObject, owlClass, idClass, anonymousClass, oneOf, complexClass, Axiom, Property, Individual, Namespace dataValue, Identifier, Restriction, Relation의 15개의 클래스를 정의하였다. 특히, owlDocument는 내부 데이터 모델의 최상위 객체로 온톨로지의 모든 의미요소들을 포함하며, Class, Property, Individual등 자원 식별자를 통해 접근하는

객체에 대한 HashTable을 유지함으로써 보다 빠른 객체 참조가 가능하도록 하였다. 그림 3은 내부 데이터 모델의 클래스 다이어그램을 보여주고 있다.

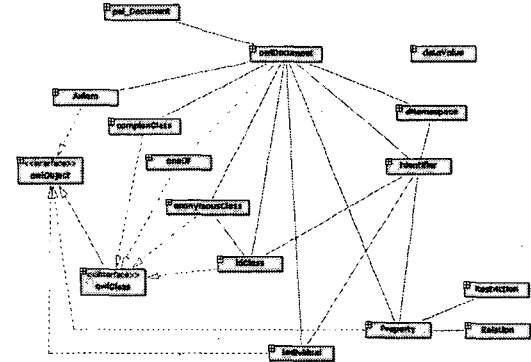


그림 3. 내부 데이터 모델의 클래스 다이어그램

3.4 사용자 인터페이스(User Interface)

OWL은 다른 온톨로지 언어에 비해 풍부한 표현력을 갖고 있기 때문에, 사용자 인터페이스는 이러한 다양한 표현을 충분히 수용할 수 있도록 구성하여야 한다. 본 연구에서는 6개의 큰 인터페이스를 각각의 탭으로 구성하였고, 추가적으로 정의에 필요한 부분은 다이얼로그를 통해 편집할 수 있도록 하였다. 표 1은 각 인터페이스별 기능을 나타낸다.

표 1. 인터페이스 모듈별 기능

Class	Property
<ul style="list-style-type: none"> Annotation 정의 <ul style="list-style-type: none"> - ID, Label, Comment subClassOf equivalentClass Restriction 정의 <ul style="list-style-type: none"> - Cardinality - Property Restriction Class Combination <ul style="list-style-type: none"> - intersectionOf - unionOf - complementOf 	<ul style="list-style-type: none"> Annotation 정의 <ul style="list-style-type: none"> - ID, Label, Comment subPropertyOf equivalentProperty inverseProperty Domain Range Property 특성 정의 <ul style="list-style-type: none"> - Functional - inverseFunctional - Symmetric - Transitive
<ul style="list-style-type: none"> Annotation 정의 <ul style="list-style-type: none"> - ID, Label, Comment Individual간 관계 <ul style="list-style-type: none"> - sameIndividualAs - differentFrom Relation 정의 	<ul style="list-style-type: none"> AllDifferent disjointWith
<ul style="list-style-type: none"> versionInfo priorVersion backwardCompatibleWith inCompatibleWith import 	<ul style="list-style-type: none"> Namespace 추가/삭제 Default Namespace 설정

3.5 직렬화 모듈(Serializer)

직렬화 모듈은 내부 데이터 모델의 각 요소들을 XML-DOM 모델로 변환하기 위한 메소드를 제공한다. 모든 요소들이 XML-DOM 모델로 변환되면 JDOM의 XMLOutputter 객체를

이용하여 Tagging을 수행하여 최종적으로 텍스트 형식의 OWL 문서로 변환된다. 그림 4는 직렬화 모듈의 처리 과정을 표현하고 있다.

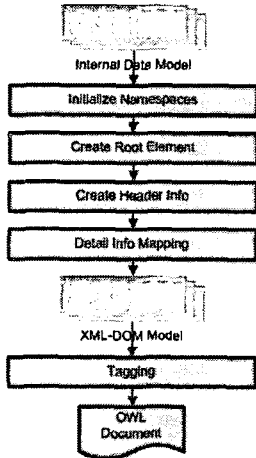


그림 4. 직렬화 모듈의 처리 과정

3.6 시각화 및 탐색 모듈(Visualizer)

시각화 및 탐색 모듈에서는 온톨로지의 의미 요소인 Class, Property, Individual를 각각의 노드(Node)로 표현하며, 이들 간의 관계는 아크(Arc)로 표현한다. 시각화 및 탐색 모듈의 구현은 Touch Graph의 Graph Layout[12]을 사용하였다. 내부 데이터 모델이 변경되면 연관된 노드와 아크도 동적으로 변경되며, Visualization 탭에서 수정된 데이터는 내부 데이터 모델에서도 수정된다. 탐색 과정에서는 선택된 노드와 인접한 노드를 중심으로 화면을 재구성하여 표시하여 인접 노드를 중심으로 관계를 추적하는 것이 가능하다. 그림 5는 시각화 탐색 모듈의 실행 화면을 보여준다.

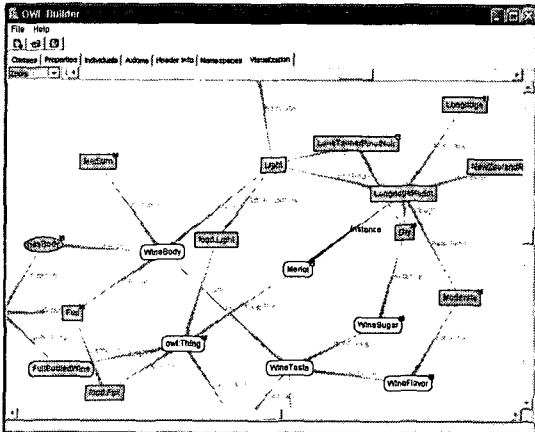


그림 5. 시각화 및 탐색 모듈의 실행 화면

5. 결론

시맨틱 웹이 추구하는 자원 의미의 표현력을 지원하기 위한 핵심 기반 구조로 온톨로지를 들 수 있으며, OWL은 이러한 기반 구조의 발전 단계 중 가장 최근에 부상한 웹 온톨로지 언어이다. OWL은 풍부한 표현력과 다양한 제약 조건을 제공하고 있으며, 광범위하게 사용되어질 수 있는 온톨로지의 표준이다. 하지만 현재 이러한 풍부한 표현력을 모두 지원해 줄 수 있는 OWL 기반의 온톨로지 저작도구는 개발되어 있지 않은 실정이며, 기존의 온톨로지 저작도구는 특정 영역에서만 사용되고 있어 포괄적으로 사용할 수 있는 온톨로지를 구축하기 위해 광범위한 활용이 불가능한 문제점을 가지고 있다. 따라서 본 논문에서는 OWL에서 지원하는 다양한 표현력을 충분히 수용하여 효율적으로 OWL을 생성, 저작할 수 있는 OWL Builder를 설계 및 구현하였다.

향후 연구과제로는 웹 상에 분산되어 있는 OWL 온톨로지들을 하나로 통합하는 방안과 다른 언어로 작성된 온톨로지들 OWL로 변환하여 통합 할 수 있는 방안에 대한 연구가 진행되어야 할 것이다.

[참고문헌]

- [1] 이재호, "시맨틱 웹의 온톨로지 언어", 한국정보과학회 정보과학회지, 제21권 제3호 pp.18-27, 2003
- [2] 김홍기, 김학래, 이강찬, 정지훈, 이재호 외, "월드와이드 웹에서 시맨틱 웹으로", 마이크로소프트웨어 시맨틱 웹 특집, pp. 242-301, 2002
- [3] Berners-Lee, t., et al., "The Semantic Web," Scientific American, 2001.
- [4] McGuinness, D., Fikes, R., Hendler, J. and Stein, L., "DAML+OIL : an ontology language for the Semantic Web," IEEE Intelligent Systems, Vol. 17, No. 5, pp.72-80, 2002.
- [5] Geflin, J., Hendler, J. and Luke, S., "SHOE : a knowledge representation language for Internet applications," tech. report CS-TR-4078. Dept. of Computer Science, Univ. of Maryland at College Park, 1999.
- [6] W3C, OWL Web Ontology Language Overview, <http://www.w3.org/TR/owl-features/>
- [7] Protégé-2000, <http://protege.semanticweb.org/>
- [8] OilEd, <http://oiled.man.ac.uk/>
- [9] OntoEdit, http://www.ontoprise.de/products/ontoedit_en
- [10] DUET, <http://grcinet.grci.com/maria/www/CodipSite/Tools/Tools.html>
- [11] Jena, <http://www.hpl.hp.com/semweb/jena2.htm>
- [12] TouchGraph, <http://www.touchgraph.com/>