

OWL Web Ontology Language 를 위한 Ontology Parser 의 설계

이미경, 박천수, 손주찬*

*한국 전자 통신 연구원 지능형 로봇 연구단

e-mail : lmk63398@etri.re.kr

A Design of Ontology Parser for OWL Web Ontology Language

Mi-Kyoung Lee Shu-Cheon Park Joo-Chan Sohn*

*Intelligent Robot Research Division, ETRI

요 약

시맨틱 웹에 이용되는 웹 온톨로지 언어들은 RDF/RDFS, DAML+OIL, OWL 등이 있으며, 현재 W3C 에서는 OWL 을 웹 온톨로지 표준 언어로 삼고 있다. 기존의 웹 온톨로지 문서들의 파서는 대부분 RDF 를 기반으로 한 Triple 모델을 기반으로 하여 파싱한다. 그러나 OWL 의 경우는 triple 형태로 변환시키면 OWL Full 의 형태를 가지게 되고 OWL 온톨로지의 표현력과 데이터의 손실을 가져오게 된다. 따라서 OWL 문서의 파싱을 위하여 우리는 OWL Abstract Syntax 를 이용하여 Tree 모델을 가지는 OWL 파서를 만들고자 한다. 본 논문에서는 시맨틱 웹에서 사용되는 웹 온톨로지들을 파싱하여 온톨로지 객체 모델을 생성해주는 기능을 가지는 온톨로지 파서를 설계, 구현하였다. 논문에서 설계한 온톨로지 파서는 RDF, DAML+OIL, OWL 웹 온톨로지 문서들을 파싱하여 온톨로지 객체 모델을 생성할 때, RDF 온톨로지의 경우는 Triple 모델 형태로 파싱을 하지만, OWL 온톨로지의 경우에는 OWL Abstract Syntax Tree 모델 형태로 파싱한 후, OOM 으로 변환시켜준다. 이를 위해 웹 온톨로지 언어의 종류 구분과 OWL 온톨로지의 경우, OWL Full, OWL DL, OWL Lite 의 서브 타입을 구별하는 기능도 추가하였다.

1. 서론

시맨틱 웹(Semantic Web)은 웹 상의 정보에 의미를 부여하여 지능화된 웹 환경을 제공해준다[1]. Tim Berners-Lee 는 시맨틱 웹이 기존의 웹과 완전히 구별되는 새로운 웹의 개념이 아니라 현재의 웹을 확장하여 웹 상의 정보에 잘 정의된 의미를 부여하고 이를 통해 컴퓨터와 사람이 협동적으로 작업을 수행할 수 있는 패러다임이라고 정의하였다[2].

시맨틱 웹에서의 “잘 정의된 의미”를 다루기 위해서 이용되는 것이 웹 온톨로지 언어이다. 의미를 부여하기 위한 수단으로 RDF(Resource Description Language)가 표준으로 자리 잡았고 RDF 로 표현될 수 없는 어휘들을 수용하기 위해 DAML+OIL, OWL(Web Ontology Language)라는 웹 온톨로지 언어가 생성되었다. 이에 따라 온톨로지를 표현할 수

있는 언어들에 대한 접근 방법, 처리 방법에 대한 연구가 필요하게 되었으며, 여러 가지 웹 온톨로지 언어로 표현된 문서들을 처리하기 위해서 온톨로지 파서가 필요하게 되었다. 기존의 온톨로지 파서들은 대부분 RDF 기반의 Triple 모델을 이용하여 온톨로지를 파싱한다. 하지만 Triple 모델을 이용하면 표현력이 풍부한 DAML+OIL, OWL 온톨로지를 Triple 모델로 변환해야하므로, 온톨로지 데이터의 손실을 가져오게 된다.

따라서, 본 논문에서는 OWL 온톨로지를 파싱할 때 Triple 모델을 이용하여 OOM 을 생성할 경우의 단점을 보완하기 위해 기존에 구현한 온톨로지 객체 모델 생성 시스템의 Triple-view 온톨로지 파서에 OWL Abstract Syntax Tree-view 를 가지는 모듈을 추가하여 설계한다.

2. 관련 연구

2.1 OWL (Web Ontology Language)

W3C 의 Semantic Web activity 인 OWL 은 DAML+OIL 를 기반으로 발전된 형태로 시맨틱 웹을 위한 웹 온톨로지 표준 언어이다. OWL 은 DAML+OIL 과 유사한 형식을 가지며, DAML+OIL 의 네임 스페이스와 속성 클래스 이름 등을 변경하고 RDF/RDFS 의 변화를 수용하였다[3]. OWL 은 DL(Description Logic)을 기반으로 만들어진 RDF 확장 언어이며, OWL 의 axioms 은 DAML+OIL 의 axioms 보다 더욱 풍부한 표현력을 가지고 있으며, 클래스나 속성 간의 subsumption 이나 equivalence 등의 다양한 성격을 선언하는데 사용된다. OWL 은 기본적으로 Description Logic 의 추론 능력과 표현력을 가지고 있기 때문에 DL 의 장점을 가지고 있으며, 현재 OWL 의 서브 타입 종류로는 OWL Lite, OWL DL, OWL Full 로 나눌 수 있다.

- OWL Lite : 분류 계층과 간단한 제약의 특성만을 기본적으로 필요로 하는 사용자를 위한 언어
- OWL DL : Reasoning System에서 모든 결과가 확실한(한정된 시간 내에 결과가 나오는) 최대한의 표현력을 필요로 하는 사용자를 위한 언어
- OWL Full : (결과를 보장할 수 없는) RDF의 syntactic freedom과 최대한의 표현력을 필요로 하는 사용자를 위한 언어

OWL Test Cases[4]에서 정의된 OWL Syntax Checker 는 OWL 온톨로지의 서브 타입을 OWL DL 과 Lite 로 구분하는 역할을 한다.

```

axiom ::= 'DatatypeProperty(' datavaluedPropertyID ['Deprecated'] ( annotation
  { 'super(' datavaluedPropertyID ')' } ['Functional']
  { 'domain(' classID ')' } { 'range(' dataRange ')' } )'
| 'ObjectProperty(' individualvaluedPropertyID ['Deprecated'] ( annotation
  { 'super(' individualvaluedPropertyID ')' }
  [ 'inverseOf(' individualvaluedPropertyID ')' ] [ 'Symmetric' ]
  [ 'Functional' | 'InverseFunctional' | 'Functional' 'InverseFunctional' | 'Transitive' ]
  { 'domain(' classID ')' } { 'range(' classID ')' } )'
| 'AnnotationProperty(' annotationPropertyID ( annotation )' )'
| 'OntologyProperty(' ontologyPropertyID ( annotation )' )'
dataRange ::= datatypeID | 'rdfs:Literal'
    
```

그림 1. OWL Abstract Syntax

그림 1 은 OWL Abstract Syntax 의 일부이다[5]. OWL Abstract Syntax 는 OWL 의 문법 규칙을 이해하기 쉽도록 BNF 를 이용하여 기술하였고, OWL Abstract Syntax 를 Tree 형태로 구현하면 XML 의 DOM 과 같이 Tree 형태의 모델링이 가능해진다.

2.2 Jena2

웹 온톨로지 파서 중 가장 널리 알려진 HP 의 Jena 는 시맨틱 웹 애플리케이션을 위한 자바 프레임워크이다. Jena 파서는 ARP(Another RDF Parser)를 이용하여 RDF 그래프 형식으로 데이터를 읽어들이고 Jena 데이터 모델로 변환하여 처리된다[6]. Jena 는 기존의 RDF 파싱 뿐만 아니라 DAML+OIL, OWL, N3, DB 등의 데이터를 파싱 해주고 RDF/XML, RDF/XML abbreviation, N-Triple 형태의 출력 형태를 지원해준다. Jena 에서 지원하는 DAML+OIL, OWL 온톨로지의 파싱은 RDF 파

서(ARP)를 확장한 구조로 되어 있어서 RDF-centric view 를 가지고, RDF 모델 위에 DAML+OIL 이나 OWL 전용의 layer 를 가진다[6,7].

Jena 는 RDF 데이터 모델을 이용하여 OWL 온톨로지를 파싱하기 때문에 ARP 에서 생성한 Triple 모델과 OWL abstract syntax 가 1:1 로 매핑되지 않으므로, 손실되는 데이터가 생기며 Triple 모델은 OWL Full 의 형태를 가지게 되므로, OWL Lite, OWL DL 의 타입 구분이 모호해지게 된다.

3. 온톨로지 객체 모델 생성 시스템

3.1 시스템의 구조

그림 2 는 온톨로지를 생성, 접근, 처리 할 수 있는 온톨로지 객체 모델의 구조이다. 본 시스템은 웹 온톨로지에 접근하여 분석하고, 메모리상에 모델을 생성한다. 그리고 외부 애플리케이션에서는 모델 접근 API 를 통하여 모델에 대한 정보를 교환하고, 메모리 상에 생성된 모델에 대한 검증과 serialize 를 하는 기능을 제공하는 모듈들로 구성되어 있다[8].

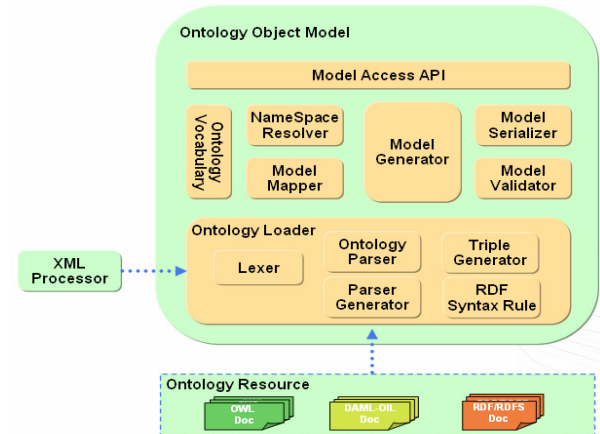


그림 2. 온톨로지 객체 모델 생성 시스템의 구조

Ontology Loader 모듈에 들어가는 온톨로지 파서는 웹 온톨로지를 읽어들이어서 Statement 를 구성하게 된다. XML Process 를 통해 처리된 정보를 이용하여 Lexer 에서 어휘별 토큰을 생성한다. Lexer 모듈에서 생성된 토큰을 이용하여 Ontology Parser 모듈에서는 RDF 문법 규칙에 따라서 토큰의 구문을 분석하고 Triple 형태의 Statement 를 생성한다. 이때, 온톨로지 파서를 생성하기 위해서 Parser Generator 인 ANTLR 를 이용한다[9].

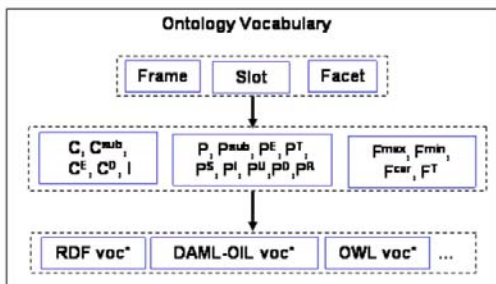
Ontology Loader 에서 생성된 Triple 모델은 Model Generator 를 통해 전달된다. 이때 RDF 문법에 따라 syntax 를 체크하고, triple 를 생성하며, triple 객체를 모델로 넘겨주면서 RDF API 를 제공하게 된다[9].

온톨로지 언어에 따른 Namespace 를 처리하기 위해 Namespace Resolver 를 거쳐서 메모리 상에 모델을 생성하게 된다. Model Generator 는 모델을 생성하기 위해 Ontology Vocabulary 에서 이미 정의된 어휘를 사용하여 도메인 온톨로지 모델을 생성하게 된다. Model Mapper 은 RDF 기반의 온톨로지 언어 외에 또 다른 형태의

지식 표현 언어를 위한 매핑에 사용되는 모듈로 Onotology Vocabulary 에서 정의되지 않은 기존의 어휘에서 확장하거나 구현하여 어휘를 정의할 수 있도록 한다. 생성된 모델은 Model Validator 을 통해 모델이 유효한 모델인지를 검증할 수 있는 모듈이다. 이렇게 검증이 끝나면 Model Serializer 을 통해 모델링한 메모리상의 모델을 원하는 온톨로지 언어로 생성할 수 있다. 온톨로지를 새로 구축하거나 기존의 온톨로지를 로드하여 수정할 경우 Model Access API 를 통해 외부 애플리케이션과의 인터페이스를 제공한다[8].

3.2 온톨로지 객체 모델(OOM)

온톨로지 객체 모델은 프레임 기반(frame-base)의 최상위 온톨로지 계층, 공통된 공리(axioms)와 어휘를 표현하는 공통 온톨로지 계층, 온톨로지 언어에 의존적인 기능 온톨로지 계층으로 구성된다. 이러한 계층 구조를 제공함으로써 각각의 온톨로지 언어에서 표현하고자 하는 어휘의 중복성을 줄이고, 새로운 온톨로지 언어나 기존의 지식베이스 시스템에서 표현된 정보를 상위 온톨로지 계층과 공통 온톨로지 계층을 이용하여 새로운 어휘를 정의 함으로써 확장성을 제공할 수 있을 뿐만 아니라 재사용성을 높일 수 있다[8]. 그림 3 은 Ontology Vocabulary 의 온톨로지 언어에서 제공될 수 있는 어휘들에 대한 계층도를 나타낸다.



voc^{*} : vocabulary
 C : class, C^{sub} : subClassOf, C^E : equivalentClass, C^D : disjointWith, I : instance
 P : property, P^{sub} : subPropertyOf, P^E : equivalentProperty, P^T : transitive,
 P^S : symmetric, P^I : inverseOf, P^U : unique, P^D : domain, P^R : range
 F^{max} : cardinality, F^{min} : minCardinality, F^{car} : maxCardinality, F^T : type

그림 3. 온톨로지 객체 모델의 계층도

4. OWL 온톨로지 파서의 설계

4.1 시스템의 구조

3 장에서 설명한 온톨로지 객체 모델 생성 시스템 [8]은 웹 온톨로지를 Triple 모델로 변환 후, 그림 3 의 온톨로지 객체 모델을 생성해준다. 따라서 기존의 온톨로지 파서 시스템들의 단점인 Triple 을 이용한 모델링을 하므로 OWL constructor 의 풍부한 표현력을 유지할 수 없다. 또한, 그림 3 의 온톨로지 객체 모델은 Ontology Vocabulary 를 기본으로 설계되었기 때문에 OWL 온톨로지의 경우에는 OWL Abstract Syntax Tree 형태로 모델링 하는 것이 데이터와 표현력을 유지하는 장점을 가지게 된다.

OWL 온톨로지의 경우, OWL 의 문법에 유효한지를 검사 해야하며, OWL DL, Lite 의 서브 타입을 체크하기 위해 한번 더 Validator 를 거쳐야 한다. OWL 온톨로지의 경우, Triple 모델로 변환하여 온톨로지 객체 모델으로 모델링하게 되면 OWL Vocabulary 를 중심으로 설계된 온톨로지 객체 모델에 1:1 형태로 매핑되기 어렵다.

따라서 이러한 OWL 전용 파싱 기능을 제공하기 위해서 우리가 구현한 온톨로지 객체 모델 시스템의 온톨로지 파서 모듈을 그림 4 와 같이 변경하여 설계한다. 진하게 표시된 모듈이 OWL 온톨로지의 파싱을 위해 새롭게 추가된 부분이다.

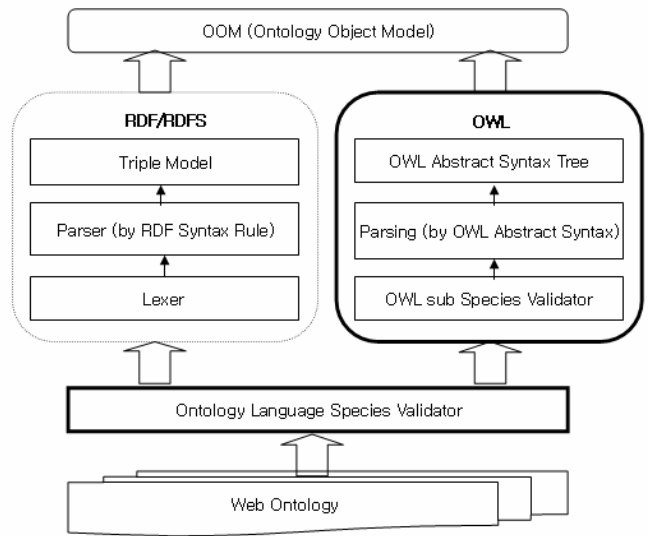


그림 4. Ontology Parser 의 구조

웹 온톨로지를 온톨로지 객체 모델로 변환하기 위해서는 온톨로지 파서를 거치게 된다. 파싱을 하기 전에 온톨로지 타입을 체크하여 RDF 기반의 Triple 형태와 OWL 기반의 OWL Abstract Syntax Tree 모델 형태로 파싱을 할 것인지를 선택한다. 만약 온톨로지가 OWL 로 구축된 것이라면, OWL 온톨로지의 서브 타입(OWL DL, Lite)을 체크하여 파싱 규칙과 Tree 모델을 선택하게 된다. 그리고 만들어진 모델을 이용하여 온톨로지 객체 모델을 생성하게 된다. OWL 문서의 Triple 모델이 필요할 경우에는 OWL Abstract Syntax Tree 로 만들어진 모델을 Triple 모델로 변환해주기 위해서는 변환 규칙을 담은 매핑 테이블을 이용할 수 있다[10].

4.2. 시스템의 모듈 구조

온톨로지 파서 부분에서 추가될 OWL 용의 파서 모듈의 구조를 살펴본다.

(1) Ontology Language Species Validator

웹 온톨로지의 타입을 체크한다. 온톨로지 언어가 RDF, DAML+OIL, OWL 온톨로지인지를 구분하는 역할을 하여 RDF, DAML+OIL 온톨로지의 경우는 Triple model 로 변환하여 OOM 을 생성하고, OWL 온톨로지

인 경우에는 OWL Abstract Syntax Tree Model 로 변환하여 OOM 으로 생성한다.

온톨로지의 타입을 체크하기 위해서, 이 모듈에서는 온톨로지 문서의 네임스페이스를 이용하여 타입을 구분한다.

(2) OWL Sub Species Validator

만약 온톨로지가 OWL 로 구축된 경우, OWL 의 서브 타입이 OWL DL, OWL Lite 에 따라서 OWL Abstract Syntax 가 달라진다. 따라서 OWL sub species validator 모듈을 거쳐서 OWL 의 서브 타입을 체크하게 된다. OWL DL 과 Lite 는 서로 다른 Abstract Syntax 를 가지기 때문에 Paring 모듈 부분에서 다른 문법 규칙을 적용하여 Tree 모델을 생성한다. 이 모듈에서는 OWL Lite 가 OWL DL 에 비해 사용되는 vocabulary 가 제한적이고 restriction 의 차이점을 이용하여 구분한다.

(3) Parsing

OWL 온톨로지의 Syntax 가 Validation 인지를 검사하고 OWL Abstract Syntax Tree 모델로 나타내기 위하여 파싱하는 역할을 하는 모듈이다. 만약 OWL 온톨로지의 서브 타입이 Lite 일 경우는 OWL Lite Axioms 의 규칙에 따라 유효성을 검사하고 OWL Lite 용 Abstract Syntax Tree 모델을 생성하게 된다.

(4) OWL Abstract Syntax Tree

OWL 온톨로지를 OWL Abstract Syntax 를 이용하여 Tree 모델 형태로 생성하는 모듈이다. OWL Abstract Syntax Model 은 온톨로지 객체 모델의 구조와 유사하여 모델 생성시 손실되는 데이터가 발생하지 않는다.

5. 결론 및 향후 과제

시맨틱 웹을 위해서는 웹 온톨로지가 필요하고, 온톨로지를 구축하기 위해서 웹 온톨로지 언어가 사용되고 있다. 웹 온톨로지 언어로 구축된 시맨틱 웹 애플리케이션을 위한 프레임 워크로 웹 온톨로지 문서들을 파싱하고 온톨로지 객체 모델을 생성하는 시스템을 설계, 구현하였다. 기존에 구현한 시스템[8,9]은 중립적인 온톨로지 객체 모델을 가지지만, DAML+OIL, OWL, RDF/RDFS 의 모든 온톨로지 언어가 RDF 기반의 Triple 형태로 파싱된 후, 온톨로지 모델을 생성하게 된다. 그러나 OWL 온톨로지인 경우, 표현력이 풍부한 OWL 의 특성을 살리기 위해서는 Triple 모델로의 변환보다는 데이터와 표현력의 손실이 없는 OWL Abstract Syntax 를 이용한 Tree 모델로 변환하는 OWL 파싱 모듈을 설계하였다. 본 논문에서 설계한 OWL 온톨로지 파서 기능을 추가하면, 우리가 구현한 온톨로지 객체 모델 생성 시스템은 온톨로지 구축 언어에 상관없이 OWL, RDF/RDFS, DAML+OIL 의 모든 온톨로지 언어의 파싱을 지원하고 공통의 온톨로지 객체 모델을 생성해준다. 본 논문에서 구현한 온톨로지 객체 모델은 웹 온톨로지 문서들의 공통된 모델을 생성해 주기 때문에 확장성이 뛰어나고, OWL 전용의 파싱 모듈을 거치기 때문에 데이터와 표현력의 손실이 생

기지 않는다.

향 후, 본 논문에서 설계한 OWL 온톨로지 파서의 기능을 가지는 모듈을 추가 구현한 후, 실제 OWL 온톨로지를 모델링 할 때 RDF view 의 기존의 시스템과 비교하여 온톨로지 객체 생성 시스템의 성능 검증이 필요하다.

참고문헌

- [1] Asuncion Gomez-Perez and Oscar Corcho, "Ontology Language for the Semantic Web," IEEE Intelligent Systems, vol.17, no.1, January/February, 2002, pp.54-60
- [2] Berners-Lee, T., Hendler, J. and Lassila, O., "The Semantic Web," Scientific American, 2001
- [3] Mike Dean et al.(Eds), "OWL Web Ontology Language Reference," W3C Candidate Recommendation 18 August, 2003
- [4] J.J. Carroll, J.D.Roo. Web Ontology Language(OWL) Test Cases. W3C Recommendation 10 February 2004 <http://www.w3.org/TR/owl-test/>
- [5] Peter F.Patel-Schneider, OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation 10 February 2004 <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>
- [6] Brian McBride, "Jena:Implementing the RDF Model and Syntax Specification," Semantic Web Workshop, WWW2001, 21 December, 2001
- [7] Jena2 - A Semantic Web Framework, <http://www.hpl.hp.com/semweb/jena2.htm>
- [8] 박천수, "온톨로지 객체 모델 생성 시스템 설계", 한국 정보처리 학회 20 회 추계 학술대회, 10 권 2 호, 2003, pp.1297-1300
- [9] 이미경, "시맨틱 웹을 위한 온톨로지 파서의 설계", 한국 정보처리 학회 20 회 추계 학술대회, 10 권 2 호, 2003, pp.1109-1112
- [10] Sean Bechhofer, OWL Web Ontology Language Parsing OWL in RDF/XML, W3C Working Group Note 21 January 2004, <http://www.w3.org/TR/2004/NOTE-owl-parsing-20040121/>