

# OWL 온톨로지 파서와 추론 시스템 설계 및 구현

황명권 공현장 김판구

조선대학교 전자정보공과대학 전자계산학과  
광주 동구 서석동 375번지

Tel: +82-62-230-7799, Fax: +82-62-230-7636, E-mail: mg.hwang@gmail.com

## Abstract

의미적인 정보검색을 위한 시맨틱 웹에 대한 연구가 본격화되었다. 시맨틱 웹을 위한 핵심은 개념과 개념들 사이의 관계를 정의한 온톨로지이다. 온톨로지를 분석하고, 분석된 결과에 포함되어 있는 새로운 사실들을 추론하여 가능한 많은 결과를 이끌어 내는 것이 의미적인 정보검색의 기반이라 할 수 있다. 본 논문은 이러한 온톨로지에 정의된 개념들을 분석하는 범용적이고 빠른 파서와 파서를 통해 분석된 사실을 바탕으로 더욱 많은 새로운 사실을 추출할 수 있는 온톨로지 기반의 추론(Inference) 시스템을 설계하고 구현하였다.

## Keywords:

Ontology, Parser, Inference engine

## 1. 서론

기존의 웹은 사람에 의해 제공된 데이터를 다루고 있으며, 이러한 데이터를 처리하여 정보를 보여주는 역할만을 담당하고 있다(Semantic Gap). 최근에는 컴퓨터가 사물들의 개념을 파악하고, 사물들 사이의 관계를 이해하여 새로운 사실을 추론함으로써 사람과 같이 데이터를 의미적으로 처리할 수 있는 시맨틱 웹(Semantic Web)에 대한 연구가 활발해졌다. 시맨틱 웹을 실현하기 위한 핵심은 '컴퓨터가 사물들에 대해 어떻게 개념을 정의하고 관계를 파악하여 처리하느냐'인데 이 역할을 온톨로지(Ontology)가 담당하고 있다. 온톨로지는 상-하위 계층 구조를 기본으로 개념들을 정의하고, 개념들 사이의 관계에 대해 실세계를 모델로 정확히 형성할 수 있어 사람과 기계가 함께 개념들을 이해할 수 있고, 정의된 개념들을 바탕으로 새로운 의미를 도출할 수 있도록 한다.[1] 시맨틱 웹에서 의미적 검색을 위해 온톨로지가 핵심적인 역할을 함으로써, RDF(S), OWL(Web Ontology Language), DAML+OIL(DARPA Markup Language), XOL(Ontology Exchange Language), SHOE(Simple HTML Ontology Extension), OML(Ontology Markup Language) 등의

많은 온톨로지 구축 언어가 개발되었다. 특히, RDF(S)와 OWL은 2004년 2월에 W3C(World Wide Web Consortium)에 의해 국제 표준으로 제정되어 현재 가장 널리 사용되고 있는 온톨로지 구축을 위한 언어이다. 또한 이들 온톨로지 언어들을 기반으로 한 온톨로지 구축 도구도 스탠포드 대학(Stanford University)에서 개발한 Protégé를 비롯하여, 독일의 Karlsruhe 대학의 KAON, 영국 Manchester 대학의 OIEd 등 50여가지가 현재까지 연구 개발되었다.[2] 하지만 이처럼 시맨틱 웹의 실현에 핵심이 되는 온톨로지 구축을 위해 많은 연구가 이루어 졌으나, 여전히 의미적 정보 검색의 실현은 이루어지지 않고 있다. 그 이유는 자원들에 대한 사실(Fact)들을 추출하는 파서(Parser)와 사실들을 바탕으로 새로운 사실을 추론하는 추론 엔진(Inference Engine)에 대한 연구가 미비하기 때문이다. 특히, 각 온톨로지 도구들의 파서는 자신이 작성한 온톨로지 분석에 치중하여 설계되어 있어서, 다른 도구에서 동일한 언어를 사용하여 구축된 온톨로지의 분석은 어렵다. 이에 본 논문에서는 OWL 언어를 기반으로 구축된 모든 온톨로지를 분석하는 범용적이고 빠른 파서와 분석된 사실들을 바탕으로 새로운 사실을 추론하는 추론엔진을 설계하고 구현하였다.

본 논문의 2장은 OWL언어의 특징과 기존의 파서와 추론엔진들에 대한 조사를 서술하고, 3장에서 본 논문의 핵심인 범용적인 OWL 파서와 추론엔진에 대하여 기술한다. 그리고 4장은 구현된 시스템을 실험하였고, 5장에서 본 논문에 대한 결론과 향후 연구 방향에 대해 서술하였다.

## 2. 관련연구

본 연구의 범용적인 OWL 파서를 위해, OWL 언어의 전반적인 내용과 규칙들을 조사하였고, 기존에 가장 많이 사용되고 있는 OWL 언어로 작성된 온톨로지를 파싱하고 추론하는 JENA2와 F-OWL,

그리고 Hoolet에 대하여 조사 및 분석하였다.

### 2.1) OWL

OWL은 RDF(S)의 어휘 표현력을 바탕으로 [표 1]과 같이 rdfs:Class의 하위에 17개, rdf:Property의 하위에 23개의 어휘를 새롭게 확장한 온톨로지 구축 언어이다.

표 1 - 확장된 OWL 어휘

rdfs:Class	rdf:Property
owl:AllDifferent	owl:allValuesFrom
owl:AnnotationProperty	owl:backwardCompatibleWith
owl:Class	owl:cardinality
owl:DataRange	owl:complementOf
owl:DatatypeProperty	owl:differentFrom
owl:DeprecatedClass	owl:disjointWith
owl:DeprecatedProperty	owl:distinctMembers
owl:FunctionalProperty	owl:equivalentClass
owl:InverseFunctionalProperty	owl:equivalentProperty
owl:Nothing	owl:hasValue
owl:ObjectProperty	owl:imports
owl:Ontology	owl:incompatibleWith
owl:OntologyProperty	owl:intersectionOf
owl:Restriction	owl:inverseOf
owl:SymmetricProperty	owl:maxCardinality
owl:Thing	owl:minCardinality
owl:TransitiveProperty	owl:oneOf
	owl:onProperty
	owl:priorVersion
	owl:sameAs
	owl:someValuesFrom
	owl:unionOf
	owl:versionInfo

현재 OWL은 OWL Lite, OWL DL 그리고 OWL Full로 사용 목적에 따라 구분되어 있으며, 온톨로지 구축에 가장 많이 활용되고 있다. 또한 OWL 표준은 OWL Guide, Reference, Semantics and Abstract Syntax, Test Cases, Use Cases and Requirements와 XML Presentation Syntax의 문서로 구성되어 있다. 각 문서에는 OWL에 대한 구체적인 정의와 사용에 대한 스펙이 자세히 제공되고 있어 OWL의 활용분야는 더욱 넓어지고 있다.[3]

### 2.2) JENA2

규칙기반 추론을 수행하는 JENA2는 HP 시맨틱 웹 프로그램 연구실에서 시맨틱 웹을 위한 응용 프로그램들을 구축하기 위해 제공되는 Java 프레임 워크이다. JENA2는 RDF(S)와 OWL의 파싱과 추론을 위한 프로그램 환경을 제공하는 것 외에도 다음과 같은 주요한 기능을 제공한다.[6]

- RDF 언어를 다루기 위한 API 제공
- RDF/XML 내에서의 RDF, N3와 N-Triples를 읽고 쓰는 기능 제공

- OWL 언어를 다루기 위한 API 제공
- 내장 메모리 기반과 영구 저장소(Database) 기반의 환경 제공
- RDQL 제공 : RDF를 위한 질의언어

위와 같은 기능을 제공하지만, OWL을 완벽하게 추론하기에는 다음과 같은 한계점이 존재한다.

- OWL Lite까지만 지원함
- 추론 규칙이 부정(Negation) 표현을 지원하지 못함
- 추론 규칙이 서술어(Predicate)의 항을 한 개 또는 두 개만 허용함

위와 같이, JENA2는 어휘 표현이 풍부한 OWL 언어를 모두 추론하기에는 범위가 좁고, 추론 규칙이 아직 부족하다.

### 2.3) F-OWL

F-OWL은 OWL 전용 추론엔진이다. F-OWL의 OWL 온톨로지 추론루틴은 Flora-2를 사용하여 수행되는데, Flora-2는 F-Logic, HiLog와 Transaction Logic등과 같은 통합언어를 XSB 추론 엔진으로 변환해주는 객체 지향 지식 기반 언어와 어플리케이션 플랫폼을 말한다.[7] F-OWL의 주요한 기능들은 다음과 같다.

- OWL 온톨로지 추론
- Flora-2의 Axiom들을 이용한 지식 일관성 체크
- 기존의 지능형 시스템들과의 호환성을 위한 프로그래밍 인터페이스 제공

F-OWL은 OWL 온톨로지 전용 추론엔진이지만, JENA2와 같이 모든 OWL 어휘를 추론하기에는 추론규칙이 부족한 단점이 있다.

### 2.4) Hoolet

OWL 온톨로지를 추론할 수 있는 또다른 추론엔진인 Hoolet은 First Order Logic을 사용하는 OWL DL 추론엔진이다. Hoolet의 주요 기능들은 다음과 같다.

- First Order Logic을 사용
- Axiom과 규칙의 형태에 제약을 받지 않음

위와 같이 간단한 특징을 가진 Hoolet은 JENA2와 F-OWL처럼 전용추론엔진이라기 보다는 파싱과 일관성 체크엔진 수준의 시스템이다.

### 3. 범용적인 파서와 추론 엔진

표 3 - OWL 속성 정의 부분의 파싱

#### 3.1) OWL 구문의 파싱과 추론 규칙

파싱은 온톨로지 언어로 작성된 구문을 트리플 (Triple)로 변환하는 것이다. 또한, 트리플은 주어 (Subject)-서술어(Predicate)-목적어(Object)로 구성된 문장(Statement)으로 컴퓨터가 알기 쉽게 형식적이고 구조화된 형태를 의미한다. OWL 표준 문서인 OWL Test Cases[4]와 Semantics and Abstract Syntax[5]를 살펴보면 각각의 OWL구문에 대한 트리플로의 변환을 잘 설명하고 있다. 표 2와 3은 OWL의 클래스 정의와 속성 정의에 대한 구문을 트리플로 변환한 것을 보이고 있다.

표 2 - OWL 클래스 정의 부분의 파싱

<pre>&lt;rdf:RDF   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"   xmlns:owl="http://www.w3.org/2002/07/owl#"   xmlns:first="http://www.w3.org/2002/03owl/Ontology/premises001#"   xml:base="http://www.w3.org/2002/03owl/Ontology/premises001" &gt;    &lt;owl:Ontology rdf:about="" /&gt;    &lt;owl:Class rdf:ID="Car"&gt;     &lt;owl:equivalentClass&gt;       &lt;owl:Class rdf:ID="Automobile"&gt;     &lt;/owl:equivalentClass&gt;   &lt;/owl:Class&gt;   &lt;first:Car rdf:ID="car"&gt;     &lt;rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" /&gt;   &lt;/first:Car&gt;   &lt;first:Automobile rdf:ID="auto"&gt;     &lt;rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" /&gt;   &lt;/first:Automobile&gt; &lt;/rdf:RDF&gt;</pre>		
주어	서술어	목적어
first:Car	rdf:type	owl:Class
first:Automobile	rdf:type	owl:Class
first:Car	owl:equivalentClass	first:Automobile
first:car	rdf:type	first:Car
first:car	rdf:type	owl:Thing
first:auto	rdf:type	first:Automobile
first:auto	rdf:type	owl:Thing

표 4 - 추론 규칙 정의

추론	규칙
전이	Triple(Subject ?p)(Predicate "http://-#type")(Object "http://-#TransitiveProperty") Triple(Subject ?x)(Predicate ?p)(Object ?y), Triple(Subject ?y)(Predicate ?p)(Object ?z) =>Assert(Triple(Subject ?x)(Predicate ?p)(Object ?z))
대칭	Triple(Subject ?p)(Predicate "http://-#type")(Object "http://-#symmetricProperty") Triple(Subject ?x)(Predicate ?p)(Object ?y) =>Assert(Triple(Subject ?y)(Predicate ?p)(Object ?x))
역 속성	Triple(Subject ?p1)(Predicate "http://-#inverseOf")(Object ?p2) Triple(Subject ?x)(Predicate ?p1)(Object ?y) =>Assert(Triple(Subject ?y)(Predicate ?p2)(Object ?x))
범주 속성 상속	Triple(Subject ?x)(Predicate "http://-#subPropertyOf")(Object ?y) Triple(Subject ?y)(Predicate "http://-#domain")(Object ?z) =>Assert(Triple(Subject ?x)(Predicate "http://-#domain")(Object ?z))

<pre>&lt;rdf:RDF   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"   xmlns:owl="http://www.w3.org/2002/07/owl#"   xmlns:first="http://www.w3.org/2002/03owl/equivalentProperty/premises001#"   xml:base="http://www.w3.org/2002/03owl/equivalentProperty/premises001" &gt;    &lt;owl:ObjectProperty rdf:ID="hasHead"&gt;     &lt;owl:equivalentProperty&gt;       &lt;owl:ObjectProperty rdf:ID="hasLeader"&gt;     &lt;/owl:equivalentProperty&gt;   &lt;/owl:ObjectProperty&gt; &lt;/rdf:RDF&gt;</pre>		
주어	서술어	목적어
first:hasHead	rdf:type	owl:ObjectProperty
first:hasLeader	rdf:type	owl:ObjectProperty
first:hasHead	owl:equivalentProperty	first:hasLeader

표 2의 클래스 정의 부분에서 “<owl:Class rdf:ID=“Car”>” 부분은 ““Car”를 ‘owl:Class’의 형태(‘rdf:type’)로 정의한다”라고 할 수 있다. 그러므로 주어(first:Car), 서술어(rdf:type), 목적어(owl:Class)의 형태로 트리플을 추출할 수 있다. 표 2와 3과 같이 파싱은 온톨로지에 명시된 사실만을 트리플로 변환한다.

여기서는 사실을 나타내는 트리플 내의 주어, 서술어, 목적어를 이용하여 온톨로지가 명시하지 않은 사실들을 추론할 수 있다. 표 3에서 파싱을 통해 주어인 ‘first:hasHead’가 목적어인 ‘first:hasLeader’와 동일 속성(owl:equivalentProperty)임을 파악하지만, 그 대칭에 대해서 컴퓨터는 파악하지 못한다. 이 부분은 추론 엔진이 담당해야 할 부분이다. 추론 엔진은 일정한 규칙을 정하고 그 규칙에 따라 각 어휘에 해당하는 부분에서 새로운 사실을 추론한다. 표 4는 OWL 온톨로지 추론을 위해 정의한 20가지 규칙 중 일부를 보이고 있다.

#### 3.2) OWL 파서

본 연구에서 범용적이고 빠른 파서를 구현하기 위해 2단계의 과정을 거쳐 분석하도록 설계하였다. 그림 1은 파서의 구조를 나타내고 있다.

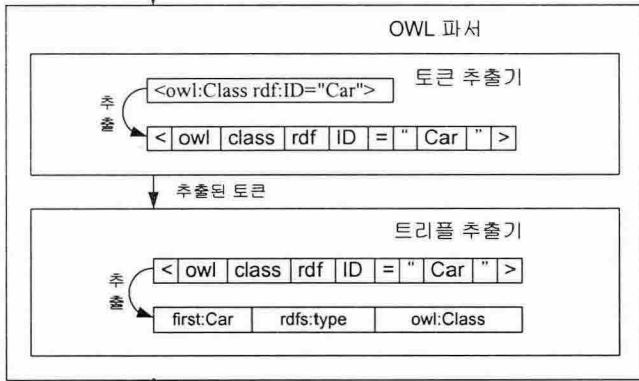
```

<owl:Class rdf:ID="Car">
  <owl:equivalentClass>
    <owl:Class rdf:ID="Automobile"/>
  </owl:equivalentClass>
</owl:Class>
<first:Car rdf:ID="car">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
</first:Car>
<first:Automobile rdf:ID="auto">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
</first:Automobile>

```

OWL 온톨로지

OWL 온톨로지 구문



추출된 트리플

Fact 트리플 저장소

Subject	Predicate	Object
first:Car	rdfs:type	owl:Class
first:Automobile	rdfs:type	owl:Class
first:Car	owl:equivalentClass	Automobile
...	...	...

그림 1-파서의 구조

그림 1과 같이 OWL 파서에서 1단계는 OWL 온톨로지의 '<' 기호부터 '>' 기호까지의 구문을 읽어 토큰으로 나눈다. 나뉘어진 토큰은 2단계의 트리플 추출기로 전달된다. 트리플 추출기에서는 나뉘어진 토큰들에서 S, P, O를 추출하여 Fact 트리플 저장소에 전달한다. 이렇게 전달된 Fact 트리플들은 추론 엔진에서 분석을 위한 기반 데이터가 된다.

### 3.3) 추론 엔진

OWL 파서를 통해 파악된 사실을 바탕으로 새로운 사실을 추론하기 위해 표 4와 같이 일정한 20가지의 규칙을 정의하였다. OWL에서 표 4를 바탕으로 추론이 적용되는 어휘는 표 5와 같다.

표 5-OWL 구문 추론 규칙

적용 추론규칙	OWL 어휘
전이추론	equivalentClass, subClassOf, sameAs, TransitiveProperty, equivalentProperty, subPropertyOf
대칭추론	sameAs, inverseOf, equivalentProperty, SymmetricProperty, equivalentClass, differentFrom, disjointWith, complementOf
역속성추론	inverseOf, inverseFunctionalProperty, FunctionalProperty
범주속성상속추론	range, domain

표 4의 추론 규칙과 표 5의 추론규칙 적용 어휘를 바탕으로 Fact 트리플 저장소에 저장되어 있는 파서에 의해 분석된 트리플을 하나씩 체크함으로써 추론은 시작된다. 그림 2는 전이추론 중 'rdfs:subClassOf' 어휘에 대한 추론 방법을 설명하고 있다.

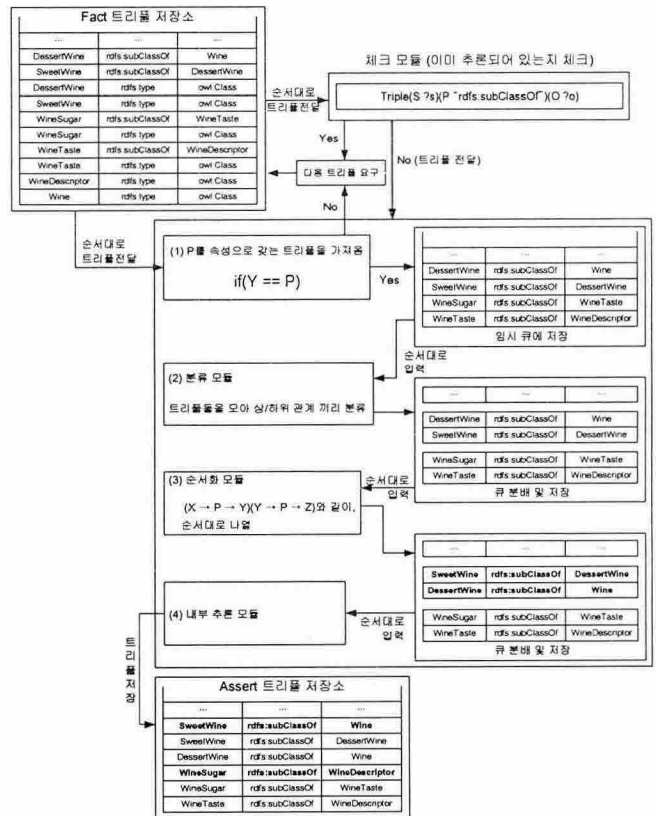


그림 2-rdfs:subClassOf 어휘 추론 방법

그림 2의 추론 방법의 설명은 다음과 같다.

- 추론엔진으로 파싱된 트리플이 입력된다.
- 체크 모듈 : 서술어가 'rdfs:subClassOf'일 경우, 이 어휘에 대한 추론이 이미 되었는지 여부를 검토한다.
- 추론 되지 않은 트리플일 경우, 임시 큐에 저장하고 본 어휘에 대한 추론을 계속 진행하기 위해 Fact 트리플 저장소에 있는 모든 트리플을 입력으로 받는다.
- Fact 트리플 저장소에 있는 트리플 중 서술어가 'rdfs:subClassOf'인 트리플을 추출하여 모두 임시 큐에 저장한다.
- 분류 모듈 : 임시 큐에 저장된 정렬되지 않은 트리플들을 모두 입력 받아 상-하위 관계에 있는 트리플들로 분류하여 또 다른 임시 큐에 저장한다.
- 순서화 모듈 : 상-하위 관계로 정의된

트리플들을 최하위 클래스부터 최상위 클래스로의 일관된 순서로 열거하여 준비된 임시 큐에 저장한다.

- 내부추론 모듈 : 마지막으로 최하위 클래스에서 최상위 클래스 순서로 나열된 트리플들을 ‘배분 법칙’에 의하여 추출된 트리플들을 Assert 트리플 저장소에 저장한다.

이와 같은 추론 방법은 20가지 추론규칙에 대해 각각 설계하고 구현하여, 전체 추론이 일괄적으로 진행된다. 위와 같이 OWL 구문의 파싱을 거친 사실 트리플과 추론엔진에 의한 Assert 트리플들을 함께 이용하여 의미적 정보검색을 실현할 수 있다.

#### 4. 실험

본 연구를 통해 설계하고 구현된 파서와 추론엔진을 통해 시맨틱 웹의 온톨로지의 예로 가장 많이 활용되는 Wine 온톨로지에 대해 파싱하고 추론하였다. 본 시스템의 파서를 통해 총 913개의 Fact 트리플을 얻을 수 있었다. 파싱된 결과를 이용하여 온톨로지 편집과 기본적인 사실을 바탕으로 한 검색은 가능하나 그것이 의미적이라 할 수 없다. 이 Fact 트리플들을 20가지의 추론규칙을 통해 숨겨진 사실들을 추론한 결과 그림 3과 같이 847개의 새로운 트리플을 얻을 수 있었다. 본 추론 엔진은 ‘owl:hasValue’, ‘owl:disjointWith’ 등의 OWL DL과 OWL Full 어휘까지도 추론이 되어 시맨틱 웹에서의 정보검색에 더욱 의미적으로 활용할 수 있다.

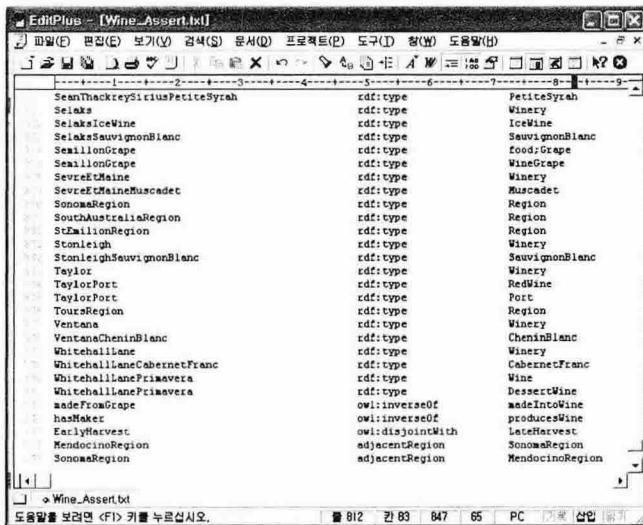


그림 3 - Wine 온톨로지 추론 결과

#### 5. 결론 및 향후 연구방향

본 논문은 시맨틱 웹에서 필수 요소인 온톨로지를 파싱하고, 파싱된 결과에 대하여 추론하는 시스템을 설계하고 구현하였다. 시스템에서 파서는 토큰 추출과 트리플 구성이라는 2단계로 구성되어져 정확하고 빠른 분석이 가능하였으며, 추론엔진은 OWL 언어의 각 어휘에 해당하는 20가지 추론규칙을 정의하고 추출된 사실 트리플을 바탕으로 새로운 사실을 추론할 수 있었다. 구축된 시스템은 기존에 가장 많이 사용되는 추론엔진인 JENA2, F-WOL과 Hoolet의 풍부하지 않은 표현력과 추론 규칙의 부족함이라는 단점을 개선할 수 있었다.

본 연구에서 OWL 언어의 지역 제약 속성 (Local Restriction Property), 교집합 (intersection), 합집합 (union), 여집합 (complement) 등의 표현을 간단한 ‘주어-서술어-목적어’로 구성된 트리플로 추출하는 것은 한계가 있음을 알 수 있었다. 이는 차후에 트리플에서 확장된 그래프 기반의 파싱과 추론으로 이루어져야 할 향후 연구과제이다.

#### References

- [1] J. S. Dong, C. H. Lee, H. B. Lee, Y. F. Li, H. Wang, "A combined approach to checking web ontologies", International World Wide Web Conference, Pages: 714-722, 2004.
- [2] Michael Denny, "Ontology editor survey results", [http://xml.com/2002/11/06/Ontology\\_Editor\\_Survey.html](http://xml.com/2002/11/06/Ontology_Editor_Survey.html), 2002. 06.
- [3] Mike Dean, Guus Schreiber, "OWL Web Ontology Language Reference", <http://www.w3.org/TR/owl-ref/>, W3C Recommendation 10 February 2004
- [4] Jeremy J. Carroll, Jos De Roo, "OWL Web Ontology Language Test Cases", <http://www.w3.org/TR/owl-test/>, W3C Recommendation 10 February 2004
- [5] Peter F. Patel-Schneider, Patrick Hayes, Ian Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax", <http://www.w3.org/TR/owl-semantics/>, W3C Recommendation 10 February 2004
- [6] Andy Seaborne, Chris Dollin, Jeremy Carroll, "Jena - A Semantic Web Framework for Java", <http://jena.sourceforge.net/>
- [7] Harry Chen, Youyong Zou, Lalana Kagal, Tim Finin, "F-OWL: An OWL Inference Engine in Flora-2", <http://fowl.sourceforge.net/>