

## 시멘틱 웹에서의 자동 추론

숭실대학교 강민구·박영택\*

### 1. 서론

시멘틱 웹의 vision은 기존의 웹이 가지는 한계를 극복하는 것이다. 즉, 현재 웹은 정보가 syntactic 구조를 기반으로 표현되기 때문에, 소프트웨어가 정보를 처리하는데 한계가 있었다. 따라서 사람의 일상적인 업무를 대신해서 수행하는 에이전트는 웹에 산재한 문서를 정보로 처리하지 못했고 에이전트가 의미 있는 일을 하기 위해서는 많은 수작업이 뒤따랐다.

이와 같이, 지금까지 사람에 의해서 수행되어온 많은 작업을 자동화하기 위해서는 에이전트 소프트웨어가 해석할 수 있는(machine interpretable) 방식으로 content와 process들을 표현할 때만 가능하다. 시멘틱 웹이 출현하게 된 중요한 동기 중의 하나는 웹을 사용하는 사용자나 웹에서 동작하는 소프트웨어 에이전트 모두가 웹 상의 정보나 서비스를 쉽게 이해하고 처리할 수 있도록 하기 위함이다. 이와 같이 소프트웨어가 해석하고 처리될 수 있는 형태로 정보가 표현되기 때문에 시멘틱 웹에서는 소프트웨어 에이전트가 다양한 추론 방식을 적용함으로써 연역적 추론이 가능하다. 이와 같은 자동 추론 방식은 그동안 인공지능 분야에서 활발히 연구 되어온 결과가 활용된다. 시멘틱 웹에서는 logic layer가 존재하므로써 자동 추론의 기능을 강조된다.

온톨로지(Ontology)는 이러한 semantic interoperability를 가능하도록 하는데 매우 중요한 역할을 수행한다. 그동안 시멘틱 웹 연구 분야에서는 다양한 방식으로 온톨로지를 구축하는 방식들이 제안되어 왔다. 그중에서 DAML+OIL은 온톨로지를 구축하기 위해서 출현한 언어로서, DAML+OIL을 설계하는 과

정에서 고려된 중요한 사항 중의 하나는 자동 추론(automated reasoning)을 활용할 수 있는 시멘틱스이다. 즉, DAML+OIL axiom들은 추론기(reasoner)가 FOPC 수준의 추론이 가능하도록 설계가 되어 있다. 예를 들면, "class Male과 class Female이 disjointWith"이고 "John is Male type"이라는 statement로부터 추론기는 "John이 Female type"이라는 statement가 false라는 것을 추론할 수 있다. 이와 같은 자동 추론 기능은 소프트웨어 에이전트가 다양한 서비스를 수행하는 데 기여할 뿐만 아니라 온톨로지를 설계하고 구현하는 과정에서 매우 중요한 역할을 수행하게 된다.

시멘틱 웹에서 자동 추론 기술은 다음과 같은 용도로 활용이 가능하다.

첫째로, 소프트웨어 에이전트가 정보와 서비스를 "이해"하고 이를 처리할 수 있도록 한다. 시멘틱 웹에서 소프트웨어 에이전트가 정보를 이해하고 처리하기 위해서는 표현된 온톨로지로부터 새로운 내용을 자동 추론하는 기술은 필수적이다. 지능형 에이전트는 추론 기능을 활용하여 온톨로지를 만족하는 facts로부터 implicit하게 표현된 fact를 explicit하게 추론함으로써 추론 기능이 가능할 수 있도록 한다.

둘째로, 온톨로지는 여러 사람들에 의해서 공동으로 구축되는 것이 일반적이다. 이와 같이 공동 작업으로 온톨로지가 구축되는 경우는 온톨로지 상에 inconsistency 문제가 생길 수 있다. 따라서, 온톨로지 구축 도구는 온톨로지 상의 consistency 문제를 처리할 수 있어야 한다. 즉, 대용량의 온톨로지에 존재하는 inconsistency 문제를 수동으로 발견하고 처리하는 것은 거의 불가능한 일이기 때문에 이를 자동으로 처리하기 위한 추론 기능이 필요하다.

셋째로, 온톨로지의 중요한 활용 중의 하나는 온톨로지 합병(ontology integration)이다. 여러 온톨로

\* 중신회원

지들의 integration은 inter-ontology 사이의 class와 property들의 relationship을 정확히 정의하는 것을 필요로 한다. 이런 과정에서 추론은 integrated hierarchy을 추론하고, 온톨로지를 integration할 때 발생하는 inconsistency 문제를 자동으로 탐지하는 데 유용하게 이용된다.

현재는 다음과 같이 온톨로지를 설계하고, 여러 온톨로지를 합병하는 과정에서 자동 추론 방식이 이용되어 inconsistency 문제를 해결하고 있다.

	추론 기능 활용 용도
온톨로지 설계	<ul style="list-style-type: none"> <li>• 클래스 consistency, implied 관계 check</li> <li>• 대용량 온톨로지/다중 author일 때 유용</li> </ul>
온톨로지 integration	<ul style="list-style-type: none"> <li>• Inter-온톨로지 관계 추론</li> <li>• Integrated 클래스 구조/consistency check</li> </ul>
온톨로지 deployment	<ul style="list-style-type: none"> <li>• Facts들이 온톨로지에 consistent한 가를 check</li> <li>• Individual들이 온톨로지 클래스의 인스턴스인 가를 check</li> </ul>

### 1. DAML+OIL 지식의 자동 추론

Stanford KSL 연구팀에서는 DAML+OIL 지식을 autonomous하게 추론하기 위해서 추론 기반 질의응답기를 구축하고 있다. 소프트웨어 에이전트를 지식 서버로 보고 DAML로 표현된 지식베이스를 효과적으로 추론하여 질의·응답을 수행하기 위해서 JTP[1]와 OQL[2]를 활용하고 있다. JTP는 Java로 구현된 추론기로 시멘틱 웹에서 추론하는데 용이한 API를 가지고 있다. 현재 medium 크기의 DAML+OIL 지식베이스로부터 질의·응답이 가능한 것을 테스트 하였고 앞으로 대용량 지식베이스에 테스트를 할 예정으로 있다. OQL(OWL Query language)는 OWL로 표현된 시멘틱 웹 지식을 위한 ACL이다.

스탠포드 KSL 연구팀에서는 DAML+OIL로 표현된 정보로부터 새로운 정보를 추론하기 위해서 DAML+OIL을 FOL로 변환하는 방식을 사용하고 있다[3]. W3C에 제안된 DAML+OIL specification은 FOL 시멘틱스와 model-theoretic 시멘틱스를 포함하고 있기 때문에[4] DAML+OIL을 FOL로 변환하면 FOL 자동 추론기(automated theorem prover)를 활용해서 추론이 가능해진다.

DAML+OIL은 기본적으로 RDF와 RDFS에 새로운 property, class, constant를 추가한 형태이므로 DAML+OIL의 모든 statement는 RDF statement이

다[5]. 따라서, DAML+OIL을 FOL로 변환하기 위해서는 RDF statement를 FOL로 변환하기 위한 간단한 규칙(rule)들을 사용한다. DAML+OIL의 FOL로의 변환은 아래의 순서로 이루어진다.

- ① DAML+OIL statement를 RDF statement로 mapping 한다.
- ② 변환된 RDF statement는 property P, subject S, object O를 포함하고 있는데 이것을 FOL sentence Form인 (PropertyValue P S O)형태로 다시 변환한다.
- ③ 변환된 statement의 property, class, constant들에 axiom들 add한다.

예를 들면 다음과 같은 DAML+OIL 지식은 “Rose” 클래스와 “MyFavoriteFlower” 인스턴스를 표현하고 있다. DAML+OIL은 클래스를 표현할 때 Restriction 클래스를 이용하여 클래스의 시멘틱스를 보다 구체화할 수 있는 기능이 있다. 따라서, 다음의 지식에서는 “Rose” 클래스의 인스턴스들은 “hasRoseColor” property를 가지고, 이 property의 값은 “RoseColor” 클래스 인스턴스라는 것을 보여 주고 있다.

```

<daml:Class rdf:ID = "Rose">
  <rdfs:subClassOf rdf:resource = "Flower"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource = "hasRoseColor"/>
      <daml:toClass rdf:resource = "RoseColor"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>

<Rose rdf:ID = "MyFavoriteFlower">
  <hasRoseColor rdf:resource = "Red"/>
</Rose>

```

위에 표현된 DAML+OIL 지식은 “MyFavoriteFlower” 인스턴스를 잘 표현해 주고 있다. DAML+OIL 시멘틱스를 이해하는 사람은 MyFavoriteFlower의 hasRoseColor property의 값이 Red이고 Red의 type은 RoseColor type이라는 것을 이해할 수 있다. 그러나, 소프트웨어 에이전트는 Red가 Rose

Color type이라는 syntactic 내용이 없기 때문에 추론 기능을 활용할 때 Red가 RoseColor type이라는 것을 추론할 수 있다. 이와 같은 추론은 기본적으로 unification을 바탕으로 하는 theorem proving 기법을 활용한다.

위에서 표현된 DAML+OIL 지식은 변환기에 의해서 다음과 같은 FOL로 변환이 되는데, 이와 같은 변환은 자동 추론기가 DAML+OIL axiom을 활용하여 자동 추론을 가능하게 해준다. 아래의 변환된 예에서 보면 알 수 있듯이 DAML+OIL 지식은 시맨틱스 상으로 동일한 의미를 가지는 FOL로 변환이 된다.

```
(Type Rose daml:Class)
(PropertyValue rdfs:subClassOf Rose Flower)
(Type GnR daml:Restriction)
(PropertyValue rdfs:subClassOf Rose GnR)
(PropertyValue daml:onProperty GnR hasRoseColor)
(PropertyValue daml:toClass GnR RoseColor)

(Type MyFavoriteFlower Rose)
(PropertyValue hasRoseColor MyFavoriteFlower Red)
```

그러나, 우리가 관심이 있는 Red의 type이 RoseColor라는 predicate (Type Red RoseColor)는 assert가 안되어 있으므로 (Type Red ?Type)이라는 질의어를 처리하는 것이 불가능하다. 이러한 추론을 위해서 DAML+OIL axiom을 forward chaining 추론함으로써 (Type Red RoseColor)를 추론하게 된다. 현재 DAML+OIL axiom은 10개 클래스의 14개 axiom과 26개 프로퍼티의 69개의 axiom이 정의되어 있다[6]. 다음은 DAML+OIL의 subClassOf axiom이다.

```
(<=>(PropertyValue subClassOf ?csub ?csuper)
  (and(Type ?csub rdfs:Class)
    (Type ?csuper rdfs:Class)
    (forall(?x)(=>(Type ?x ?csub)
      (Type ?x ?csuper))
```

자동 추론기는 위의 axiom과 앞에서 변환된 FOL facts들에 unification을 적용하면 Rose를 ?csub로 치환(substitution)하고, GnR을 ?csuper로 치환하여 다

음과 같은 결과를 유도한다.

```
(=>(PropertyValue subClassOf Rose GnR))
(=>(Type MyFavoriteFlower Rose)
  (Type MyFavoriteFlower GnR)))
```

(PropertyValue subClassOf Rose GnR)과 (Type MyFavoriteFlower Rose)이 위의 mapping된 statement에서 이미 존재하므로 reasoner는 (Type MyFavoriteFlower GnR)을 reasoning할 수 있게 된다. 이 결과와 아래의 toClass axiom을 이용하면 Red가 RoseColor type임을 reasoning 할 수 있게 된다.

```
(=>(and(PropertyValue onProperty ?r ?p)
  (PropertyValue toClass ?r ?c))
  (forall(?i)
    (<=>(Type ?i ?r)
      (forall(?j)
        (=>(PropertyValue ?p
          ?i ?j)
          (Type ?j ?c))))))
```

Reasoner가 GnR을 ?r으로, hasRoseColor를 ?p로, RoseColor를 ?c로, MyFavoriteDrink를 ?i로 치환하면 toClass axiom은 아래와 같은 결과를 도출하게 된다.

```
(=>(PropertyValue hasRoseColor MyFavoriteFlower ?j)
  (Type ?j RoseColor))
```

따라서, Reasoner가 Red를 ?j와 치환하게 되면 우리가 원하는 (Type Red RoseColor)를 추론하게 된다.

## 2. 온톨로지 병합에서의 자동 추론

시맨틱 웹에는 다양한 온톨로지가 존재하므로, 소프트웨어 에이전트는 온톨로지 변환(ontology translation) 문제에 직면하게 된다. 각각의 소프트웨어 에이전트는 자신의 지식을 표현하기 위해서 서로 다른 온톨로지를 활용하고 있기 때문에 서로 대화를 하기 위해서는 온톨로지 변환 과정을 필요로 한다. 여기서, 온톨로지 변환은 하나의 온톨로지를 이용하여 표현된 dataset을 다른 온톨로지를 이용하여 표현하

는 것을 의미한다[7].

기존의 온톨로지 변환은 syntactic 변환만을 다루고 있다[8]. 이와 같은 이유는 semantic 변환을 자동으로 한다는 어려움을 극복하지 못하고, human 전문가가 온톨로지 변환 과정을 마지막으로 확인하고 이를 관리하는 과정이 필요하기 때문이다. 그러나, 완전 자동은 아니더라도 가능한 부분은 자동화함으로써 human 전문가의 부담을 줄여주는 것은 큰 의미가 있는 작업이다. 현재, Yale 대 연구팀은 온톨로지 변환(translation)을 위해서 온톨로지 병합(merge)과 자동 추론 방식을 활용하는 연구를 수행하고 있다 [7]. 온톨로지 병합은 human 전문가의 도움을 받아서 이루어지고, 합병된 온톨로지의 변환은 OntoEngine 이라는 forward/backward chaining 추론기관을 활용하고 있다.

Yale 연구팀은 두개의 온톨로지의 서로 다른 term을 연관시키기 위해서 bridging axiom을 정의해서 forward chaining이나 backward chaining을 적절히 활용하는 방안을 제안하고 있다. 이러한 연구 방식은 기본적으로 Stanford KSL의 연구팀과 유사한 연구 방법을 제안하고 있는 데, DAML+OIL로 표현된 내용을 FOL 형식으로 변환하고 여기에 theorem prover를 적용하는 방법이다. 이러한 방식은 기존의 인공지능 연구에서 많이 처리해온 방식으로 modus-ponens을 포함한 다양한 추론 기법이 적용될 수 있기 때문이다.

OntoEngine은 Yale 연구팀이 온톨로지 변환을 위해서 만든 추론 기관으로 INF(Implicative Normal Form) 형식의 clause들을 처리할 수 있다. 온톨로지가 OntoEngine에 의해서 로드될 때, 대부분의 axiom들은 INF 형식으로 변환이 된다. INF형식은 다음과 같이 조건부(premise)와 결론부(conclusion)에 atomic formula 들의 conjunction이 있다.

$$P_1 \wedge \dots \wedge P_j \dots \wedge P_n \Rightarrow Q_1 \wedge \dots \wedge Q_k \dots \wedge Q_m$$

OntoEngine은 온톨로지의 fact들을 변환하기 위해서 axiom들을 활용한다. 다음은 참고서적(bibliography) 도메인에서 온톨로지 변환을 위한 axiom의 예이다. 이 예에서는 Yale 대학의 참고서적 표현을 위한 온톨로지(@yale\_bib:)와 CMU 대학의 참고서적 표현을 위한 온톨로지(@cmu\_bib:) 간의 변환을 다루고 있다.

(forall (?a - Article ?p - Proceedings ?t - String)

```
(iff (and (contains ?p ?a) (title ?p ?t))
      (and (@cmu_bib:inProceedings ?a ?p)
            (@yale_bib:booktitle ?a ?t))))
```

위의 axiom은 CMU 온톨로지의 dataset인 (inProceedings McDermott80 Proceedings01)이 Yale dataset인 (booktitle McDermott80 - Inproceedings "Proc. AAAI 1" - Literal)과 같다는 것을 의미한다. 이러한 변환이 가능하기 위해서는 부수적으로 (title Proceedings01 "Proc. AAAI 1")과 (contains Proceedings01 McDermott80)이 만족되어야 한다.<sup>1)</sup>

다음은 OntoEngine에서 사용하고 있는 기본적인 forward chaining 추론 과정을 요약한 것이다.

**Procedure** Process(facts)

**Input** : dataset as facts

**Repeat**

*oneFact* = next fact in facts

*anynew* = Forwardchaining(*oneFact*)

**Until** last fact in facts

**Function** Forwardchaining(fact) return boolean

Get best INF clause from corresponding

Premises

*newFacts* = Modus\_Ponens(fact, clause)

**Repeat**

*anyFact* = next fact in *newFacts*

*anyNew* = Forwardchaining(*newFact*)

**Until** last fact in *newFacts*

**Return** *anyNew*

**Function** Modus\_Ponens(fact,clause) return facts

**Repeat**

*oneAtf* = next AtomicFormula from leftside of clause

*substi* = Unify(*oneAtf*, fact)

Add *substi* into the whole substitutions and check its consistency

**Until** get the consistent whole substitutions

Get *newFact* by substituting rightside of

1) 여기서는 간략한 설명을 위해서 contains predicate에 대한 axiom을 생략하였다.

clause with substitutions

**Repeat**

*newFact* = next fact in newFacts

**If** *newFact* belongs to target ontology

**Then** store it

**Else** add it into facts which will be returned for further inference

**Until** last fact in newFacts

**Return** facts

### 3. RuleML 추론 방식

MIT는 이기종 rule system들의 상호 운용성을 높이는 연구를 하고 있다. MIT는 이를 위해서 Benjamin N.Grosz를 중심으로 RuleML 기반의 자동추론방법을 연구하고 있다[9]. 현재 e-business 등 많은 분야에서 Rule을 이용하여 지식을 표현하고 있는데 MIT에서는 RuleML을 이용한 시스템으로 이기종 시스템간의 지식을 공유하는 방법을 연구하고 있다.

#### 3.1 RuleML

RuleML(Rule Markup Language)는 XML을 기반으로 하는 rule언어이다[10]. Rule은 시맨틱 웹이 지향하고 있는 시스템을 개발하기 위해서는 꼭 필요한 기술이다. 그러나 시맨틱 웹 연구는 온톨로지 마크업 언어에 치중한 나머지 rule에 대한 연구는 부족한 실정이었다. RuleML Initiative는 2000년 8월에 발족 되었으며 시맨틱 웹 연구에 있어서 온톨로지 마크업 언어와 rule언어 사이의 차이를 줄이기 위해 연구를 진행 하고 있다.

RuleML은 이기종 시스템 사이의 rule 교환을 가능하게 하고 rule 지식 표현을 위해 XML 문법을 사용하므로 상호 운용성이 풍부하다. RuleML은 DTD를 이용해서 구조를 표현하고, 이에 적합한 RuleML을 이용하여 의미를 표현한다. 현재 RuleML Initiative는 Java Rules Engine API specification, W3C RDF working group, DAML group, W3C P3P Activity등과 활발한 협력을 하고 있으며 실제로 RuleML 0.8과 DAML+OIL이 Web Ontology Group에서 1.0 version을 목표로 개발 중에 있다. RuleML은 다음과 같은 특징을 갖는다.

- RDF/RDFS, DAML+OIL, ISO Topic Map, IEEE Upper Ontologies와 같은 지식표현 기술

과 메커니즘을 공유한다.

- Rule을 이용하기 위해서 온톨로지를 이용한다.
- Rulebase, predicate symbol과 같은 온톨로지 태그를 지원한다.
- XML Schema와 XML DTD를 이용하여 정의할 수 있다.
- XML의 부가적인 문법을 지원하고, stylesheet를 사용 할 수 있다.
- 기존에 존재하는 W3C 표준 문서들에 rule메커니즘을 제공할 수 있다.

#### 3.2 RuleML을 이용한 자동추론

RuleML로 표현한 rule은 XSLT stylesheet를 이용하여 기존의 자동추론기가 사용할 수 있는 포맷으로 변환한다. 변환된 포맷을 이용하여 JESS, CLIPS, Prolog 등의 자동추론기가 추론하여 결과를 도출한다. 현재 RuleML의 최신 버전인 RuleML0.8은 RuleML 온톨로지를 Jess 포맷으로 바꿔주는 XSLT가 포함되어 있다.

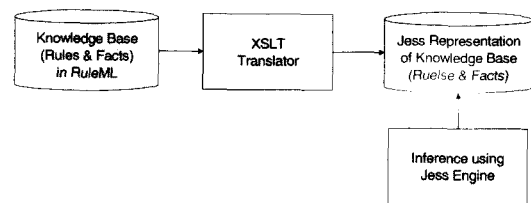


그림 1 RuleML을 이용한 자동 추론 과정

아래는 RuleML로 표현한 간단한 rule의 예제이다.

```

<rulebase label="myRules">
  <imp>
    <_head><atom>
      <rel>likes</rel>
      <ind>John</ind>
      <var>x</var>
    </atom></_head>
    <_body><atom>
      <rel>likes</rel>
      <var>x</var>
      <ind>wine</ind>
    </atom></_body>
  </imp>
</rulebase>
  
```

```
</imp>
</rulebase>
```

위의 RuleML rule은 XSLT를 이용하여 Jess에서 사용할 수 있도록 변환된다.

```
(defrule myRules-1
  "This rule has been generated from
  RuleML"
  (likes ?x wine)
=>
  (likes John ?x)
```

위의 RuleML이 Prolog 형태로 변환된 모습은 아래와 같다.

```
like(John,X) :- likes(X, Wine).
```

MIT에서는 RuleML을 이용 가측관계에 대한 rulebase를 만드는 작업을 수행했으며, XSLT를 이용하여 XSB, JESS, n3/cn engine에서 자동추론 할 수 있는 시스템을 연구중이다.

#### 4. FaCT 추론 기관

DAML+OIL은 SHIQ Description Logic[11]과 equivalent한 특성이 있어서 FaCT 추론 기관이 DAML+OIL 온톨로지를 구축하는 과정에 활용될 수 있다. DAML+OIL를 설계할 때 고려 사항 중의 하나는 클래스 consistency 및 subsumption 관계를 파악하는 알고리즘이 decidable 해야 한다는 것이었는데, 이것은 추론 엔진의 활용을 위함이었다.

FaCT(Fast Classification of Terminologies)는 다음과 같은 특성을 가지고 있으므로 실시간으로 DAML+OIL 온톨로지 구축에 활용이 가능하다.

- ① SHIQ 추론기에 활용이 될 수 있다.
- ② Optimized tableaux 구현 방식을 사용하여 속도가 빠르다.
- ③ CORBA 기반의 클라이언트/서버 구조를 가지고 있다.

FaCT(Fast Classification of Terminologies)는 클래스 consistency와 implicit한 subsumption 관계를

매우 빠르게 처리하기 위한 optimized 알고리즘을 채택하고 있다. 예를 들면, 2,740개의 클래스로 구성된 온톨로지의 consistency를 check하고, 이들간의 클래스 hierarchy를 구성하는데 700MHz Pentium III CPU로 약 45초가 소요된다[12]. 이러한 이유로 인해서 FaCT는 OilEd[13]와 Protege[14] 시스템에 적용이 되어서 온톨로지 구축에 활용이 되고 있다.

OilEd나 Protege로 구축된 DAML+OIL 온톨로지는 SHIQ Description Logic으로 변환되고 CORBA 인터페이스를 통해서 FaCT로 전달된다[15]. 그러면, FaCT는 전달된 온톨로지의 inconsistency가 있는지를 확인하고 이를 지적하게 된다. 또한, implicit한 subsumption이 존재하면 이를 explicit하게 만들어 주기 위해서 axiom을 추가하게 된다. FaCT가 발견한 inconsistency는 사용자의 확인을 받아서 최종 수정되는 단계를 거치게 된다.

Description logic은 DAML+OIL 온톨로지 구축을 위해서 다음과 같은 기능을 수행할 수 있다.

- ① Subsumption check  
하나의 concept이 다른 concept의 sub-concept 인지를 check한다.
- ② Consistency check  
Concept 정의에 있어서 inconsistency가 존재하는 지를 check한다.
- ③ Taxonomy construction  
Concept hierarchy를 explicit하게 구축한다.
- ④ Classification  
주어진 concept을 subsume하거나, 주어진 concept이 subsume되는 concept를 알아낸다.

#### 5. 자동 추론 관련 Projects

DAML+OIL working 그룹에서는 DAML 기반 시스템에 추론 기법을 적용하여 DAML query 언어, DAML justification 언어를 구축하는 것을 목적으로 연구를 수행하고 있다. 다음은 2001년 7월 기준 자동 추론 연구를 수행하는 DAML contractor들과 간략한 연구 내용이다.

#### 6. 결 론

시멘틱 웹은 많은 온톨로지로 구성되며 사용자 뿐만 아니라 소프트웨어 에이전트도 이러한 온톨로

연구팀	연구 내용
Cycorp	• Texanomy 추론을 위한 OpenCyc
Lockheed Martin, VIS, Kestrel	• UBOT 프로젝트 : DAML 온톨로지의 consistency checking • ConsVISor 프로그램 : axiom이 DAML을 만족하는지 check
Stanford KSL	• DAML로 표현된 시스템의 추론 구조 및 방법 • JTP를 포함한 다양한 추론 응용 연구
Teknowledge	• 추론을 위한 theorem prover 연구
UMBC	• ITTALKS 응용 분야에 활용되는 에이전트의 추론 구조
University of Manchester	• SHIQ description logic을 위한 FaCT 추론 엔진 • OilEd와 Protege에 FaCT를 적용하여 온톨로지 구축 과정에서 subsumption 및 consistency check
Yale/BBN/Kestrel	• 에이전트로부터 받은 서비스 description을 planning 기법으로 처리

지를 활용하게 된다. 온톨로지의 구축 과정은 기존의 지식베이스 구축 과정과 마찬가지로 inconsistency, incompleteness, incorrectness 문제를 극복할 때, 그 효용성이 크게 된다. 대부분의 온톨로지는 대용량의 크기를 가질 것이고, 공동 작업에 의해서 다수의 human이 온톨로지 구축 작업에 참여할 것이다. 이와 같은 상황에서 위의 세가지 문제를 사람이 수작업으로 해결하기는 불가능하다. 따라서, 인공지능에서 연구되어온 자동 추론 기법과 이를 support해주는 논리 기반의 지식표현 방식이 필수적일 것이다.

또한, 다중 온톨로지를 사용하는 경우에 온톨로지 변환의 문제가 직면되고 있는 실정이다. 서로 다른 온톨로지를 이용하는 경우에 에이전트는 온톨로지 변환 과정을 통해서만 문제를 해결할 수 있다. 물론, 최종적인 단계에서는 온톨로지 변환을 확인하고 검증하는 것은 human expert이지만 대부분의 작업은 자동 추론 방식으로 처리되어야 문제의 복잡도를 줄일 수 있을 것이다.

시멘틱 웹은 궁극적으로 소프트웨어가 정보와 서비스를 이해하여 사람이 수작업으로 수행할 대부분의 업무를 자동화하는 것이다. 기존의 syntactic 기반의 웹에서는 불가능하였던 소프트웨어 에이전트의 작업이 시멘틱 웹에서는 가능토록 하는 것이 궁극적인 목표이다. 이와같은 에이전트가 동작을 하여 서비스를 자동으로 제공하기 위해서는 자동 추론 기법이 필수적으로 활용되어야 한다.

### 참고문헌

- [1] <http://www.ksl.stanford.edu/software/JTP/>
- [2] Richard Fikes, Deborah McGuinness, and Sheila McIlraith, "Tools for DAML-Based Services, Document Templates, and Query-Answering," Stanford University, 2002.
- [3] Deborah L. McGuinness, Richard Fikes, James Hendeler, and Lynn Andrea Stein, "DAML+OIL: An Ontology Language for the Semantic Web," IEEE Intelligent Systems, September, 2002.
- [4] F. van Harmelen et al., "A Model-Theoretic Semantics for DAML+OIL(March 2001)," W3C Note 18, World Wide Web Consortium, Dec. 2001; [www.w3.org/TR/2001/NOTE-daml+oil-model-20011218](http://www.w3.org/TR/2001/NOTE-daml+oil-model-20011218)
- [5] D. Connolly et al., eds, Annotated DAML Ontology Markup, World Wide Web Consortium, Note 18, Dec. 2001, [www.w3.org/TR/daml+oil-walkthru](http://www.w3.org/TR/daml+oil-walkthru)
- [6] Richard Fikes, and Deborah L McGuinness, "An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL," Stanford University, March, 2001.
- [7] Dejing Dou, Drew McDermott, and Peishen Qi, "Ontology Translation by Ontology Merging and Automated Reasoning," Yale University, 2002.
- [8] H.Chalupsky, "Ontomorph: A translation system for symbolic logic," In Proc. Int'l. Con. On Principles of Knowledge Representation and Reasoning, pp. 471-482, 2000. San Francisco: Morgan Kaufmann.
- [9] Benjamin N. Groszof, "Representing E-Business Rules for the Semantic Web : Situated Courteous Logic Programs in RuleML," MIT Sloan School of Management, Dec., 2002.
- [10] Benjamin N. Groszof, "Standardizing XML Rules : Preliminary Outline of Invited Talk," MIT Sloan School of Management, 2000.
- [11] I. Horrocks and U. Sattler, and S. Tobies, "Practical reasoning for expressive description logics," number1750 in LNAI, pp. 161-180, Springer-verlag, 1999.

[12] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens, "OilEd" a Reason-able Ontology Editor for the Semantic Web," University of Manchester, 2001.

[13] <http://oiled.man.ac.uk/>

[14] <http://protege.stanford.edu>

[15] <http://www.cs.man.ac.uk/~horrocks/FaCT/>

---

### 강민구



2002 경희대학교 행정학 학사  
2002~현재 숭실대학교 컴퓨터학부 석사과정  
관심 분야 : 인공지능, 에이전트, 시멘틱 웹  
E-mail : happytest@ailab.ssu.ac.kr

---

### 박영택



1978 서울대학교 전자공학과 학사  
1980 KAIST 전산학과 석사  
1992 U. of Illinois at Urbana-Champaign 전산학과 박사  
1981~현재 숭실대학교 컴퓨터학부 교수  
관심 분야 : 인공지능, 에이전트, 기계학습  
E-mail : park@comp.ssu.ac.kr

---

### ● 제30회 춘계학술발표회 ●

- 일 자 : 2003년 4월 24~26일
- 장 소 : 제주대학교
- 문 의 처 : 한국정보과학회 사무국 Tel. 02-588-9246/7  
<http://www.kiss.or.kr>, E-mail:kiss@kiss.or.kr