

EOL : SUNHI 표현범위를 가진 인식론적 온톨로지 표현 언어

이건수^o 홍인표 김민구
아주대학교 정보통신전문대학원
{lks7256^o, ntnplus0, minkoo}@ajou.ac.kr

EOL : Epistemological Ontology Language with SUNHI Expression Power for Ubiquitous Computing Environment

Keonsoo Lee^o, Inpyo Hong, Minkoo Kim
Graduate School of Information and Communication, Ajou University

요 약

유비쿼터스 컴퓨팅 환경에서 서비스를 제공함에 있어 능동적인 수행 능력은 사용자의 만족도를 높여주는 핵심 요소이다. 시스템의 기능을 부여하기 위해서는 지식을 관리, 처리, 활용하는 기능이 필요한데, 이들 기능은 그 지식이 어떻게 표현되어 있는지에 큰 영향을 받는다. 일차 술어 논리 기반 지식 표현 방법은 폭넓은 표현 범위와 유연한 지식 정의, 추론 방법으로 선호되고 있지만, 복잡한 계산 비용을 갖고 있기 때문에, 전문적인 지식 처리 시스템이 아닌 경우, 불필요한 계산 비용이 소요된다. Description Logic은 Frame기반 지식 표현 방식으로 일차 술어 논리를 사용하는 것보다 지식을 표현할 수 있는 범위는 제한적이지만, 빠른 추론 결과를 보장해 준다. 유비쿼터스 컴퓨팅 환경에서는 분산된 다양한 오브젝트들이 협력 과정을 통해 사용자에게 능동적인 서비스를 제공하게 되고, 이들 개별적인 오브젝트들은 저사양의 계산 능력을 갖고 있다고 가정된다. 그러므로, 저사양의 컴퓨팅 오브젝트들을 조합하여 능동적인 서비스를 성공적으로 제공하기 위해서, 각각의 오브젝트들은 제한된 지식을 효과적으로 관리할 수 있는 방법이 필요하다. 이를 위해 본 논문에서는 Frame 기반의 Description Logic을 기반으로 SUNHI의 표현 범위를 가진 인식론적 온톨로지 표현 언어를 제안하고, SUNHI의 표현 범위의 효율성을 증명하고자 한다.

1. Introduction

유비쿼터스 컴퓨팅 환경에서, 사용자는 스스로의 명시적인 명령 없이 수행되는 자율적인 서비스를 기대하고 있다 [6]. 이러한 자율화된 서비스를 제공하기 위해서는 각 서비스 수행 주체가 되는 오브젝트들의 능동적인 동작이 보장되어야 한다. 이는 곧, 지식 처리 기능의 보편화를 의미한다. 가령, 간단한 유비쿼터스 컴퓨팅 환경에서의 서비스를 하나 가정해 보자. 사용자가 TV 앞의 쇼파에 앉으면, 사용자가 평소 즐겨보던 방송을 자동으로 상영하는 서비스를 위해서는, 사용자의 신원을 인식하고, 공간안에서의 사용자의 위치를 파악하고, 그 사용자의 선호 방송 정보를 추출해내고, 그 방송이 현재 제공 가능한지, 가능하지 않다면, 대체 방송은 무엇인지를 결정하고, 적절한 음량을 선택하는 일련의 과정을 거쳐 TV를 동작시키게 된다. 이 모든 과정이 자동으로, 또 능동적으로 수행되기 위해서는 사용자, 공간, TV, 방송에 대한 지식이 필요하다.

지식을 표현하고, 그 지식에 근거하여 새로운 지식을 추론하고, 추론된 지식을 활용하기 위해 여러 가지 방법들이 연구되어 왔다. 이들 방법으로는 수리논리 (Mathematical Logic), 의미망 (Semantic Network), 프레임 (Frame), 생성 규칙 (Production Rules), 그리고 이들 방법의 조합인 합성 (Hybrid)등이 있다. 이들 중, 수리 논리를 기반으로 한 일차 술어 논리 기반 표현 방법 (First Order Predicate Calculus)이 지식 기반 시스템을 구현하는데 가장 많이 이용되고 있다. 이는 모델을 표현함에 있어 모호성을 배제할 수 있고, 기술이 편리하며, 계산의 정확성을 검증하기가 용

이하기 때문이다. 이처럼, FOPC 기반 지식 처리 방법의 경우, 지식 표현의 한계가 없으며, 표현된 지식의 추론 결과 증명이 용이하다는 장점이 있지만 동시에, 추론 과정이 복잡하고, 계산 비용이 크다는 단점이 존재한다. 즉 전문적인 지식 관리 시스템이 아니라면, FOPC 기반 지식 처리 방법은 관리, 유지 및 활용 비용이 너무 크다.

유비쿼터스 컴퓨팅 환경에서의 서비스는 단일 장치가 독자적으로 수행하기 보다는 주변의 다른 컴퓨팅 오브젝트들과의 협력 과정을 통해 이루어진다. 위의 예에서 살펴본 사용자에게 적절한 방송을 상영하는 서비스의 경우, 사용자의 위치/신원 파악을 위한 센서와 실제 방송을 상영해주는 TV의 협력으로 서비스가 수행된다. 이때, 필요한 지식 처리를 위한 추론 시스템은 책임 문제가 발생한다. FOPC 기반 지식 처리 시스템을 사용한다고 했을 때, TV나 센서가 이러한 고비용 계산을 필요로 하는 추론 시스템을 관리하는 것에는 우리가 따르기 마련이다. 결국, 저사양의 컴퓨팅 오브젝트들의 분산 환경에서 효율적인 지식 관리를 위한 방법이 필요하다. 이런 요구에 부합하여, Description Logic을 기반으로 온톨로지를 사용하는 방법이 제안되었다.

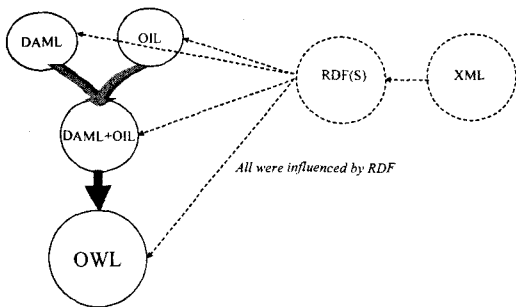
Description Logic은 Frame 기반 지식 표현 방법으로 단일 객체를 표현하는 컨셉 (Concept) 과 객체들 사이의 관계를 표현하는 롤 (Role)로 지식을 표현한다. 이러한 지식은 크게 술어지식기반 (Terminological Knowledge Base) 과 영언지식기반 (Assertional Knowledge Base) 으로 나뉜다. Description Logic 역시 단계별 표현 범위를 갖고 있고, 표현 범위가 넓을수록 추론을 위한 계산 복잡도가 증가한다. 본 논문에서는 Description Logic을 통해 지식을 표현함에 있어서 사용자가 직관적으로 이해할 수 있는 언

어로 EOL을 제안한다. EOL은 직관적인 지식 정의 및 활용이 가능하며 Tbox와 Abox 지식을 모두 표현할 수 있다. 본 언어가 제공하는 지식 표현 범위는 주요 적용 도메인인 유비쿼터스 컴퓨팅 환경에서 자동화된 서비스를 제공하기 위해 필요한 지식을 표현하며 동시에 적절한 계산 복잡도를 확보하기 위하여 SUNHI의 표현 범위를 지원한다.

2. Background

2-1. Description Logic과 Ontology

온톨로지는 공유되는 개념에 대한 정형화되고 명백한 명세서로 정의된다 [7]. 공유되는 개념이란 의미는 환경에 대한 모델을 만들고, 그 모델에 대한 합의가 이루어짐을 의미한다. 즉 동일한 환경에 존재하는 다양한 시스템들이 지능적인 동작을 수행하기 위해 작성한 환경 모델을 서로 이해할 수 있다는 것이다. 정형화되었다는 것은 기술된 모델의 문법이 정해져 있기 때문에 기계가 읽을 수 있다는 것이고, 명백하다는 점은 모델 기술을 위한 문법을 통해, 컨셉, 속성, 관계, 제약 조건 등 모델을 기술하기 위한 요소들이 명시적으로 선언되어 있어야 한다는 점이다. 이렇게, 환경에 대한 정형화되고 명백한 명세서를 공유함으로써, 그 환경에 존재하는 다양한 시스템들은 지식을 공유할 수 있고, 보다 지능적인 협업 동작을 보여줄 수 있게 된다. 이처럼, 지식 공유를 위한 수단으로 온톨로지가 사용됨에 따라, 온톨로지를 표현하기 위한 방법들이 연구되었다. 온톨로지를 표현하기 위한 언어로서는 RDF, OIL, DAML, DAML+OIL, 그리고 OWL 등이 존재하는데, 이들 언어의 관계는 <그림 1>과 같다.



<그림 1> 온톨로지 표현 언어

이들 언어가 표현하는 온톨로지는 프레임 기반의 지식 표현 방법을 사용하고 있는데, 개념(Concept) 이라는 프레임 안에 또 다른 개념과의 관계를 역할(Role)이란 속성으로 갖고 있게 된다. 가령, '아버지'란 개념을 살펴보자. 민중 국어사전에 의하면, 아버지는 자녀를 둔 남자를 자식에 대한 관계로 일컫는 말을 의미한다. 아버지라는 개념은 isA 관계로 남자이고, hasA 관계로 자식을 갖는다. 이를 OWL로 표현하면 <표 1>과 같다.

<표 1> OWL 예제

```

<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:domain rdf:resource="#Father"/>
  <rdfs:range rdf:resource="#Human"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasSex">
  <rdfs:domain rdf:resource="#Human"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#resource"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="Human"/>
<owl:Class rdf:ID="Male"/>

<owl:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#Human" />
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasSex" />
    <owl:allValuesFrom rdf:resource="#Male" />
  </owl:Restriction>
</owl:Class>

<owl:Class rdf:ID="Father">
  <rdfs:subClassOf rdf:resource="#Man"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild">
        <owl:minCardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  
```

이처럼 온톨로지는 대상을 설명함으로써, 지식을 표현한다. 이 같은 설명을 위한 논리가 Description Logic이다 [1]. Description Logic은 프레임 로직을 기반으로 대상을 기술한다. 그 대상은 크게 두 가지로 나뉘는데, 술어 지식 (Terminological Knowledge Base : Tbox)과 정언 지식 (Assertional Knowledge Base : Abox)이 그것이다. 술어지식은 모델을 기술하기 위해 사용되는 일반 어휘들의 의미관계를 서술한 지식이다. 정언지식은 술어지식을 바탕으로 실제 대상을 할당한 것으로 주어진 환경에서의 술어 지식에 대한 해석이라고 볼 수 있다. 가령 가족은 아버지와 어머니, 그리고 자식으로 구성된다고 할 때, 가족, 아버지, 어머니, 자식이라는 일반 어휘가 술어지식이 되고, 자식인 철수, 아버지인 영수, 어머니인 영희인 철수네 가족에서 철수, 영수, 영희, 철수네 가족이 정언지식이 된다. 이상의 Description Logic은 그 어휘의 표현 정도에 따라 단계를 나눌 수 있는데, <표 2>는 Description Logic의 표현 범위에 따른 단계를 보여준다.

<표 2> Description Logic Family

| DL Family | expressive power | semantics |
|-----------|---|---|
| S | A T ⊥ ¬A C∩D ∇RC ∃R.T | atomic concept universal concept bottom concept atomic negation intersection value restriction limited existential quantification |
| | C | negation of arbitrary concepts |
| | R+ | transitive roles |
| U | C∪D | union |
| E | | full existential quantification |
| N | ≥nR, ≤nR | number restriction |
| H | | role hierarchy |
| I | R' | inverse role |
| Q | ≥nRC, ≤nRC | qualified number restriction |
| O | = 1 | nominal |

3. 제안된 언어 : EOL

Description Logic은 Tbox와 Abox를 사용하여, 환경에 대한 모델을 만든다. Tbox의 공유를 통해 상이한 시스템에서도 타 시스템의 지식 모델을 사용할 수 있고, FOPC 기반 지식보다 그 활용 및 계산에 있어 보다 효율적인 사용이 가능하다. 그러나, Description Logic의 사용에 있어서 주의해야 할 점이 존재하는데, 그 첫 번째 문제는 Description Logic 역시 그 표현범위가 증가함에 따라 추론 계산의 복잡도가 증가한다는 점이다. 추론 방법으로 주로 사용되는 Tableau-based Algorithm의 경우, SHIQ의 표현범위로 증가하면, 지수 복잡도(Exponential Time Complexity)로 증가된다 [9]. OWL의 경우 OWL-Lite의 경우, Sound & Completeness가 증명되었지만 OWL-DL은 아직도 완벽하게 지원되는 추론 엔진이 없고, OWL-FULL의 완벽 지원은 요원한 상황이다 [5]. 일반적으로 유비쿼터스 컴퓨팅 환경에서 기대되는 지능을 상황 인식과 그에 따른 지능화된 서비스 제공이라고 가정할 때, 센서를 통해 인지하는 정보, 그 공간에 존재하는 장치, 제공 가능한 서비스, 그리고 사용자의 프로파일 정보를 지식으로 활용할 수 있어야 한다 [10]. 이들 정보들은 서비스와 사용자들 사이의 계층 구조, 서비스가 발현되기 위해 필요한 컨텍스트 정보와의 관계, 서비스가 실현되기 위해 사용되는 장치들의 관계, 그리고 동일 상황에서 개별 사용자의 선호 서비스의 목록 등으로 구분될 수 있다.

각 서비스와 필요한 장치, 점검해야 하는 환경 정보와의 관계를 위해서는 existential quantification, value restriction, number restriction을 표현할 수 있어야 하고, 하나의 서비스 완료를 위한 플랜을 구성하기 위한 장치들의 병렬 사용을 기술하기 위해서 intersection, union을 표현할 수 있어야 한다. 또한, 대체 자원을 찾기 위해서 transitive role이 표현되어야 하고, 서비스 요

청 주체의 변화에 대처하기 위한 inverse role, 다수의 사용자에게 동시에 제공되는 서비스 기술을 위한 number restriction이 필요하다. 또 inverse role과 transitive role의 표현이 의미를 갖기 위해서는 role hierarchy 정보를 갖고 있어야 한다. 이상의 조건을 고려했을 때, SUNHI의 표현정보를 가지고 있다면, 주어진 공간에서 환경 모델을 구축하고, 센서 정보를 통해 상황을 인식하고, 그 상황에서 존재하는 사용자에게 적합한 서비스를 자동으로 선택하여 제공하기 위한 지식을 표현할 수 있다.

<표 3> EOL의 BNF Syntax

```

<concept> ::= <composite_concept> <restriction>+
            | <concept_name> <restriction>+
<composite_concept> ::= <intersection_of_concepts>
            | <union_of_concepts>
            | <negation_of_concept>
<restriction> ::= <universal_restriction>
            | <existential_restriction>
            | <at_most_restriction>
            | <at_least_restriction>

<universal_restriction> ::=
            (all <role_name>.<concept>)
<existential_restriction> ::= (some <role_name>)
<at_most_restriction> ::=
            (at_most <digit> <role_name>)
<at_least_restriction> ::=
            (at_least <digit> <role_name>)

<negation_of_concept> ::= (not <concept>)
<intersection_of_concepts> ::=
            (and <concept> <concept>)
<union_of_concepts> ::= (or <concept> <concept>)

<concept_name> ::=
            <uppercase_letter><letter_or_digit>+
<instance_name> ::=
            <uppercase_letter><letter_or_digit>+
<role_name> ::= <lowercase_letter><letter_or_digit>+
<condition> ::= transit | inverse | subrole
<Maximal_concept> ::= *TOP
<letter_or_digit> ::= <letter> | <digit>
<letter> ::= <lowercase_letter>|<uppercase_letter>
<lowercase_letter> ::= a|b|c|...|z
<uppercase_letter> ::= A|B|C|...|Z
<digit> ::= 0|1|2|3|...|9
    
```

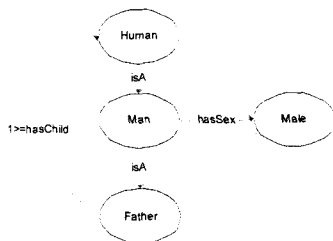
<표 3>은 SUNHI의 표현범위를 가진 EOL의 BNF 문법을 보여준다. 이상의 문법과 <표 4>의 명령어를 조합하여, 사용자는 지식을 표현, 추론 및 활용할 수 있다. EOL은 사용자 직관적인 명령어와 표현 문법을 제공함으로써, 추가적인 에디터의 도움 없이 지식을 표현할 수 있고, 실제 코드의 가독성을 높여줌으로써, 비전문가들도 쉽게 지식을 활용하여 지능적인 시스템 구축을 도와준다.

<표 4> Commands for defining knowledge

| Commands for defining concept | |
|----------------------------------|---|
| DefineConcept | (defineconcept <concept_name> <concept>) |
| Concept subsumption | (implies <concept_name1> <concept_name2>) |
| Concept Disjointness | (disjoint <concept_name1> <concept_name2>) |
| Commands for defining role | |
| Define Role | (defineroles <role_name> <concept1> <concept2>) |
| Role transitive | (transitiveroles <role_name>) |
| Inverse Role | (inverse <role_name> to <role_name>) |
| Role Hierarchy | (subrole <role_name> of <role_name>) |
| Commands for asserting instances | |
| Asserting Instance | (instance <concept> <instance_name>) |
| Commands for querying | |
| Unsatisfiability check | (!satisfiable <concept>) |
| Equivalence check | (!equivalent <concept1> <concept2>) |
| Subsumption check | (!ssubsume <concept1> <concept2>) |
| Disjointness check | (!disjoint <concept1> <concept2>) |
| Superconcepts retrieve | (superconcepts <concept>) |
| Role lists retrieve | (roles <concept>) |
| Instance retrieve | (instance <concept>) |
| Role relation retrieve | (role <condition> <role_name>) |
| ... | ... |

4. Test & Result

<표 1>의 OWL 예제를 EOL로 표현한 결과는 <표 5>와 같다. 단순히 코드상으로 비교했을 때 78%의 코드양의 감소를 확인할 수 있다. 또한 직관적인 지식 정의 명령어는 Protege등 에디터의 도움 없이 사용자가 직접 타이핑을 통한 지식 입력이 가능하고 실제 코드의 가독성을 높여준다. 지식 처리분야의 비전문가를 대상으로 <표 1>과 <표 5>의 내용을 바탕으로 <그림 2>의 트리플 구조를 유추할 수 있는가를 확인한 결과, EOL의 우월성을 확인할 수 있었다.



<그림 2> 온톨로지 예제

<표 5> EOL Example

```

(defineconcept Human)
(defineconcept Male)
(defineconcept Man (and Human (all hasSex.Male))
(defineconcept Father (and Man (at_least 1 hasChild))
(definerole hasChild.Human Father)
(definerole hasSex.*TOP* Human)
    
```

5. Conclusion

지능적인 시스템에 대한 요구는 컴퓨터의 발전 과정과 함께 증가하고 있다. 오늘날 유비쿼터스 컴퓨팅 환경의 대두와 함께, 지능형 시스템에 대한 기대가 한층 높아지고 있다. 지능적인 동작은 지식의 표현, 학습, 추론 그리고 응용 능력을 기반으로 하고 있다. 지식이 보다 세밀하게 표현될수록, 보다 지능적인 동작을 기대할 수 있지만, 동시에, 지식 관리를 위한 시스템 부하가 증가하기 마련이다. 유비쿼터스 컴퓨팅 환경에 존재하는 컴퓨팅 오브젝트들은 상대적으로 적은 컴퓨팅 능력을 갖고 있기 때문에, FOPC 같은 방법을 사용하거나, 고수준의 지식 표현 능력을 기대하기는 어렵다. 이에 본 논문에서는 SUNHI의 표현 능력으로 지식을 표현하기 위한 Frame기반 온톨로지 표현 언어인 EOL을 제안하였다. EOL은 직관적인 문법으로 비숙련자들도 쉽고 빠르게 지식을 표현하고 활용하기 위한 수단으로 활용될 수 있다.

6. 참고문헌

- [1] Franz Baader, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider. The Description Logic Handbook, Cambridge University Press, 2003
- [2] Volker Haarslev, Ralf Moller. Description of the RACER System and its Application, In IJCAR-01, volume 2083 of LNAI. Springer-Verlag, 2001
- [3] M. Lenzerini, D. Nardi, A. Schaerf. Reasoning in Description Logics, Principle of Knowledge Representation, Studies in Logic. CSLI Publication, 1996
- [4] Franz Baader, Ulrike Sattler. An Overview of Tableau Algorithms for Descriptoin Logics. Kluwer Academic Publishers, 2001
- [5] I. Horrocks. The FaCT system. In H. de Swart, editor, Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98, number 1397 in Lecture Notes in Artificial Intelligence, pages 307--312. Springer-Verlag, Berlin, May 1998.
- [6] Weiser, M., The Computer for the Twenty-First Century. Scientific American, Sept. 1991, pp. 94-110
- [7] Pim Borst, Hans Akkemanns, An Ontology Approach to Product Disassembly EKAW97 Sant Feliu de Guixols, Spain October 15-19th 1997
- [8] 조성원, 이건수, 송세현, 김민구 ALKETge : 유비쿼터스 환경을 위한 지식표현 언어, 2005 정보과학회 추계 학술대회
- [9] Horrocks, I. U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic SHIQ proc. of the 13th Conf. on Automated Deduction(CADE-17)
- [10] K Lee and M Kim "Conflict Resolution Method for Multi-Context Situation" PRIMA 2005 Malaysia pp.285-294