

유비쿼터스 지능공간을 위한 온톨로지 모델링과 추론

김연호[○], 양정진

가톨릭대학교

bluesky2603@catholic.ac.kr, jungjin@catholic.ac.kr

Ontology Modeling and Reasoning for Ubiquitous Smart Spaces

Yeon Ho Kim[○], Jung Jin Yang

Catholic University of Korea

요 약

유비쿼터스 지능공간은 개인이 원하는 서비스를 능동적이며 효율적으로 제공하는 공간이다. 그러한 환경을 제공하기 위해서 올바른 온톨로지의 모델링과 적절한 추론이 필요하다. 온톨로지 모델링의 어려움을 생각하며 그 방법을 모색한다. 또한 온톨로지 추론 엔진 중 RacerPro와 JESS의 특징을 이용하여 온톨로지의 보다 효율적인 활용방안을 찾는다.

1. 서론

유비쿼터스 컴퓨팅은 이제 더 이상 미래의 허황된 꿈이 아니다. 현대인은 적어도 한 개 이상의 컴퓨팅이 가능한 장비를 소지하게 되어, 유비쿼터스의 이념인 언제, 어디서나 서비스를 받을 수 있게 되었다. 유비쿼터스 지능공간은 최소한 개인의 신체정보 혹은 개인의 위치정보를 센싱할 수 있는 장비를 소지하고 있다는 것을 전제로, 독립적 개인에게 종속된 서비스를 제공할 수 있다.

유비쿼터스 지능공간이 더욱 의미를 갖기 위해서는 수동적인 서비스만을 제공하기보다 능동적인 서비스를 제공할 수 있어야 한다. 상황을 인식해서 그 상황에 맞는 서비스를 하기 위해서는 적절한 온톨로지의 모델링이 필수적이다.

온톨로지 모델링에 대한 검증된 방법론이 부족한 현시점에서 강조되는 것은 설계자의 공학자로서의 마음가짐이다. 변화에 대해 유연하게 대처할 수 있는 확장성과 실생활을 오류가 없이 표현하는 정확성을 항상 생각해야 하는 어려움을 갖고 있다. 이런 어려움을 극복하기 위해서는 많은 사람들이 공통으로 겪는 문제점을 발견해서 그 해결책을 제시하는 모델링 패턴이 필요하다.

온톨로지 모델링에 이어 능동적 서비스를 제공하기 위해서는 효율적인 추론과 그의 활용이 필요하다. 온톨로지 추론은 논리적 타당성과 결과 도출의 속도가 중요하다.

2장에서는 온톨로지와 그의 모델링에 대하여 논의와 모델링의 어려움을 극복하기 위한 방안, 및 그를 바탕으로 작성한 유비쿼터스 지능공간에 대한 모델링을 한 결과에 대해 이야기한다. 3장에서는 모델링을 한

온톨로지를 활용하기 위해 추론에 대해 이야기하면서 RacerPro 및 JESS 추론 엔진의 특징을 살펴보고 온톨로지를 어플리케이션에서 사용하기 위한 방법을 논한다.

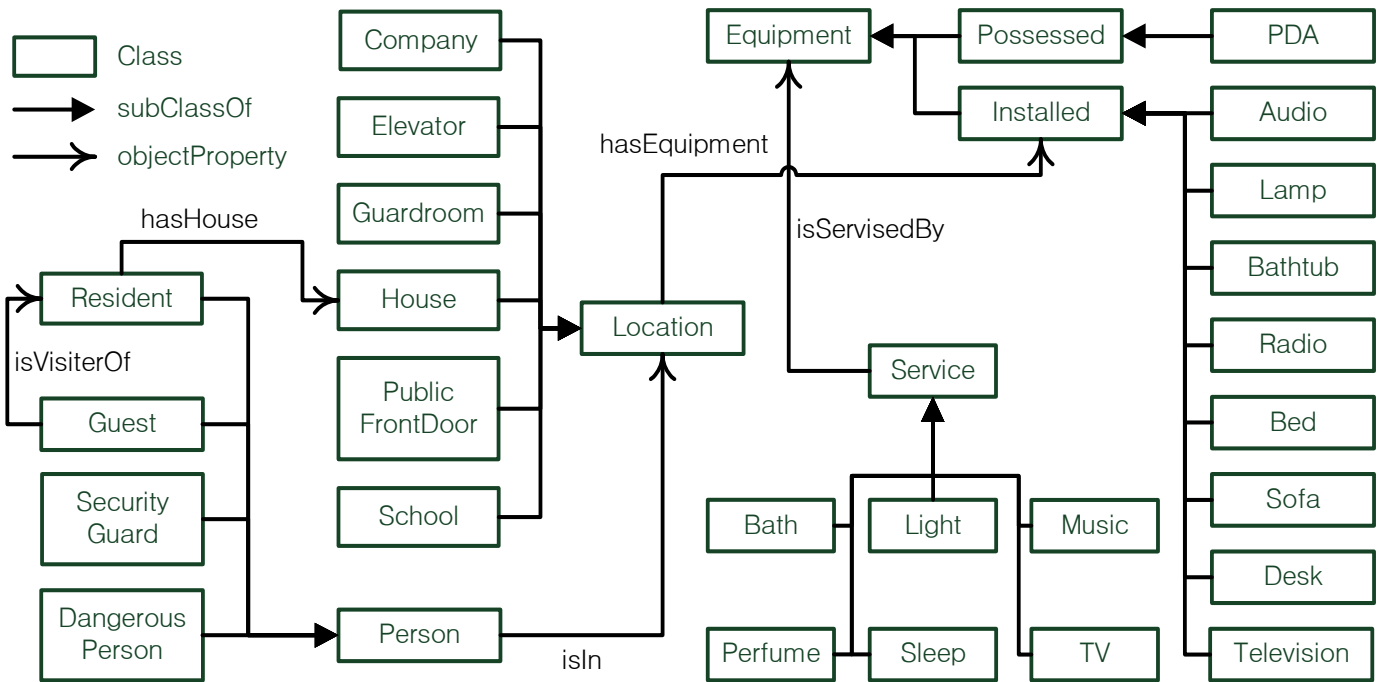
2. 온톨로지 모델링

온톨로지란 어휘나 개념의 정의 또는 명세로서 정보시스템 분야에서는 시스템이 다루는 내용에 해당하는 구성요소 즉, 개념을 의미한다. 온톨로지는 도메인 지식과 운영 정보를 분리하도록 하는데, 이것은 또한 지식의 공유, 논리적 추론 및 지식의 재사용을 가능하게 한다는 장점을 갖는다. [1]

그리고 온톨로지 모델링이란 특정 도메인에서 사용되는 어휘들과 그들간의 관계를 정의하는 것이다. 그러므로 올바른 모델링을 하기 위해서는 해당 도메인에 대한 전반적 이해와 함께 효율적인 온톨로지 모델링을 위한 방법론이 필요하다.

하지만 현재 온톨로지 모델링을 위한 검증된 방법이 정립되어 있지 않다. 단지 공학자로서 더 좋은 모델링을 위한 자세를 강조하고 있다. 이것은 소프트웨어 공학에서 객체 모델링을 할 때와 같이 풍부한 경험과 폭 넓은 지식을 필요로 하기 때문이다. 이런 문제점을 해결하기 위해 소프트웨어 공학에서와 같이 모델링 패턴을 찾으려는 노력이 많이 있다. 그 예로 W3C의 Semantic Web Best Practices and Deployment Working Group (SWBP) [2] 를 들 수 있다.

특히 SWBP에서는 표준 온톨로지 언어로 제안된 OWL [3]을 이용한 모델링 방법에 대해 소개하고 있다. 온톨로지 언어로는 Resource Description Framework Schema (RDFS) [4], DAML+OIL [5], 및 OWL이 있다.



[그림 1] 유비쿼터스 지능공간 온톨로지 모델링

이중 OWL은 RDFS와 같은 다른 언어보다 보다 표현력이 좋은 언어로써 현재 온톨로지 모델링에 넓게 사용되고 있다. OWL은 Resource Description Framework (RDF) [5] 를 기반으로 만들어진 언어이다. RDF는 웹 리소스에 관련된 정형화된 문장들을 기술하는 모델로서 객체를 웹 식별자로 인식하고 triple 형태인 property와 property value로 웹 리소스를 정의한다.

Tutorial at the Third International Semantic Web Conference (ISWC 2004)를 참조해서 모델링을 했다

[그림 1]은 유비쿼터스 지능공간의 모델링의 결과를 간략하게 보여주고 있다. 지능공간 모델링을 크게 사람(Person), 지역(Location), 장비(Equipment)에 중점을 두고 모델링을 하였다.

우선 지역(Location) 클래스는 설치된 장비(Installed Equipment)에 대한 정보를 갖고 있어, 해당 지역에서 제공할 수 있는 서비스의 정보를 제공한다. 또한 사람(Person)에 대한 정보를 갖고 있어서 개인화된 서비스를 제공할 수 있게 한다.

지역 클래스는 다시 계층적으로 회사(Company), 엘리베이터(Elevator), 경비실(Guardroom), 집(House), 공동현관(PublicFrontDoor) 그리고 학교(School)와 상속의 관계를 갖는다. 따라서 사용자의 정확한 위치에 대해 추론을 할 수 있으며, 해당 지역에서만 제공받을 수 있는 서비스를 위해 사용자에게 특정지역으로 이동을 추천할 수 있다.

사람(Person) 클래스의 계층적인 모델링 통하여

사람의 정보를 관리하고 이를 통해 지능공간에서의 안정성과 신뢰성을 추구한다. 특히 손님(Guest)은 반드시 방문할 주민(Resident)의 정보를 갖고 있어야 한다.

또한 장비(Equipment) 클래스는 설치된 장소 혹은 소유하고 있는 사람의 정보를 갖고 있으며, 하위 클래스들은 제공할 수 있는 서비스(Service)들의 정보를 갖고 있다. 서비스 클래스의 하위 클래스에는 목욕 추천 서비스(Bath), 조명 서비스(Light), 음악 서비스(Music), 향기 서비스(Perfume), 수면 추천 서비스(Sleep), 텔레비전 시청 서비스(TV)가 있다.

```

<Bed rdf:ID="Bed_1">
  <hasSleepService>
    <SleepService rdf:ID="sleep">
      <isSleepServedBy rdf:resource="#Bed_1"/>
    </SleepService>
  </hasSleepService>
  <hasPerfumeService rdf:resource="#lavenderPerfume"/>
  <hasPerfumeService rdf:resource="#chamomilePerfume"/>
  <hasPerfumeService rdf:resource="#applePerfume"/>
  <hasPerfumeService rdf:resource="#rosePerfume"/>
  <isInstalledIn rdf:resource="#privateRoom_1"/>
</Bed>
    
```

[그림 2] 부분적인 OWL 인스턴스 예

3. 추론

온톨로지는 TBox와 ABox로 구분이 된다. TBox는 온톨로지의 스키마를 포함하고 있으며, ABox는 인스턴스로 구성된다. 근래에는 온톨로지 규칙에 대한 추론의 필요성이 증대되어 SWRL(Semantic Web Rule Language) [6] 에 대한 추론에 대한 연구도 진행되고 있다.

TBox 추론이란 Terminological Box를 말하며, subsumption 관계를 추론하는 것을 의미하는데 이 과정에서는 클래스와 서브클래스의 관계를 추론하게 된다. Subsumption 추론은 하나의 클래스가 다른 클래스를 포함하는 것을 의미한다.

ABox 추론이란 Assertional Box를 말하며, 인스턴스 관계를 추론하는 기능으로 이 과정에서는 하나의 인스턴스가 정의된 클래스에 포함되는지를 검사한다.

Semantic Web Rule Language 즉, SWRL 추론이란 경험적 규칙을 SWRL로 작성하여 TBox와 ABox 관계를 추론하는 것을 의미하는데 이 경우는 규칙의 조건부에 정의한 TBox와 ABox 관계가 성립하는 경우 규칙의 결론을 유도하여 새로운 지식을 추론한다. [7]

3.1. RacerPro

RacerPro (Renamed ABox and Concept Expression Reasoner Professional)는 tableaux 알고리즘을 기반으로 만들어진 추론 엔진이다. Tableaux 알고리즘은 서술논리(Description Logic: DL)로 표현된 지식을 추론하기 위한 방법이다. Tableaux 알고리즘은 이미 알고 있는 지식을 기반으로 아직은 규칙에 만족하지 않는 요소에 대해 부정이 적용될 수 없음을 보여 그 요소가 규칙을 만족한다고 추론하는 방식이다.

그런 특징과 함께 RacerPro는 대용량의 TBox에 대한 추론을 효율적으로 수행할 수 있는 장점을 갖고 있다. 또한 ABox에 대한 추론도 지원하고 있으므로 온톨로지 모델링을 할 때 사용한다면 보다 양질의 온톨로지를 만들 수 있을 것으로 보인다.

가장 널리 알려진 온톨로지 저작도구인 Protégé는 RacerPro를 연동할 수 있는 방법을 제시하고 있어 온톨로지 모델링단계에서 해당 온톨로지의 타당성을 높일 수 있다.

3.2. JESS

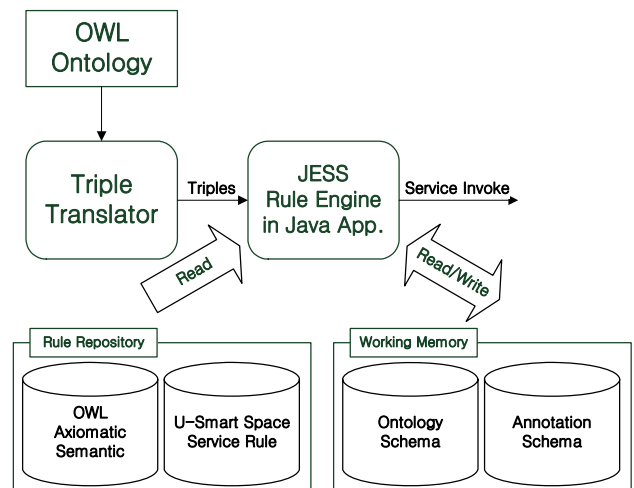
JESS (Java Expert System Shell)은 CLIPS를 확장하여 만들어진 rule 기반의 추론엔진이다. 여기에서 rule 기반의 추론엔진이 가지는 의미는 TBox나 ABox로 할 수 있는 추론의 범위를 넘어선 추론을 위함이다.

즉, 온톨로지의 클래스, Property, 인스턴스만을 위한 추론이 아닌 이러한 것을 활용해서 만든 rule을 이용하면 경험적 지식을 구현하여 더욱 실질적인 활용이 가능하게 된다.

또한 JESS는 그 이름에서 알 수 있듯이 Java 프로그래밍 언어로 구현이 되어있다. 이것은 JESS 상에서 컴파일을 하지 않고도 실행할 수 있는 어플리케이션의 구현이 가능하게 하고, 반대로 Java 어플리케이션에서 직접 JESS에서 정의한 fact와 rule을 사용할 수 있는 것을 의미한다.

3.3. OWL의 JESS 변환

온톨로지를 어플리케이션에서 사용하기 위해서는 두 가지 방안이 있다. 1) JESS shell에서 구현할 수 있는 Java 어플리케이션을 구현해서 사용하는 방법과 2) Java 어플리케이션을 구현해서 JESS를 로딩하여 사용하는 방법이다. 보다 유연하고 효과적인 시스템을 구현하기 위해서는 두 번째 방법을 사용하는 것이 효율적이다.



[그림 3] Rule 기반 추론 구조

온톨로지를 어플리케이션에서 활용하기 위해서는 온톨로지의 JESS 파일로의 변환이 필수적이다. Protégé를 이용해서 작성한 온톨로지는 확장자가 owl인 XML 포맷으로 저장된다. 특정 XML 포맷을 또 다른 형태의 XML 포맷으로의 변환은 XSLT (Extensible Stylesheet Language Transformations)를 이용해서 구현한다.

4. 결론 및 발전방향

온톨로지를 모델링하는 것에는 검증된 방법이 없는 것이 사실이다. 하지만 소프트웨어 객체를 모델링할 때와 같이 모델링 패턴을 이용한다면 보다 양질의

온톨로지를 모델링할 수 있을 것이다. 또한 RacerPro를 이용한다면 더욱 좋은 결과를 얻을 수 있다.

JESS는 Java로 작성된 오픈 소스 API이다. 앞에서 제시한 방법으로는 온톨로지와 rule을 직접 응용프로그램에 탑재해야 하는 제약사항이 있다. 하지만 온톨로지 repository를 작성하여 응용프로그램과 연동한다면 보다 높은 효율을 얻을 수 있을 것으로 예상된다. 또한 어플리케이션을 JADE를 활용하여 작성한다면 에이전트로서의 기능을 갖게 되어 더욱 자율적인 서비스를 가능하게 할 수 있을 것으로 보인다.

5. 후기

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반 기술개발사업의 08B3-S1-10M 과제로 지원된 것임.

6. 참고문헌

- [1] Eun Jung Ko, Ontology-Based Context Modeling and Reasoning for U-HealthCare, IEICE Trans Inf & Syst, VOL.E90-D.NO.8, 1262-1270, AUGUST 2007, <http://www.w3.org/2001/sw/BestPractices/>
- [2] W3C, OWL Web Ontology Language Overview Recommendation, 10 Feb. 2005, <http://www.w3.org/TR/owl-features/>
- [3] W3C, RDFS (RDF Vocabulary Description Language 1.0: RDF Schema) Recommendation, 10 Feb. 2004, <http://www.w3c.org/TR/rdf-schema/>
- [4] <http://www.daml.org>
- [5] <http://www.w3c.org/TR/rdf-primer/>
- [6] W3C, Ian Horrocks, Peter F. Patel-Schneider, Harold Boloy, Said Tabet, Benjamin Grosf, Mike Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004.
- [7] 최정화/박영택, 온톨로지 추론 기술 동향, 정보과학회지, 제24권 제12호, 47-55, 2006, 한국정보과학회