

# 유비쿼터스 환경에서 개인화된 서비스를 위한 온톨로지와 규칙 기반 자원 추론

강선희\*, 박종현\*\*, 홍성범\*, 김영국\*, 강지훈\*

\*충남대학교 컴퓨터공학과

\*\*충남대학교 소프트웨어연구소

e-mail : {shkang\*,jonghyunpark\*\*,sbhong\*,ykim\*,jhkang\*}@cnu.ac.kr

## Ontology and Rule-Based Resource Reasoning for a Personalized Service in Ubiquitous Environments

Sun-Hee Kang\*, Jong-Hyun Park\*\*, SungBum Hong\* , Young-Kuk Kim\*, Ji-Hoon Kang\*

\*Dept. of Computer Engineering, ChungNam National University

\*\* Software Research Center, ChungNam National University

### 요 약

유비쿼터스 컴퓨팅 환경에서는 공유 가능한 자원들이 산재되어 존재하며 사용자는 이를 기반으로 최적의 서비스를 제공받기를 원한다. 그러나 환경 내에는 다양한 상황과 서비스들이 존재하며 사용자 개인의 선호 정보 역시 매우 다양한 것이 현실이다. 그러므로 사용자가 원하는 서비스의 제공을 위해서는 사용자가 어떠한 상황에서 어떠한 서비스를 요청했으며, 어떤 자원이 사용자의 현재 상황에 적절한지를 판단하여, 사용자 요구사항에 맞는 자원을 추론하는 과정이 반드시 필요하다. 본 논문에서는 사용자가 최적의 서비스를 제공받을 수 있도록 주변의 공유 가능한 자원들을 추론하고 이들을 추천하기 위한 방법을 제안한다. 이를 위하여 사용자의 상황을 인식하기 위한 방안으로 온톨로지를 이용한 상황추론 방법을 제안한다. 또한 사용자 선호 정보를 반영하여 개인 맞춤형 자원을 추천하기 위한 추론방법의 하나로 규칙을 이용한 추론방법을 제안한다.

### 1. 서론

유비쿼터스 환경 내에는 수없이 많은 자원과 자원의 조합으로 이루어질 수 있는 서비스가 존재한다. 그리고 사용자는 컴퓨팅 환경이 발전함에 따라서 특별한 사용자 요청이나 개입이 없더라도 사용자의 상황을 인지하고, 그에 적절한 서비스를 받을 수 있기를 기대한다. 또한 유비쿼터스 환경 내에서 사용자의 상황과 각 서비스를 원하는 사용자의 요구사항 역시 다양하게 존재한다. 그러므로 상황, 자원, 사용자 정보 등을 고려한 지능화된 자율적 서비스를 수행하기 위해서는 우선, 자원과 서비스를 포함한 상황을 인지할 수 있어야 한다. 그러나 이렇게 상황을 일반화한다 할지라도 사용자 개개인의 선호 정보와 같은 동적인 상황이 존재하는 것이 현실이다 그러므로 인지된 상황들과 사용자의 선호도와 같은 개인화된 정보들을 이용한 추론을 통하여, 사용자의 기호에 맞도록 서비스에 최적화된 자원을 찾을 수 있어야 한다[1].

그러나 자원 및 서비스를 표현하는 방법은 매우 다양하며, 다양한 방법으로 표현된 정보를 추론에 이용하는 것은 많은 시간과 컴퓨팅 자원이 필요하다. 그

러므로 구성원에 의해 합의된 지식을 단일화된 표현 방식을 사용하여 표현하는 것이 필요하며, 대표적으로 온톨로지(Ontology)를 이용할 수 있다[2]. 본 논문에서는 온톨로지를 이용하여 유비쿼터스 환경 내에서 가능한 서비스와 자원을 정의하였으며, 이를 이용하여 온톨로지 추론을 한다.

한편 온톨로지는 합의된 지식을 표현하고, 일반화하는 것은 가능하지만 사용자들에게 공통적으로 적용되므로 개별적이며, 동적으로 변하는 정보를 반영하는 것에 한계가 있다. 그러나 개인화된 서비스 추론을 위해서는 사용자의 선호 정보, 사용 이력 등이 반영된 추론이 이루어져야 한다. 본 논문에서는 사용자 정보를 반영할 수 있는 다양한 추론 기법 중 규칙(Rule) 기반 추론을 통하여 사용자에게 개인화된 서비스를 제공할 수 있는 자원 추론을 보이고자 한다.

본 논문의 2 절에서는 유비쿼터스 환경에서 서비스를 제공하는 기존의 연구들과 본 논문에서 제안한 온톨로지와 규칙과의 차이점을 밝히고, 3 절에서는 우리 환경에 맞는 온톨로지를 정의 및 추론, 4 절에서는 규칙 기반의 추론에 대해서 설명하고, 5 절에서 결론을 기술한다.

### 2. 관련연구

최근 유비쿼터스 컴퓨팅 환경에서 상황인지를 통해

※ 본 연구는 21 세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크지원기반 기술사업의 08B3-01-30S 과제로 지원된 것임.

서비스를 제공하는 많은 연구들이 이루어지고 있으며, 이러한 연구들에서는 온톨로지기반 추론과 규칙기반 추론 방법이 사용되고 있다.

온톨로지를 이용하는 대표적인 연구들로는 CoBrA Project[3]와 CONON[4]과 같은 연구들이 있다. CoBrA Project 에서 온톨로지는 SOUPA[5]로 잘 알려져 있으며, 유비쿼터스 컴퓨팅 환경에서 지식 공유, 상황 추론, 상호운용성을 위한 표준 온톨로지로서 일반적이고 범용적인 온톨로지로서 정의되어 있다. CONON 연구는 유비쿼터스 환경에서 상황을 모델링하기 위해 온톨로지를 이용하고, 이를 추론하여 사용자의 상황 및 환경 내부의 상황 정보를 추론하고 있다. 그러나 위 연구에서 온톨로지는 가장 일반적인 상위 개념의 온톨로지로서, 상황을 일반화하는데 그 목적을 두고 있다. 그러므로 유비쿼터스 환경에서 제공 가능한 서비스와 자원을 체계적으로 분류하여 온톨로지를 정의하고, 추론에 이용하는 본 논문과는 차이를 보인다.

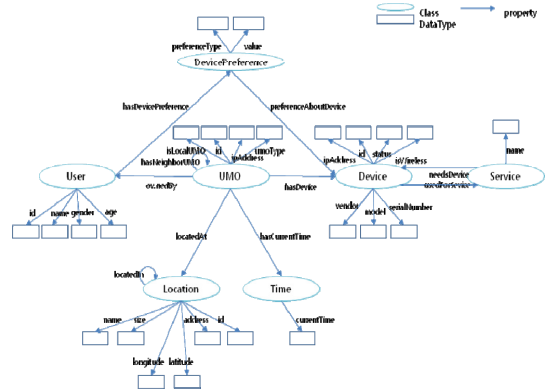
온톨로지기반 추론 함께 널리 쓰이는 추론방법의 하나인 규칙기반 추론을 이용하는 연구로는 DR-Prolog[6]와 CARE[7] 연구 등이 있다. DR-Prolog 연구에서는 nonmonotonic-rule 을 사용함으로써 불완전하고 일치하지 않는 정보들을 이용하는 추론을 하고 있다. 두 번째로 CARE 연구의 경우 온톨로지 추론을 이용해 추론된 상황을 보다 높은 수준의 상황 정보로 만들어 사용하기 위해 규칙 기반의 추론을 사용한다. 위의 두 연구에서 역시 규칙 기반 추론의 사용에 있어 사용자 정보를 이용하여 개인화된 추론을 하기 보다는 주어진 상황정보만을 이용하여 보다 고차원의 상황 정보를 얻는 추론을 수행하고 있다. 따라서 사용자 개인에 따라 개인화된 추론을 수행하는 본 논문에서의 연구와는 차이점을 보이고 있다.

### 3. 온톨로지의 정의 및 추론

#### 3.1 온톨로지

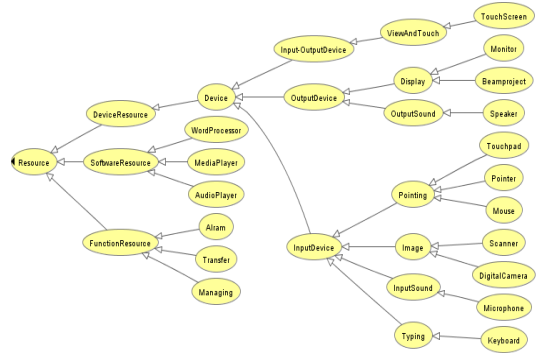
유비쿼터스 컴퓨팅 환경에서 다양한 정보를 표현하고 추론하기 위해서는 공통된 표현으로 기술하는 것이 중요하다. 이를 위해서 Web Ontology Language 인 OWL[8]을 사용하여 온톨로지를 정의하였으며, 개인화된 서비스(service)에 따른 자원 집합의 추론을 가능하도록 하는데 초점을 맞추어 정의되었다.

(그림 1)은 온톨로지기반 추론을 위해 필요한 전반적인 관계 및 속성들을 나타내고 있다. 그림과 같이 온톨로지 내에 정의된 프로퍼티(Property)들은 각 클래스(Class)들의 관계를 나타내고 있다. 이것은 향후 온톨로지기반 추론 과정에서 각 클래스들의 관계를 통해 사용자의 상황, 서비스에 따른 자원 집합을 추론할 수 있도록 하는 역할을 한다. 그러나 상황과 개인화된 서비스에 따른 자원을 추론 하기 위해서는 자원 정보에 대한 온톨로지가 체계적으로 정의되는 것이 무엇보다 중요하다.



(그림 1) Class-Property Diagram

본 논문에서는 자원 온톨로지를 유비쿼터스 컴퓨팅 환경 내에 존재하는 자원들을 체계적으로 분류하여 정형화하여 정의하였다. 특히 하드웨어 자원의 정보와 그 특징에 초점을 맞춰 하위 개념과 상위 개념을 정의하였다.



(그림 2) 자원 온톨로지

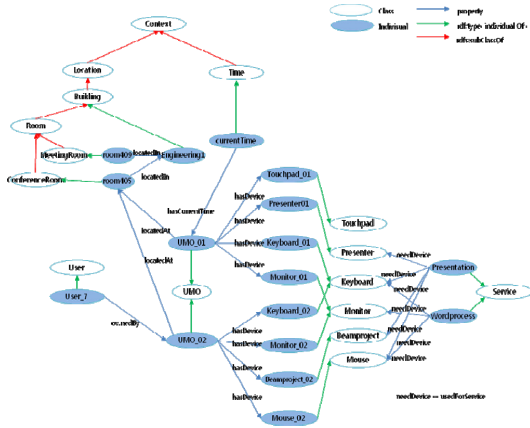
(그림 2)에서는 자원 온톨로지를 보여주고 있으며, 정의된 자원 온톨로지는 주변에서 자주 사용하는 하드웨어 자원들 위주로 구성되어 있다. 특히 상위로 갈수록 하위의 자원들의 공통적인 특징을 가진 개념으로 그룹화 함으로써, 일반적인 자원 온톨로지를 표현할 수 있게 되었다. 이는 추론을 통해 사용자에게 가장 적합한 자원을 제공할 때에도 구체적인 특징을 가지고 하드웨어 자원을 추론할 수 있도록 한다.

#### 3.2 온톨로지기반 추론

온톨로지로서 표현된 지식은 온톨로지 추론을 통해 하위 수준의 데이터가 상위 수준의 의미 있는 데이터를 만드는데 이용된다. 이때, 온톨로지 추론을 위해서는 온톨로지 내에 각 클래스들 사이의 관계가 정의되어 있어야 하며, 이러한 관계들을 파악함으로써 온톨로지 기반 추론이 이루어진다.

사용자에 의해 특정 서비스가 요청된 경우, 온톨로지 기반 추론에서는 우선 사용자의 상황을 추론해야 한다. 또한 상황 이외에도 현재 상황에서 사용자 요청에 적절한, 제공 가능한 자원들의 그룹에 대해서도 추론을 해야 한다.

(그림 3)은 정의한 온톨로지에 실제 값들이 들어 있을 때, 각 값들이 어떠한 관계를 가지게 되는지를 보여주는 예제 관계도 이다.



(그림 3) 온톨로지 추론을 위한 individual 예제

이러한 온톨로지 내의 관계들은 SPARQL[9] 을 이용하여 질의할 수 있으며, 이 과정을 통해 추론의 결과를 얻을 수 있게 된다. 위의 (그림 3)에서 보면 사용자가 “Wordprocess” 서비스를 제공받기 위해서는 {Keyboard, Monitor, Mouse}와 같은 자원이 필요하다. 이것은 이미 온톨로지에 정의된 Service 클래스와 Device 의 클래스의 관계인 “needDevice”에 의해 추론될 수 있다.

<표 1> “Wordprocess” 서비스를 위한 SPARQL 질의 및 결과 예제

<pre>&lt;SPARQL 질의&gt; SELECT ?Device WHERE { ?Service :needsDevice ?Device. ?Service :name " Wordprocess ". }</pre>
<pre>&lt;결과&gt; UMO_01 → Keyboard_01, Monitor_01, UMO_02 → Keyboard_02, Monitor_02, Mouse_02</pre>

<표 1>은 사용자 요청이 “Wordprocess”이고, (그림 3)에서와 같이 온톨로지에 실제 값들이 있을 경우의 SPARQL 질의와 질의의 결과를 보여준다. 그러나 사용자의 상황은 서비스만 있는 것이 아니므로, 온톨로지 기반 추론과정에서는 사용자의 현재 위치 정보와 시간 정보들도 추론한다. 하지만 온톨로지 기반 추론을 통해 나온 결과는 <표 1>에서와 같이 동일한 자원에 대해 여러 개의 결과가 나올 수 있다. 그러나 온

톨로지 기반 추론은 이러한 문제를 극복하고 사용자 개인에게 맞는 자원을 선택해 줄 수가 없다.

4. 규칙 기반 추론

상황을 고려한 추론은 사용자에게 서비스를 위한 자원을 추천할 때, 가장 먼저 고려되어야 하는 추론임에는 틀림없다. 그러나 그 상황에 따라서도 사용자 개인의 선호도는 모두 다를 수 있다. <표 2>에서는 상황에 따른 사용자의 모니터 선택 선호도에 대해 보여 주고 있다.

<표 2> 상황에 따른 모니터 선택 선호도 예제

상황정보		상황에 따른 사용자 선호도	
Wordprocess	실내	Vendor	
		Brightness	
		Size	
	실외	낮	Brightness
		밤	Size
		밤	Vendor

온톨로지 기반 추론을 통해 사용자의 현재 위치가 “실내” 혹은 “실외” 인지, 실외라면 “낮”인지 “밤”인지에 대한 상황 정보들을 얻을 수 있다. 그러나 서로 다른 두 사용자가 같이 “실외”, “밤”이라는 상황에 직면했을 경우 온톨로지 기반 추론을 통해서만 사용자가 그 상황에서 선호하는 자원을 제공 받을 수 없다. 그러나 규칙 기반 추론의 경우, 사용자 개인에게 맞는 맞춤형 규칙의 정의가 가능하고, 이를 추론함으로써 사용자에게 적절한 자원을 제공해 줄 수 있다.

만약 사용자가 “Wordprocess”를 요청한 상태에서 몇 시간에 건물 밖에 있는 상황이라 가정하자. 이럴 경우 <표 2>에서 보인 것처럼 사용자의 선호도는 Brightness, Size, Vendor의 순서를 가진다. 그러므로 사용자의 현재 상황에 맞는 선호도는 규칙을 통해 찾아져야 한다.

<표 3> 서비스, 시간, 장소에 따른 사용자 선호도 선택을 위한 JESS 규칙 예제

```
(defrule FindPreference
(SelectUserPreference (Service Wordprocess)
 (DayType Daytime) (Location Outside)
 (First ?first) (Second ?second) (Third ?third)
=>
(?*UserPreferenceRank* add ?first)
(?*UserPreferenceRank* add ?second)
(?*UserPreferenceRank* add ?third)
(store PreferenceRank ?*UserPreferenceRank*))
```

<표 3>은 사용자 선호도를 선택하기 위한 규칙기반 추론을 위해 JESS Rule[10]을 사용한 예제이다. 위와 같이 사용자의 선호도 정보가 있을 때, 규칙기반 추론에서는 사용자 선호도 정보를 상황정보와 조합하

여 현재 상황에 맞는 선호도를 알아낼 수 있다.

<표 3>의 규칙 추론을 통해 “낮” 시간에 “건물 밖”에서 사용자가 “Wordprocess” 서비스를 요청했을 경우의 선호도는 Brightness 임을 알아냈다. 그러므로 우리는 이 정보를 최종적으로 자원 추론에 이용함으로써 각 개인에게 최적의 자원을 추론할 수 있게 된다.

<표 4>에서는 이전의 규칙기반 추론 과정에서 얻어진 개인의 모니터 밝기에 대한 선호도 정보가 반영된 자원 추론을 위한 최종 추론의 예를 보이고 있다. 이렇게 최종 적용을 함으로써, 우리는 사용자에게 최적의 자원을 찾을 수 있다.

<표 4> 모니터의 Brightness 선호도에 따른 JESS 규칙 예제

```
(defrule PreferredMonitor
(Service Wordprocess) (Location Outside)
(Monitor (Location Outside) (ID ?id)
(Brightness ?brightness))
(Time (DayType Daytime))
(UserPreferenceAdaptation
{ ?brightness >= ?preferredBrightness &&
?brightness <= ?preferredBrightness + 100})
=>
(assert (RecommendedMonitor ?id))
```

5. 결론

본 논문에서는 유비쿼터스 컴퓨팅 환경에서 다양한 사용자의 요구 사항을 만족시킬 수 있도록 사용자의 상황뿐만 아니라 사용자의 개인적인 정보를 이용하여 추론하는 방법을 제안하였다. 이를 통하여 사용자의 현 상황과 동시에 사용자의 선호 정보 등이 반영된 자원 추론이 가능해 졌고, 이는 사용자 개개인에 개인화된 서비스를 제공하는 것이 가능하도록 하였다. 그러나 현재의 개인화된 자원추론은 하드웨어 자원을 그 대상으로 시험하였다. 그러므로 향후 하드웨어 자원 이외의 다른 자원들도 그 특성을 반영하여 개인화 서비스를 제공하기 위한 방법을 모색할 것이다.

참고문헌

[1] 최환수, 강선희, 이용대, 장서윤, 박원익, 박종현, 김영국, 강지훈, “유비쿼터스 환경에서의 상황 기반 디바이스 추론 시스템,” 제 28 회 한국정보처리학회 추계학술발표논문집, pp.903-906, 2007.

[2] 강선희, 박종현, 김영국, 강지훈, “유비쿼터스 지능 공간에서 자원 공유를 위한 온톨로지기반 추론,” 2008 한국콘텐츠학회 춘계종합학술대회, pp.489-493, 2008.

[3] "About ContextBroker Architecture", <http://cobra.umbc.edu/about.html>.

[4] X. H. Wang, D. Q. Zhang, T. Gu, H. K. Pung, "Ontology Based Context Modeling and Reasoning using OWL," PERCOMW'04, pp.18-22, 2004.

[5] H. Chen, F. Perich, T. Finin, A. Joshi, "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications," MobiQuitous'04, pp.258-267, 2004.

[6] Grigoris Antoniou, Antonis Bikakis, “DR-Prolog: A System for Defeasible Reasoning with Rules and Ontologies on the Semantic Web,” IEEE Transaction on Knowledge and Data Engineering, Vol. 19, No.2, pp. 233-245, Feb. 2007

[7] Alessandra Agostini, Claudio Bettini, Diniele Riboni, “A Performance Evaluation of Ontology-based Context Reasoning,” PERCOMW'07, pp. 3-8, 2007.

[8] Web Ontology Language (OWL), <http://www.w3c.org/2004/owl>.

[9] SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>.

[10] JESS, the Rule Engine for the JavaTM Platform, <http://herzverg.ca.sandia.gov/jess/>.