

온톨로지 기반 상황인지 모델링 연구: u-Convention을 중심으로

A Study of Ontology-based Context Modeling in the Area of u-Convention

김성혁(Sung-Hyuk Kim)*

초 록

유비쿼터스 컴퓨팅의 주요 기술인 상황인지는 환경을 구성하는 다양한 종류의 정보 기기로부터 전달되는 상황 정보를 이해하고 처리하며, 다양한 도메인에 유연하게 적용할 수 있는 상황인지 모델을 필요로 한다. 시맨틱 웹 기술 기반의 온톨로지는 구조화된 공통의 포맷을 이용하고 의미적인 정보의 표현이 가능하므로, 시스템이 상황 정보를 공유하고 이해, 추론함으로써 효과적인 상황인지가 가능하다. 따라서 온톨로지를 이용한 상황인지 모델이 여러 연구에서 제시되어 왔는데, 본 논문에서는 이러한 기존 연구들에 대한 분석을 바탕으로 상황인지 모델의 범용성과 확장성을 위해 온톨로지의 구조를 계층화하고 이를 기반으로 상황인지 시스템을 구현하여 실제 u-Convention 도메인에 적용하였다. 또한 OWL-DL의 기술논리와 SWRL 규칙 추론을 결합함으로써 복합적인 상황을 효과적으로 추론하는 방법을 제시하였다.

ABSTRACT

Context-awareness as a key technology of ubiquitous computing needs a context model that understands and processes situational information coming from diverse sensors and devices, and can be applied diversely in various domains. Semantic web based ontologies use structured standard format and express meaning of information, so it is possible to recognize effectively context-awareness situations, allowing the system to share information and understand situation by inference. In this paper, we propose a layered ontology model to support generality and scalability of the context-awareness system, and applied the model to u-Convention domain. In addition, we propose a effective reasoning method to handle compound situation by combining OWL-DL and SWRL rules.

키워드: 유비쿼터스 컴퓨팅, 온톨로지, 시맨틱 웹, 온톨로지 기반 상황인지, 상황인지 모델링
ubiquitous computing, ontology, semantic web, ontology-based
context-awareness, context modeling

* 숙명여자대학교 문헌정보학과 교수(ksh@sookmyung.ac.kr)

■ 논문접수일자: 2011년 5월 17일 ■ 최초심사일자: 2011년 5월 24일 ■ 게재확정일자: 2011년 8월 16일
■ 정보관리학회지, 28(3): 123-139, 2011. [http://dx.doi.org/10.3743/KOSIM.2011.28.3.123]

1. 서론

인터넷과 네트워크 그리고 정보처리기술의 발전은 점차 현실세계와 가상세계의 벽을 허무는 유비쿼터스 컴퓨팅(ubiquitous or pervasive computing), 더 나아가 사물 인터넷(internet of things)의 개념을 등장시켰다. 유비쿼터스 컴퓨팅 환경은 현실세계에 존재하는 사물들에 내재된 센서나 태그를 이용하여 사물을 실시간으로 감지할 수 있고, 네트워크화 되어 정보를 주고받음으로써 스스로 이용자에 적합한 지능적인 서비스를 제공하는 것을 목적으로 한다.

이러한 유비쿼터스 컴퓨팅 환경의 서비스를 제공하기 위한 주요 기술로서 상황인지(context-awareness) 기술이 필요하다(조위덕, 박세웅, 최재동 2006). 상황인지란 컴퓨팅 환경에 있어서 이용자와 그를 둘러싼 주변의 환경을 인식하고 이해함으로써 시스템이 변화하는 상황을 인지하고 개인적인 취향과 의도에 따라 환경을 제어하는 것을 의미한다. 인간은 자신의 전후 상황, 즉 컨텍스트를 가지고 있으며 동일한 사람일지라도 자신의 위치나 시간, 이전의 행위, 선호도 등에 따라 다른 판단을 할 수 있다. 따라서 좀 더 지능화된 서비스를 위해서는 동일한 입력에 대해 동일한 서비스를 제공하는 것이 아니라 이용자의 이러한 컨텍스트를 정보화하여 이에 적합한 서비스를 제공하는 것이 필요하다.

상황인지를 통해 이용자에게 적절한 서비스와 정보를 제공할 수 있는 컴퓨터 시스템이 바로 상황인지시스템(context-awareness system)이다. 이를 구현하기 위한 주요 연구과제로는 첫째, 어떻게 기계가 이해할 수 있도록 상황정보

를 표현하고 저장할 것인지에 대한 상황인지 모델링(context modeling), 둘째, 물리적인 환경으로부터 얻어진 정보를 기반으로 컨텍스트를 어떻게 이해할 것인지에 관한 컨텍스트 추론(context reasoning), 셋째, 독립적이고 분산되어 있는 정보기기(computing entity)들 간의 컨텍스트 지식(context knowledge) 공유를 위한 상호운용성으로 정의할 수 있다(Chen, Finin, and Joshi 2004a).

2000년대 이후 시맨틱 웹 기술(semantic web technology)이 등장하면서 상황인지 시스템을 구현하는데 있어 시맨틱 웹 기술을 이용하여 이러한 문제들을 해결하고자 하는 연구가 활발하게 진행되어 왔다. 시맨틱 웹 기술은 주창자인 팀 버너스 리(Berners-Lee, Hendler, and Lassila 2001)가 밝혔듯이 잘 정의된 의미를 정보에 부여함으로써 컴퓨터와 인간 간의 협력이 좀 더 잘 이루어질 수 있도록 하는 현재 웹의 확장이다. 현재의 웹이 인간의 이해를 위한 디스플레이 목적의 HTML을 이용하여 기계가 정보를 구문적 수준(syntax level)으로 처리하도록 표현하는 반면 시맨틱 웹은 RDF(Resource Description Framework)를 이용하여 정보의 의미를 표현함으로써 기계가 정보의 의미를 이해할 수 있을 뿐만 아니라 동시에 공통의 표준포맷을 사용함으로써 정보의 상호운용성을 의미적 수준(semantic level)으로 향상시켰다. 또한 RDF 기반의 OWL(Web Ontology Language)은 기술 논리(Description Logic)로 개념 간의 의미적 관계를 표현할 수 있는 도구를 제공함으로써 명시된 지식에 내재된 암묵적 지식을 추론을 통해 발굴하여 새로운 지식으로 만들어 낸다.

따라서 시맨틱 웹 언어(RDF, RDFS 그리고 OWL)의 풍부한 표현력으로 여러 상황 정보(contextual information) 즉 사람(people), 이벤트(event), 장치(device), 장소(place), 시간(time), 공간(space) 등을 표현하고 추론을 통해 컨텍스트화 함으로써 효과적으로 컴퓨터 시스템이 상황 정보를 이해하고 추론할 수 있도록 할 수 있다. 또한 시맨틱 웹 환경에서 공통의 포맷(format)과 어휘(vocabulary)를 이용한 상황 정보의 상호운용성은 다른 정보와 공유되고 융합됨으로써 새로운 지식과 서비스를 만들어 내는 가능성을 제공한다(Chen, Finin, and Joshi 2004a; 이승철, 김치수, 임재현 2007).

이러한 장점으로 인해 시맨틱 웹 기술을 컨텍스트 모델링 및 상황인지시스템에 적용하고자 하는 연구가 진행되어 왔다. 그러나 대부분의 연구에서 컨텍스트를 추론하는데 있어 규칙 기반의 처리가 중심이 되어 왔으며, 이로 인해 다양한 상황에 적용하기 어려운 점이 있었다.

본 논문에서는 시맨틱 웹 기술의 장점을 컨텍스트 모델링 적용하기 위해 이전의 온톨로지 기반 컨텍스트 모델들을 분석하고, 효과적인 컨텍스트 추론을 통한 상황인지 서비스를 도출하기 위해 기술 논리 기반의 OWL DL과 규칙 기반의 SWRL(Semantic Web Rule Language)을 결합한 모델링 방법을 제안한다. 또한 u-City 환경의 다양한 도메인(u-Shopping, u-Home, u-Convention 등)에 적용가능 하도록 범용성과 확장성을 지원하기 위한 상위 온톨로지(upper ontology)와 이를 확장한 도메인 온톨로지(domain ontology)로 나누는 상황인지 모델 구조를 설계하고, 이를 기반으로 상황인지시스템을 구현하고 실제 u-Convention 도메인에 적용하

였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 상황인지 모델링에 대한 기존 연구들의 분석을 통해 u-City에서의 상황인지를 위한 온톨로지 모델링 방안을 도출하고, 3장에서는 이러한 방안을 토대로 u-Convention 도메인을 중심으로 상황인지 모델을 제시한다. 4장에서는 상황인지 모델을 이용한 상황인지 시스템 구조 및 상황인식 방법, 상황처리 방법 등을 제시하고 마지막으로 5장에서는 본 연구의 결과와 앞으로의 연구방향을 제시하였다.

2. 연구 배경

2.1 u-City환경의 상황인지 시스템

첨단 정보통신 인프라는 유비쿼터스 환경의 지능적 서비스가 도시 공간에 융합된 미래형 첨단도시로서의 U-City가 가능하도록 하였다. u-City는 도시의 기본 인프라인 첨단 정보통신망을 기반으로 유비쿼터스 정보 서비스를 주체, 경제, 교통, 시설 등 도시의 다양한 구성요소에 융합함으로써 도시민의 생활편의 증대와 삶의 질을 높이고 도시 관리 및 제반 기능의 혁신을 통하여 도시 경쟁력 향상을 목표로 한다(황의관 2008). 따라서 u-City환경에서 상황인지시스템은 u-Health, u-Home, u-Traffic, u-Shopping, u-Convention 등 다양한 도시 환경에 적용 가능해야 하며, 동시에 첨단 정보통신 인프라를 통해 얻어진 물리적인 상황 정보와 이용자의 상황(위치, 시간, 디바이스 특징, 개인 정보 및 선호도, 주변 상황 등)을 고려한 지능화된 서비

스를 제공하여야 한다.

이에 따라 u-City 환경에서의 상황인지 시스템의 주요 기능을 다음과 같이 정의할 수 있다. 첫째, 다양한 센서 및 디바이스를 통해 물리적 환경으로부터 제공되는 상황정보(contextual information)를 수집하고 인식하는 기능이다. 둘째, 수집된 상황정보를 처리하여 컨텍스트를 생성함으로써 상황을 인지하는 기능이다. 셋째, 얻어진 컨텍스트와 이용자의 역할, 선호도, 일정, 인맥 등의 전반적인 개인 정보와 교통상황, 날씨 등의 외부 정보를 융합하여 사용자 중심의 지능화된 서비스를 제공하는 기능이다. 이러한 기능들을 구현하기 위해서 본 논문에서는 시맨틱 웹 기술인 OWL을 이용한 컨텍스트 모델을 구현함으로써, 상황 정보를 적절히 표현하고 추론을 통해 컨텍스트를 생성하며, 이용자 개인 정보와 외부 정보들을 컨텍스트와 융합하여 고도화되고 복합적인 컨텍스트 도출을 통해 이용자 중심의 지능화된 서비스를 제공하고자 하였다.

2.2 온톨로지 기반 상황인지 모델

시맨틱 웹 기술이 등장한 이후 이를 이용하여 보다 유연성 있고 다양한 도메인에 적용 가능한 상황인지시스템을 구현하기 위한 다양한 연구가 진행되어 왔다. 대표적인 연구로 CoBrA (Chen, Finin, and Joshi 2004a, 2004b), Gaia (Ranganathan and Campbell 2003), 그리고 SOCAM (Gu, Pung, and Zhang 2005) 등을 들 수 있다. CoBrA(Context Broker Architecture)

는 지능형 회의실 환경 등에서 사용자의 상황적인 필요성에 기반한 서비스를 제공하기 위한 브로커 중심의 구조로, 상황 브로커가 모든 장치와 서비스 에이전트에 의해 공유될 수 있는 CoBrA-ONT라는 컨텍스트 모델을 이용하여 센서로부터 획득된 상황 정보를 해석하고 추론한다. CoBrA-ONT는 지능형 회의실 환경에서의 컨텍스트를 모델링하기 위한 주요 요소로 물리적 장소(physical locations), 시간(time), 사람(people), 소프트웨어 에이전트, 모바일 기기(mobile devices), 그리고 회의 이벤트(meeting events)로 정의하였다(Chen et al. 2004b). 이후 회의실 환경이 아닌 다양한 유비쿼터스 환경에 적용하기 위해 좀 더 일반화된 온톨로지를 정의한 것이 바로 SOUPA(Standard Ontology for Ubiquitous and Pervasive Applications)이다. SOUPA 온톨로지의 특징은 이미 존재하는 다른 어휘(vocabularies)들, 즉 DAML-Time, FOAF,¹⁾ OpenCyc 등을 이용하여 시간, 사람, 장소 등을 표현함으로써 컨텍스트 온톨로지의 재사용성과 상호운용성을 높였다는 점이다 (Krummenacher and Strang 2007).

Gaia(Ranganathan and Campbell 2003)는 방, 집, 건물, 공항과 같은 물리적인 공간을 스마트 공간화(smart space)하기 위한 운영시스템의 기능(이벤트, 신호, 파일 시스템, 보안, 및 프로세스 등)을 제공한다. Gaia는 컨텍스트 개체로 어플리케이션(application), 이용자(user), 서비스(service), 디바이스(device), 장소(location), 행위(activity)를 정의하고, 이에 대한 표준 텍스트노미와 컨텍스트를 표현하는데 온톨로지 모

1) Friend Of A Friend. <<http://foaf-project.org>>.

텔을 이용함으로써 상황인지의 다양성과 복잡성을 해결하고자 하였다. 또한 다양한 상황 정보를 수집하여 룰 기반 혹은 기계학습에 의하여 에이전트가 다른 행동을 취할 수 있도록 하기 위해 컨텍스트 모델에 기반한 상황인지를 적용하였다. 그러나 W3C 표준인 OWL 대신 그 전신인 DAML+OIL을 이용하여 온톨로지를 정의했으며, 컨텍스트를 추론하기 위해 규칙(frist-order logic)기반의 상황정보 처리를 구현하였다.

SOCAM(Service Oriented Context-aware Middleware)는 유비쿼터스 환경에서 상황인지 서비스를 제공하기 위한 미들웨어로, OWL 기반의 온톨로지인 CONtext ONtology(CONON)를 사용함으로써 의미기반 컨텍스트에 대한 표현과 다양한 형태의 컨텍스트에 대한 추론, 지식의 공유, 컨텍스트의 분류와 상호 의존성과 관련된 것을 해소하고자 하는 것을 목적으로 한다(Gu, Pung, and Zhang 2005). CONON은 유비쿼터스 환경의 일반적인 개념, 즉 사람(person), 장소(location), 정보 기기(computing entity), 행위(activity) 등을 정의한 공통의 상위 온톨로지(upper ontology)와 다양한 하위 도메인에 적용되는 도메인 종속적인 온톨로지(domain-specific ontology)로 구분되는 온톨로지의 계층적인 구조를 제안함으로써 다양한 환경으로의 적용성을 높이고자 하였다(Wang, Zhang,

Gu, and Pung 2004a). 또한 기술 논리 기반의 추론과 이용자가 정의한 규칙(First-Order Logic) 기반의 상황 추론을 가능하도록 하였다(〈표 1〉 참조).

2.3 u-City 환경에서의 상황인지를 위한 온톨로지 모델링 방안

u-Health, u-Home, u-Traffic, u-Shopping, u-Convention 등 다양한 도시 환경에서 유비쿼터스 서비스를 지원하기 위해서는 다양한 도메인에 쉽게 적용 가능해야 한다. 이러한 조건은 컨텍스트 온톨로지가 일반성과 확장성을 동시에 보장해야 함을 가리킨다. Krummenacher와 Strang(2007) 등의 연구에 따르면 SOUPA, CONON, CoDaMoS 등의 온톨로지가 상위 온톨로지와 도메인 온톨로지 구조를 계층화함으로써 이러한 일반성과 확장성의 문제를 해결하고자 하였다. 예를 들어, CONON은 사람, 장소, 행위와 정보기기로 대표되는 주요 클래스를 상위 온톨로지에 정의하고 도메인 온톨로지(u-Home, u-Shopping, u-Convention 등)에서 상위 온톨로지를 구체화하여 사용하도록 함으로써 다양한 도메인 간의 상황 정보 공유가 용이하도록 하였다.

또 다른 컨텍스트 모델링의 조건은 기존 온톨로지를 재사용하고 확장하는 것을 기본 목적으

〈표 1〉 대표적인 상황인지 시스템의 온톨로지 비교

상황인지 시스템	온톨로지 모델명	온톨로지 언어	컨텍스트 추론
CoBrA	SOUPA	OWL	DL+규칙추론(F-OWL)
Gaia	Gaia	DAML+OIL	규칙추론(First Order Logic)
SOCAM	CONON	OWL	DL+규칙추론(이용자가 정의한 FOL)

로 해야 한다는 것이다(Krummenacher and Strang 2007). SOUPA의 예에서 보듯이 이미 정의되어 있는 FOAF나 Time 온톨로지를 사용함으로써 온톨로지를 모듈화하고, 쉽게 개발될 수 있도록 하며, 동시에 같은 어휘를 사용하는 외부 시스템과의 상호운용성을 높일 수 있는 장점이 있다.

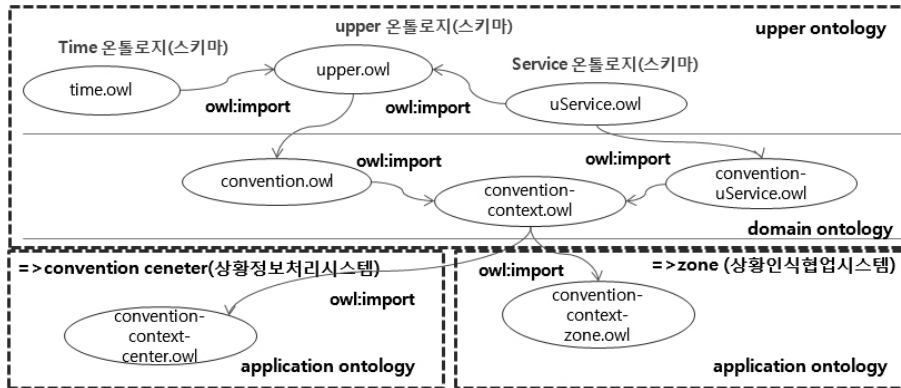
상황인지 시스템에서는 끊임없이 변화하는 상황 정보를 표현하고, 변화하는 환경에 동적으로 적응하기 위한 컨텍스트 추론 기능을 제공해야 한다. 이를 위해 컨텍스트 모델은 수집된 상황 정보를 인식하여 추론을 통해 컨텍스트를 생성하고 이에 적합한 서비스를 도출해야 한다. CONON과 SOUPA 온톨로지는 시맨틱 웹 표준인 OWL-DL을 기반으로 컨텍스트를 모델링하였다. OWL-DL은 FOL(first-order logic)의 하위 집합인 기술 논리(Description Logic)를 OWL의 표준화된 문법으로 표현한 것으로, FOL은 인공지능 분야에서 많이 연구되어 왔으나 결정 가능하지 않은(undecidable) 문제점이 있었다(박영택, 최중민 2006). 따라서 온톨로지를 모델링하는데 있어 결정가능성(decidability)이 보장 되면서도 W3C의 표준인 OWL-DL이 주목받게 되었다(Ye, Coyle, Dobson, and Nixon 2007). 기술논리(DL)의 강점은 상하관계의 추론을 통한 모델의 일관성 검사에 있다(Krummenacher and Strang 2007). 그러나 기술 논리만으로 다양하고 복잡한 컨텍스트를 표현하고 이에 따른 적합한 서비스를 추론해 내기에는 어려운 점이 있다. 따라서 기술 논리와 SWRL(Semantic Web Rule Language) 규칙을 결합하여 이러한 단점을 보완할 수 있다(김수경, 양기홍 2008).

3. 상황인지 모델링

3.1 u-City환경에서의 상황인지를 위한 온톨로지 설계

본 논문에서 다루는 온톨로지 모델은 u-City 환경에서 상황인지 서비스를 제공할 수 있는 컨텍스트 모델링을 위해 OWL을 사용하여 상황을 표현하였다. 다양한 형태의 온톨로지는 일반성(generality)의 정도에 따라 특정 도메인과 독립적인 일반적인 개념들을 기술하는 상위 온톨로지(upper ontology 또는 generic ontology), 특정 도메인에서 다루는 개념들을 기술하는 도메인 온톨로지(domain ontology) 그리고 특정한 어플리케이션에서 사용되는 개념들을 위한 어플리케이션 온톨로지(application ontology) 계층으로 구분할 수 있다(Ye, Coyle, Dobson, and Nixon 2007). 이러한 온톨로지 계층화는 온톨로지를 모듈화 하고 상위 온톨로지를 확장함으로써 다양한 도메인에 적용할 수 있는 장점이 있다. 따라서 본 연구에서는 u-City 내의 다양한 도메인에 적용할 수 있도록 하기 위해 상황인지시스템을 위한 온톨로지 구조를 크게 상위 온톨로지, 도메인 온톨로지 그리고 어플리케이션 온톨로지 계층화하여 설계하였다(〈그림 1〉 참조).

상위 온톨로지는 크게 상황 정보를 기술하는데 사용되는 사람, 장소, 시간, 장치, 상태 등을 포함하는 상위 클래스로서 ContextEntity와 ContextEntity로부터 인식되는 상황을 정의한 Context 클래스, 그리고 Context에 적합한 서비스를 정의한 Service 클래스로 구성된다. 이러한 기본적인 클래스들을 이용하여 모든 도메



〈그림 1〉 u-Convention 환경의 상황인지를 위한 온톨로지 구조

인에서 발생하는 모든 상황에 대해 표현이 가능하며, 따라서 상위 온톨로지를 다양한 도메인의 적용 및 확장이 용이하다. 이러한 온톨로지 계층화의 장점을 검증하기 위해 본 논문에서는 u-City의 다양한 도메인 중 u-Convention을 선택하고 위의 온톨로지 구조화 원칙에 따라 설계하였다.

또한 기존의 어휘를 활용함으로써 온톨로지의 재사용성과 상호운용성을 증진하기 위해, FOAF와 Time 온톨로지를 상위 온톨로지에 포함(import)하여 각기 사람(FOAF의 Person)과 시간(Time)을 표현하는데 이용하였다. 따라서 기존에 개발된 FOAF와 Time 온톨로지를 재사용함으로써 시간의 다양한 특성과 사람 및 기관, 그룹 등에 대한 온톨로지 설계에 드는 시간을 단축할 수 있었다.

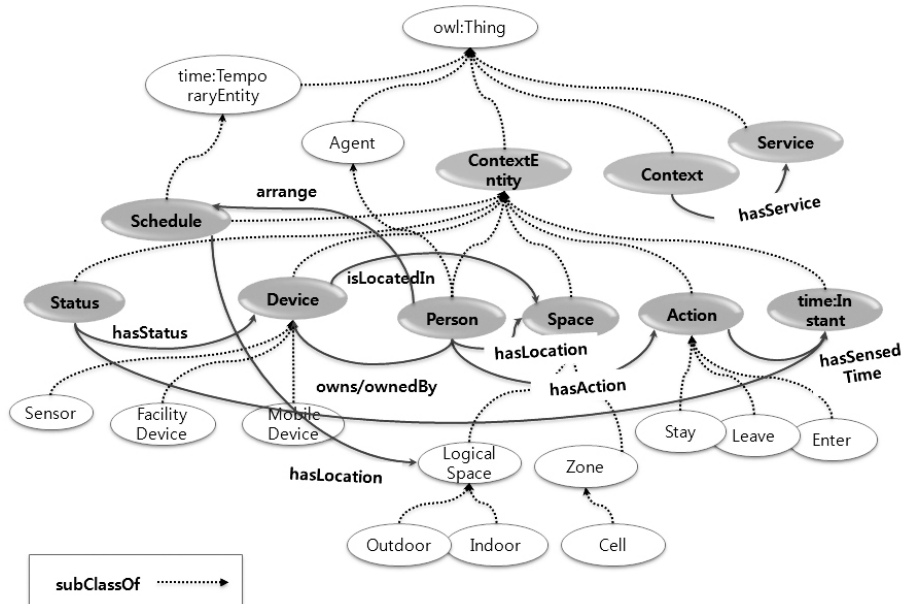
기존의 연구에서는 상황 정보(situational information)를 OWL을 이용하여 온톨로지의 인스턴스로 표현하고 SWRL이나 FOL과 같은 규칙 기반 언어를 사용하여 컨텍스트를 추론한다. 이와 달리 본 논문에서는 서술 논리(Description Logic) 기반의 OWL의 표현력을

이용하여 컨텍스트를 OWL로 명시적으로 정의한 후 인스턴스로 변환된 상황 정보를 추론하여 low-level의 컨텍스트로 도출하고 이를 기반으로 하여 결정 가능한 성질을 가지는 SWRL을 high-level 상황인식에 이용함으로써 효과적인 상황인식과 최적 서비스를 도출하기 위한 상황 모델링 방법을 제시하였다.

3.2 상위 온톨로지(upper ontology)

상위 온톨로지(upper ontology)는 직접적으로 상황을 구성하는 요소들을 정의한 Context Entity 클래스와 ContextEntity를 표현하는데 사용되는 부가적인 클래스들로 구성되어 있다 (〈그림 2〉 참조).

ContextEntity는 하위 클래스로 사람(Person), 장소(Space), 시간(Time), 일정(Schedule), 장치(Device), 상태(Status), 행위(Action)를 가진다. u-Convention 환경에서 이용자는 Person으로 표현되며 실제 이용자의 행위는 Device의 하위클래스인 MobileDevice(모바일 기기)를 통해 인식된다. 따라서 Person은 컨벤션의



〈그림 2〉 상위 온톨로지 모델

특정 공간에서 출입 및 존재 여부를 표현하는 Action 클래스와 hasAction 프로퍼티를 통해 연결된다.

Space는 사용자(Person)가 현재 위치하고 있는 장소를 기술하며 여기에는 하나의 서비스 범위로 규정된 영역, 즉 물리적 장소인 Zone과 이에 대응되는 논리적인 장소, 다시 말해 장소의 명칭인 컨퍼런스 룸101, 로비, 전시장 등인 LogicalSpace로 구분되며, Zone은 locationName 프로퍼티를 통해 LogicalSpace를 가진다.

Schedule은 시간과 장소를 갖는 예정되어 있는 일정을 기술하기 위한 클래스로, 이러한 정의는 다음과 같이 OWL에서 표현된다.

Schedule

```

a owl:Class ;
rdfs:label "Schedule"^^xsd:string ;
rdfs:subClassOf:ContextEntity ;
owl:equivalentClass
[ a owl:Class ;
owl:intersectionOf([ a owl:Restriction ;
owl:onProperty:hasLocation ;
owl:someValuesFrom:LogicalSpace
] time:Interval)
] .
    
```

Device는 상황인지를 위한 특정 영역의 센서와 고정형 디바이스로부터 생성되는 상황 정보(context information)를 기술하기 위해 사용되며, 각각의 Device에 대한 상태는 Status 클래스에 기술된다.

3.3 도메인 온톨로지(domain ontology)

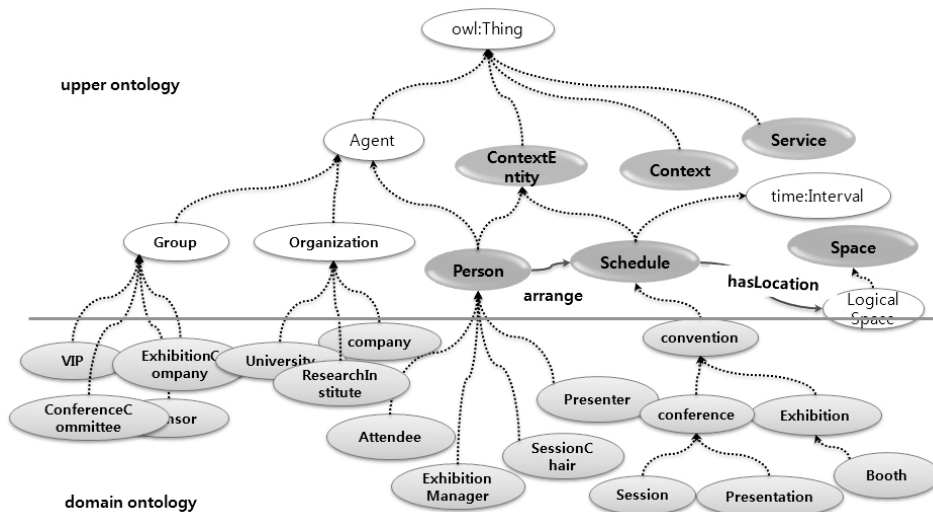
도메인 온톨로지는 상위 온톨로지를 임포트하여 도메인의 환경에 맞게 확장, 정의하게 된다(〈그림 3〉 참조). 즉 상위 온톨로지에 정의된 ContextEntity를 구성하는 Person, Device, Status, Schedule 등의 클래스 및 Context와 이에 관련된 Service 클래스의 하위 클래스를 u-Convention 도메인에서 상황인지를 위해서 확장, 정의한다. 예를 들어, 상위 온톨로지에서 Person 클래스가 프로퍼티 arrange의 값으로 Schedule을 가지도록 정의되었는데, u-Convention 도메인에서 Person 클래스는 컨벤션에 참가하는 역할에 따라 Presenter, SessionChair, Attendee, Exhibiter 등으로 하위 클래스가 정의되고 Schedule 클래스는 Conference, Exhibition, 그리고 다시 Conference 클래스는 Session과 Presentation 클래스로 구체화되어 정의될 수 있다(〈그림 4〉 참조). 여기에서 Presenter는 Person

의 하위클래스이며 동시에 givePresentation 프로퍼티의 값으로 Schedule 클래스의 하위 클래스로 정의된 Presentation 클래스의 인스턴스를 갖는 제한 클래스로 다음과 같이 정의되었다.

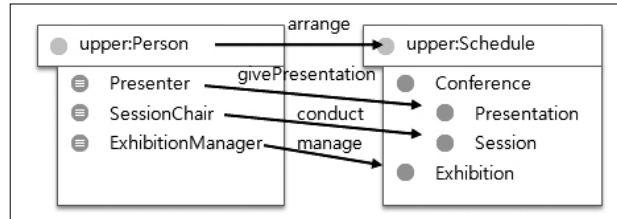
```

:Presenter
a owl:Class ;
rdfs:label "Presenter"^^xsd:string ;
rdfs:subClassOf upper:Person ;
owl:equivalentClass
[ a owl:Restriction ;
owl:onProperty:givePresentation ;
owl:someValuesFrom:Presentation
] .
    
```

따라서 Person 클래스의 인스턴스인 P1이 givePresentation 프로퍼티 값으로 Schedule 클래스의 하위 클래스인 Presentation의 인스턴스 Presentation_1을 가진다면 P1은 추론의



〈그림 3〉 u-Convention 도메인 온톨로지 모델



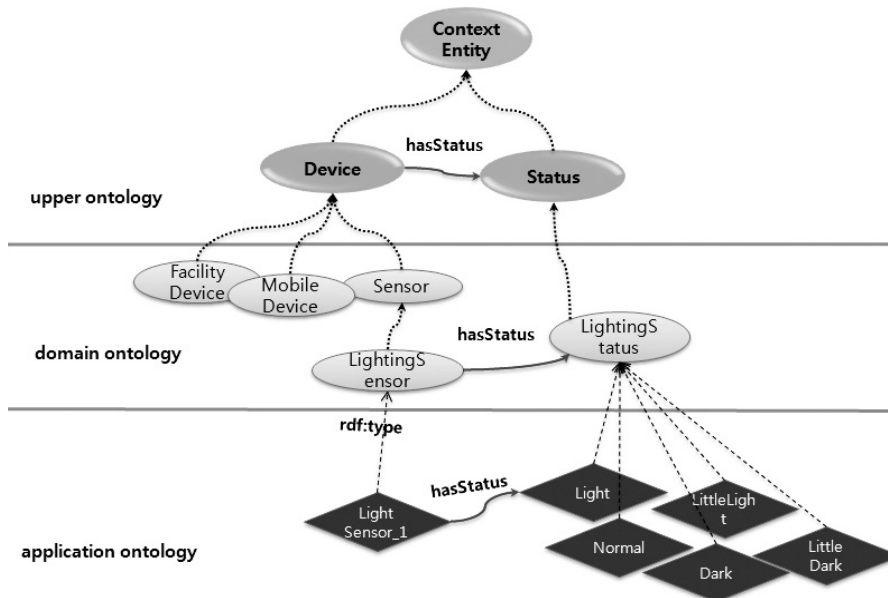
〈그림 4〉 u-Convention 도메인에서 Person과 Schedule의 관계

결과로 Presenter의 인스턴스로 분류되어 발표자로 자동 인지된다.

3.4 어플리케이션 온톨로지(application ontology)

본 논문에서는 시스템이 특정 영역(zone)에 적용되어 운용되는 상황인지시스템을 상황인식협업시스템으로 명명하였다. 상황인식협업시스템에서 사용되는 컨텍스트 모델은 도메인 온톨

로지를 확장하여 시스템이 적용된 물리적 환경에 존재하는 센서(Sensor), 디바이스(Device), 이용자(Person), 일정(Schedule) 등의 개체(instance)들과 이를 통해 인식될 컨텍스트(Context) 및 컨텍스트를 통해 제공될 서비스(Service)들의 개체(instance)들을 정의한다. 예를 들어, 컨벤션의 특정 컨퍼런스 룸에 적용된 상황인식협업시스템의 LightSensor_1이라는 조도 센서의 상태는 어플리케이션 온톨로지에서 〈그림 5〉와 같이 표현된다.



〈그림 5〉 상황인식협업시스템의 어플리케이션 온톨로지

4. 온톨로지를 이용한 상황인지

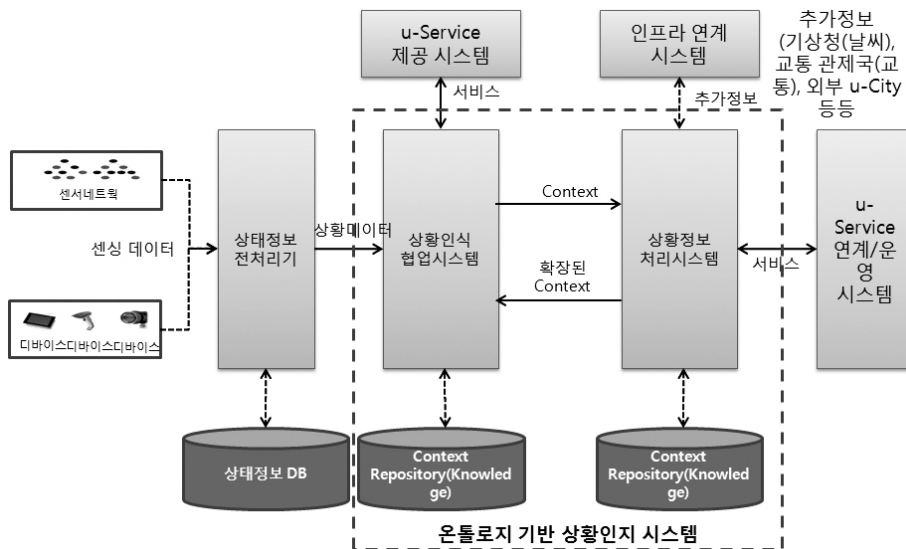
4.1 온톨로지 기반 상황인지시스템의 구조

u-City 환경에서 상황인지시스템은 특정 공간(zone)에 한정된 상황인식뿐만 아니라 이용자의 위치 이동으로 발생하는 연속된 상황들을 끊임없이(seamless) 인지하고 이용자 정보(선호도, 인맥, 관심 주제 등)뿐만 아니라 외부의 정보(날씨, 교통 정보 등)들을 종합하여 상황인지에 반영하는 복합적 상황 인식이 필요하다. 이러한 요구에 대응하기 위해 본 논문의 상황인지시스템은 특정 공간(zone) 내의 상황인식 및 이에 대응하는 서비스 도출을 담당하는 상황인식협업시스템과 공간을 담당하는 각 상황인식협업시스템의 상위 시스템으로서의 상황정보처리시스템으로 구분하였다. <그림 6>에 따

르면 상황인식시스템은 각 공간 내에서 발생하는 상황데이터를 상황인지 모델에 인스턴스로 생성하고 추론을 통해 컨텍스트를 도출함으로써 상황을 인식하여 이에 적합한 서비스를 이용자에게 제공하거나 공간 내에 적합한 서비스가 없는 경우 상황정보처리로 컨텍스트를 전달한다. 상황정보처리시스템은 이용자를 중심으로 상황을 인식하고 공간과 공간 간의 연속된 이용자의 컨텍스트를 관리하며, 또한 상황 모델에 저장되어 있는 지식 및 외부 인프라 시스템을 통한 정보의 확장을 통해 상황을 융합하고 확장함으로써 이용자에게 최적의 서비스를 도출하는 것을 목적으로 한다.

4.2 상황 데이터의 인스턴스 생성

상황을 표현하는 데이터는 이용자의 상태를 기술하는 Action 데이터, 고정 디바이스나 센서



<그림 6> 온톨로지 기반 상황인지시스템 구조

로부터 수집되는 상태를 기술하는 Sensed 데이터, 그리고 시스템 내부에서 발생하는 Event 데이터로 나뉘어지며, 상황 데이터는 상황인지 모델의 ContextEntity 클래스의 각 해당하는 하위 클래스에 인스턴스로 생성됨으로써 상황 데이터를 의미적으로 표현한다.

Action 데이터는 이용자가 zone으로 구분된 특정 영역으로 입장(Enter)하거나, 퇴장(Leave)하는 등의 이용자의 상태를 전달하며, 이것은 다음과 같은 인스턴스로 변환되어 상황인지 모델에 저장된다. 예를 들어, P1이라는 이용자가 ConferenceRoom_1에 들어왔음을 알리는 Action 데이터는 <그림 7>과 같이 상황인지시스템에 전달되고, 인스턴스로 변환되어 컨텍스트 모델에 생성된다.

4.3 컨텍스트 생성

상황인지 모델의 인스턴스로 생성된 상황 정보는 추론을 통해 컨텍스트를 생성하게 된다. 본 논문에서는 컨텍스트 생성에 OWL DL을 이용한 기술논리(DL) 추론과 SWRL을 이용한 규칙 추론을 결합함으로써 각 추론 방식의 단점

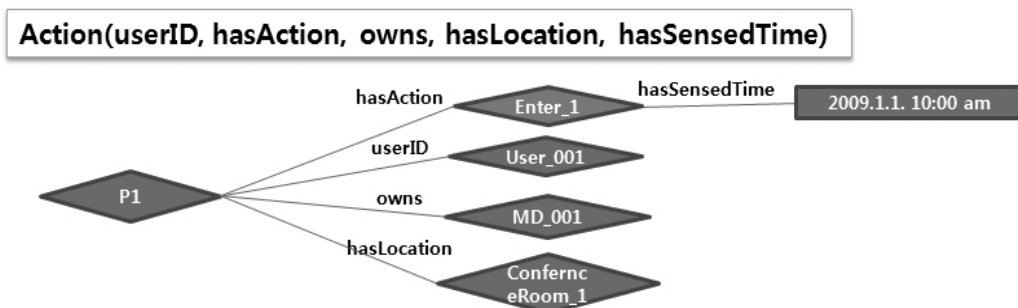
을 보완하는 효과적인 추론방식을 채택하였다.

4.3.1 low-level 컨텍스트 생성

OWL로 표현된 온톨로지는 서술 논리(Description Logic)를 기반으로 하고 있으며 서술 논리는 사람이 가지고 있는 지식을 표현할 때 지식의 용어적 부분(TBox)과 선언적 부분(ABox)로 나누어 표현한다. TBox 추론은 온톨로지 스키마에 정의된 클래스와 속성을 통해 표현된 지식에 대해 클래스 간에 형성된 암시적인 포함관계(subsumption)를 명시적으로 추론하고, ABox 추론은 인스턴스 관계를 추론하는 기능으로 인스턴스가 어떤 클래스에 포함되는지 명시적으로 추론한다.

본 상황인지시스템에서는 센서나 디바이스를 통해 수집된 상황 데이터를 추론을 통해 해당 컨텍스트의 인스턴스로 분류하는 일차적 추론 과정을 low-level 컨텍스트로 명명하였다.

이러한 low-level 컨텍스트 도출 과정을 예를 들어 설명하면 다음과 같이 상위 온톨로지의 클래스 Context의 하위 클래스로서 low-level 컨텍스트를 정의한다.



<그림 7> Action 데이터의 인스턴스 변환

low-level 컨텍스트 선언의 예:

```

UserEnterConferenceRoom ≡ Person ⊔
⊔ owns.MobileDevice ⊔
⊔ hasAction.Enter ⊔
⊔ hasLocation.ConferenceRoom
PresenterInPlatform ≡ Person ⊔
⊔ givePresentation.BeforePresentation ⊔
⊔ hasAction.Enter ⊔ ⊔ owns.MobileDevice ⊔
⊔ hasLocation.Platform
    
```

앞 4.2절의 Action 데이터의 예에서 P1이라는 이용자가 ConferenceRoom_1에 들어왔을 때 위의 컨텍스트 선언을 이용한 상황 데이터의 인스턴스 선언(assertion)과 그에 따른 컨텍스트 추론(Abox), 즉 인스턴스가 해당 컨텍스트로 분류된 결과는 다음과 같다.

```

[인스턴스]
Person(P1)
userID(P1, user001)
Action(Enter_1)
has Action(P1, Enter_1)
Mobile_Device(MD_001)
owns(P1, MD_001)
hasLocation(P1, ConferenceRoom_1)
hasSensedTime(P1, 2009.1.1 10:00 am)
    
```

```

[추론결과]
UserEnterConferenceRoom(P1)
    
```

이러한 상황 표현은 조합되어 더욱 복잡한 상황을 표현할 수 있다. 위의 예에서 PresenterInPlatform 컨텍스트는 타입이 Person인 어떤

사람이 givePresentation의 값으로 어떤 Presentation을 가지며, Mobile Device를 소유하고 있고, 지금 발표단상(Platform)에 입장(Enter)했음이 인지된 상황을 표현한다. 여기에서 발표단상(Platform)에 입장(Enter)한 이 사람(Person)이 발표하기로 예정된 이 프리젠테이션은 BeforePresentation으로 정의된 컨텍스트 하에 있는 프리젠테이션임을 나타내며, BeforePresentation 컨텍스트는 PresenterInPlatform 컨텍스트를 더욱 구체적이고 정확하게 표현하기 위해 사용된다.

4.3.2 high-level 컨텍스트 생성

앞에서 언급한 low-level 컨텍스트는 센서나 디바이스로부터 수집된 상황 데이터에 암묵적으로 내재된 상황을 컨텍스트로 명시화하는 데 유용하다. 그러나 OWL은 규칙의 표현을 배제하는 단점으로 인해 복잡한 상황의 표현과 이에 따른 서비스 도출에 제약이 있을 수밖에 없다. 따라서 본 논문에서는 복잡한 상황, 즉 상황의 확장을 통한 다양한 상황의 융복합을 정의하고 표현하기 위한 high-level 컨텍스트를 정의하는 데 SWRL(Semantic Web Rule Language)을 사용하였다.

SWRL 추론은 경험적 규칙을 SWRL로 작성하여 TBox와 ABox 관계를 추론하며, 규칙의 조건부에 정의한 TBox와 ABox 관계가 성립하는 경우, 규칙의 결론을 유도하여 새로운 지식을 추론한다. 예를 들어, 앞 절에서 low-level 컨텍스트로 정의된 PresenterInPlatform이라는 상황은 발표자가 단상에 들어옴이 인지되면 발표자가 미리 설정한 발표 시의 조도(illuminance)에 대한 선호도 값으로 발표장의 조도를 맞추

는 서비스를 제공해야 한다. 이를 위해 Presenter InPlatform 컨텍스트로 분류된(subsume) 발표자 P1의 조도에 대한 선호도 값과 센서를 통해 수집된 발표장의 조도상태 값을 비교함으로써 두 값이 같지 않다면 발표자에 대한 조도 맞춤 서비스가 필요하다는 컨텍스트를 도출하게 된다. 위의 내용을 SWRL로 표현하면 다음과 같다.

```

convention:LightingSensor(?lightSensor)
(?lightSensor upper:hasStatus ?lightStatus)
(?lightSensor upper:locatedIn ?loc_A)
PresenterInPlatform(?user)
(?user upper:hasLocation ?platform)
(?platform upper:isPartOf ?loc_B)
Equal(?loc_A, ?loc_B)
(?user convention:lightingPreference ?userlightPref)
notEqual(?lightStatus, ?userlightPref)
-> RequestLightingAdjustment(?user)
    
```

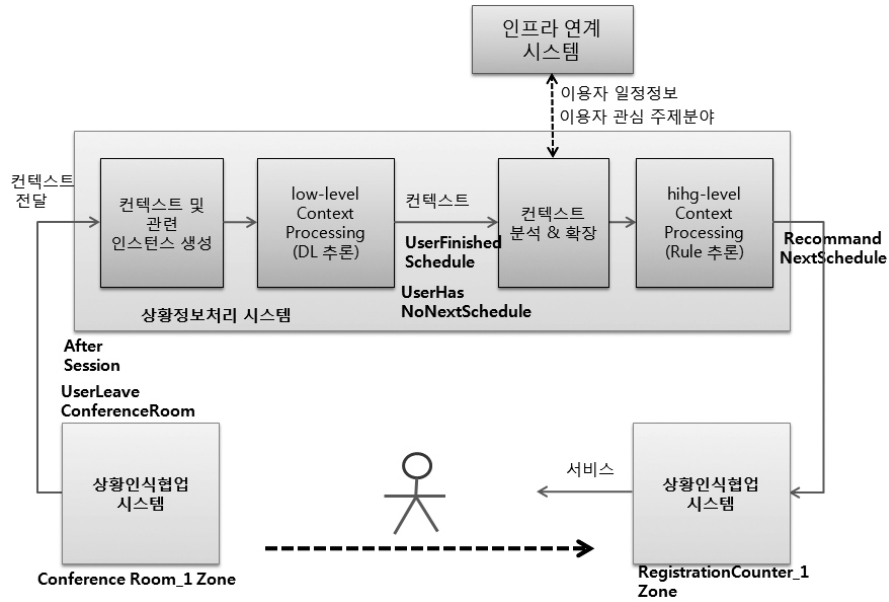
즉, PresenterInPlatform이라는 컨텍스트를 기반으로 PresenterInPlatform 컨텍스트가 발생한 장소(Space)에 존재하는 조도 센서(Lighting Sensor)의 상태 정보와 이용자(Presenter)의 조도에 대한 선호도 정보가 융합되어 Request LightingAdjustment라는 high-level 컨텍스트가 도출되었다.

4.4 상황인지 시스템에서의 복합적인 상황처리

u-Convention 도메인에서 발표장(conference room)은 하나의 zone으로 구성되며, 발표장 zone

을 담당하는 상황인식협업시스템은 발표장 내에서 발생하는 상황의 인식과 이에 따른 서비스들을 담당한다. 또한 u-Convention 도메인에는 발표장 외에도 등록대(Registration Counter), 전시장(Exhibition Room)이라는 공간이 존재할 수 있으며, 이용자는 현재의 발표장에서 일정이 종료되어 떠난 후에 이동하는 동안 발표장과 등록대를 연계한 복합적 상황인식과 이에 대한 서비스를 받을 수 있다. 예를 들어, 발표장 ConferenceRoom_1에서 발표를 마친 이용자는 세션 종료 후(AfterSession 컨텍스트) 발표장을 떠난다(UserLeaveConferenceRoom 컨텍스트). 발표장의 상황인식협업시스템은 컨텍스트를 도출하지만 이용자는 이미 떠났으므로 이에 대한 서비스는 발생하지 않는다. 따라서 상황인식협업시스템은 이 컨텍스트를 상황정보처리시스템에 전달한다. 상황정보처리시스템은 컨텍스트를 받아 컨텍스트 분석을 통해 이 이용자가 발표를 끝냈으며(UserFinishedSchedule), 그 이후의 일정이 아직 없다는 것을 인식한다(UserHasNoNextSchedule). 이에 대해 상황정보처리시스템은 상황 모델에 저장되어 있는 이용자의 관심 분야(Person 클래스에 interest 프로퍼티 값으로 연결되어 있는)를 분석하고, 이에 매칭 되는 이후 스케줄(예를 들어, 전시회의 관심 분야에 해당되는 부스들)을 이용자에게 추천하는 서비스를 제공한다(RecommendNextSchedule). 이 서비스는 상황처리시스템이 이용자의 위치가 등록대에 있음을 인식하여 등록대 zone에 있는 상황인식협업시스템을 통해 이용자에게 전달된다(<그림 8> 참조).

이러한 상황 분석 및 확장을 통한 복합적인 상황인식은 zone과 zone 간의 연계를 담당하는



〈그림 8〉 복합적인 상황처리 과정

상황정보처리시스템에서 이루어지며, 이용자의 위치 센싱 및 zone 내에서의 상황 인식과 이에 따른 하위 컨텍스트의 생성은 상황인식협업 시스템이 담당한다.

5. 결론

유비쿼터스 환경에서 이용자를 둘러싼 환경을 구성하는 센서나 디바이스들로부터 전달된 정보를 이해하고 인지함으로써 이용자의 상황을 분석하고, 이를 통해 직접적인 이용자의 입력이 없더라도 지능적인 서비스를 제공하는 것이 상황인지 시스템의 목적이다. 따라서 상황인지 시스템은 기본적으로 다양한 종류의 정보 기기로부터 전달되는 상황 정보를 통합할 수 있고 다양한 도메인에 유연하게 적용할 수 있는 상황

인지 모델을 필요로 한다. 이러한 요구에 따라 본 논문에서는 시맨틱 웹 기술 기반의 온톨로지를 이용하여 상황인지 모델을 설계함으로써, 공통의 포맷(format)과 어휘(vocabulary)를 이용하여 상황 정보를 표현하고, 이를 통해 컴퓨터 시스템이 상황 정보를 공유하고 이해, 추론할 수 있도록 하였다.

특히, 상황인지 모델의 다양한 확장과 적용을 위해 상위 온톨로지와 도메인 온톨로지, 어플리케이션 온톨로지 구조를 계층화함으로써 이러한 일반성과 확장성을 구현하고, 기존 어휘를 모델에 재사용함으로써 온톨로지를 모듈화 하였다. 또한 효과적인 컨텍스트 추론을 위해 OWL-DL의 기술논리와 SWRL 규칙 추론을 결합하여 컨텍스트를 융/복합하고 이를 통해 복합적인 상황을 추론하는 방법을 제시하였다. 이러한 추론 방법은 특정 공간 안에서의

상황인식 뿐만 아니라 공간 간의 이동으로 발생하는 연속된 상황 간의 조합을 통한 복합적 상황인식을 가능하게 한다.

앞으로의 연구방향은 이용자의 컨텍스트 의존 관계, 선후 관계, 그리고 서비스 제공 히스토리 등을 상황인지 모델에 포함시켜 구체화된

상황인식을 통해 더욱 지능적인 서비스를 제공할 수 있는 상황인지 모델링 방법에 대한 연구가 필요하며, 특정 영역(zone) 간의 상황 연계를 통한 복합적 상황인식 방법을 확장하여 서로 다른 도메인 간의 복합적 상황인식을 위한 상황 처리 방법에 관한 연구가 필요하다.

참 고 문 헌

김수경, 양기홍. 2008. 기술논리와 SWRL 기반의 웹 온톨로지 모델링. 『정보관리학회지』, 25(1): 149-171.

박영택, 최중민. 2006. 온톨로지 추론 개요와 연구동향. 『한국정보과학회학회지』, 24(4): 17-23.

이승철, 김치수, 임재현. 2007. 의도추론의 모호성 해결을 위한 온톨로지 기반상황해석 구조의 설계 및 구현. 『한국컴퓨터종합학술대회 논문집』, 34(1): 208-213.

조위덕, 박세웅, 최재동. 2006. Ubiquitous Computing: 유비쿼터스 컴퓨팅 인프라기술 동향. 『Telecommunications Review』, 16(4).

황의관. 2008. u-City 개방형 협업 소프트웨어 플랫폼 기술. 『정보과학회지』, 26(8): 17-23.

Berners-Lee, T., J. Hendler, and O. Lassila. 2001. "The semantic web." *Scientific American*, May 2001.

Chen, H., T. Finin, and A. Joshi. 2004a. "An Ontology for Context-Aware Pervasive Computing Environments." Special Issue on Ontologies for Distributed Systems, *The Knowledge Engineering Review*, 18(3): 197-207.

Chen, H., T. Finin, and A. Joshi. 2004b. "Semantic Web in the context broker architecture," Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PERCOM'04). Washington, *IEEE Computer Society*, 277-286.

Gu, T., H. K. Pung, and D. Q. Zhang. 2005. "A service-oriented middleware for building context-aware services." *Journal of Network and Computer Applications*, 28(1): 1-18.

Krummenacher, R. and T. Strang. 2007. "Ontology-based context modeling." In Workshop on Context-Aware Proactive Systems. <http://elib.dlr.de/47459/01/CAPS07C_ameraReadyVersi>.

- Manuel, R., et al. Gaia. 2002. "A middleware infrastructure to enable active spaces." *IEEE Pervasive Computing*, Oct-Dec 2002, 74-83.
- Ranganathan, A. and B. H. Campbell. 2003. "A middleware for Context-aware agents in ubiquitous computing environments." *ACM/IFIP/USENIX International Middleware Conference 2003*.
- Ye, J., L. Coyle, S. Dobson, and P. Nixon. 2007. "Ontology-based models in pervasive computing systems." *The Knowledge Engineering Review*, 22(4): 315-347.
- Wang, X., D. Z. Zhang, T. Gu, and H. K. Pung. 2004. "Ontology based context modeling and reasoning using OWL." *PerCom Workshops, 18-22, IEEE Computer Society*.

