

# 상황인식 기술을 이용한 가정 내 안전관리 시스템

이승철\*, 임재현\*

\*국립공주대학교 컴퓨터공학과  
e-mail:scin21c, defacto@kongju.ac.kr

## Safety Management System for Home using Context-Awareness Technology

Seung-Chul Lee\*, Jae-Hyun Lim\*

\*Dept. of Computer Engineering, Kongju National University

### 요 약

유비쿼터스 환경을 위한 핵심 기술인 상황인식 기술은 발전을 거듭하고 있다. 이러한 유비쿼터스 시스템은 응용 환경에 다가가고 있다. 유비쿼터스 시스템의 가장 빠른 적용이 이루어질 것으로 예상되는 곳은 가정이다. 본 논문에서는 상황인식 기술을 이용한 가정 내의 안전관리 시스템을 제안한다. 이 안전관리 시스템은 현재 가정 내에 있는 사용자를 인지하고 상황에 맞는 서비스를 제공한다. 다수의 사용자에게 의해 발생할 수 있는 상황 해석의 문제점을 해결하고 시스템의 독립 화를 위해 온톨로지 기술을 이용하였다.

### 1. 서론

유비쿼터스 환경에서의 가장 핵심적인 기술은 상황인식 기술이다[1]. 상황인식 기술은 사용자의 의도와 현재 환경을 이해하여 현재의 상황을 추론하고 그에 적합한 서비스를 실행한다. 상황인식 기술은 현재의 상황을 효율적으로 해석하고 상황을 판단해야 한다.

유비쿼터스 환경을 지원하는 컴퓨터 시스템은 가정을 중심으로 실생활에 적용되고 있다. 가정은 사람이 생활하기 위한 가장 중요하며 기초적인 공간이다. 또한 상황인식 시스템의 도움을 가장 많이 받을 수 있는 공간이기도 하다. 최근의 사회 변화에 따른 가정 내에서의 지능화 된 안전관리 시스템의 필요성은 더욱 높아지고 있다.

기존의 가정 내의 안전관리를 위한 연구는 주로 외부에서의 네트워크를 통한 제어에 중점을 두고 있다. 본 논문에서는 안전관리 시스템에 상황인식 기술을 적용하여 시스템이 자율적으로 가정 내의 안전을 위한 조치를 취하도록 한다.

상황인식 기술에 대한 연구는 주로 단일 사용자에게 국한되어 왔다. 하지만 사회생활을 하는 인간에게 있어 단일 사용자 서비스는 큰 비중을 차지할 수 없다. 다중 사용자 환경에서는 각 사용자에게 따른 상황의 추론과 의도충돌 등으로 인해 상황인식 시스템이 상황을 해석하는데 모호성을 가질 수 있으며, 상황해석의 오류를 가질 수 있다.

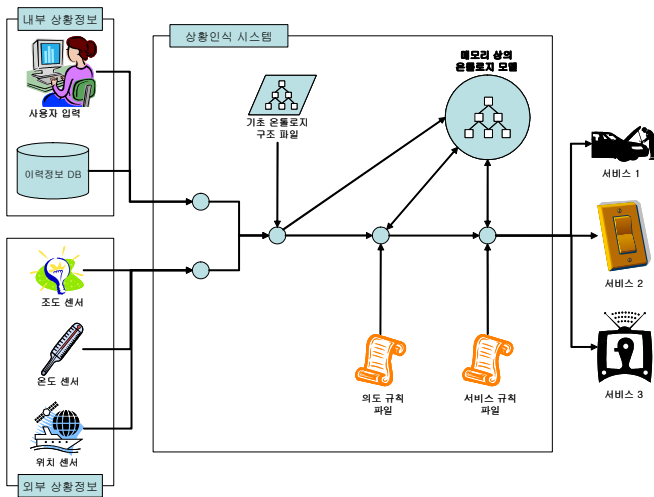
본 논문에서는 온톨로지 기반의 상황해석 구조와 규칙 기반의 추론엔진을 통해 이러한 문제점을 해결하고자 한다.

본 논문에서는 상황인식 기술을 이용한 가정 내의 안전관리 시스템을 제안한다. 이 안전관리 시스템은 상황인식 기술을 이용하여 가정 내의 사용자를 인식하고 상황을 판단하여 사고를 미연에 방지한다. 본 논문에서는 다수의 사용자가 거주하는 가정 내의 안전관리 시스템을 설계하고 온톨로지 기반 상황해석 모델을 가진 시스템을 구현하여 실험하였다.

### 2. 온톨로지를 이용한 상황해석 모델

온톨로지를 이용한 상황해석은 (그림 1)과 같은 과정을 가진다. 상황인식 시스템은 기초 온톨로지 구조를 읽어들이고 입력되는 데이터를 처리하여 기초 온톨로지 구조를 입력한다. 데이터가 입력됨으로써 단순히 클래스와 프로퍼티의 구조를 가지고 있던 기초 온톨로지 구조는 하위 클래스와 인스턴스들, 그리고 데이터 프로퍼티의 실제 값이 채워진다. 데이터와 프로퍼티 관계가 형성된 기초 온톨로지 구조는 온톨로지 모델로 불린다. 이렇게 데이터가 입력된 온톨로지 모델은 1차 추론을 거쳐 완전한 온톨로지 모델로 형성된다. 1차 추론을 거침으로써 상황해석에 필요한 5가지 요소의 값이 생성되고, 의도가 파악된다. 1차 추론을 거친 온톨로지 모델은 서비스 규칙을 이용한 2차 추론을 거쳐 서비스 검색이 이루어진다.

본 연구는 한국과학재단 특정기초연구(R01-2006-000-10555-0)지원으로 수행되었음.

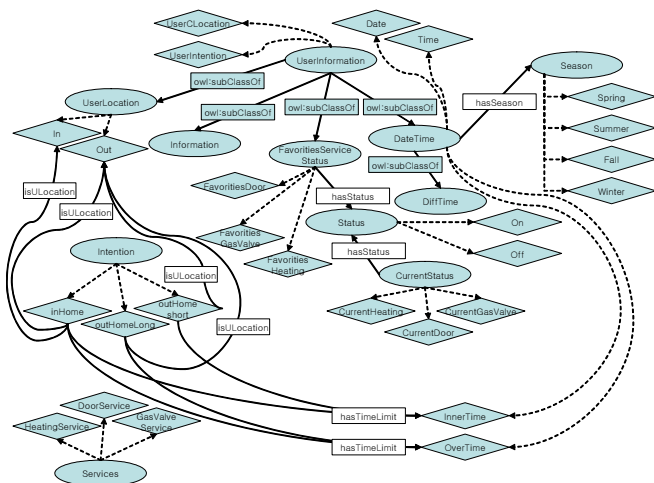


(그림 1) 상황해석 과정

### 3. 온톨로지 구조와 Rule 정의

상황 추론의 모호성과 상황 추론의 충돌은 다중 사용자 환경에서는 많은 부분에서 나타날 수 있다. 각 사용자의 상반된 의도로 인해 하나의 환경에서 서비스를 해야 하는 상황인식 시스템이 추론 자체의 모호성을 가질 수 있다. 본 시스템에서 어른과 아이가 함께 있을 경우 누구의 의도에 맞추느냐의 문제가 발생한다. 또한 각 사용자의 다른 의도로 인해 하단 서비스 간의 충돌이 발생할 수도 있다. 어른과 아이가 함께 있을 경우 상황 추론에 의해 하나의 서비스가 서로 다른 동작을 하려는 경우이다. 본 논문에서는 규칙을 통해 서비스 대상자의 우선순위를 두어 이러한 문제를 해결하고자 했다.

본 실험에서 사용된 기초 온톨로지 구조는 (그림 2)과 같은 구조를 가진다.



(그림 2) 실험에서 사용된 기초 온톨로지 구조

기초 온톨로지 구조는 클래스와 프로퍼티만이 정의되어 있다. 선언된 클래스는 실험환경에서 필요한 최소한의 정보들을 담고 있다. 각 클래스들은 필요한 인스턴스들을

가지고 있거나 DatatypeProperty를 통해 필요한 값을 가지고 있다. 혹은 ObjectProperty를 통해 다른 클래스에 속한 인스턴스나 클래스들과 관계를 가지고 있다.

실험에서 사용된 데이터는 성인과 유아의 두 가지 데이터로 각각은 개별적인 서비스 대상으로 인식이 되지만 각 개체가 가지는 특성(나이 등)에 따라 Adult, Kids 등의 Keyword를 하위 Property로 가지게 된다. 특성은 개체를 구별 지을 수 있지만, 서비스 대상이라는 최상위의 속성에 따라 시스템은 각각의 개체에 서비스를 실행하게 된다.

의도 규칙에는 총 7종류, 21개의 규칙이 포함되어 있다. 동적관계 추론은 개별적으로 정의하지 않은 subClass 관계와 계절 관계 설정이다.

의도는 각 서비스 대상의 정보(Information)을 통해 상황을 추론한다. 상황 추론에서의 충돌을 피하기 위해 서비스 검색기의 Rule에서는 서비스 대상에 우선순위를 두어 추론의 충돌에 의한 의도 추론의 모호성을 피하고 있다.

의도는 상황요소들에 연결된 Keyword를 바탕으로 해석된다. 의도 해석을 위한 규칙에는 상황요소들이 어떠한 Keyword를 가지고 있을 때 해당 의도로 해석할 수 있다는 내용을 포함하고 있다.

서비스 검색은 서비스 규칙을 이용해 추론된다. 서비스 규칙의 목적은 추론된 의도에서 실행될 수 있는 서비스를 검색하는 것이다. 온톨로지 모델의 Services 클래스에는 실행 가능한 서비스 목록이 인스턴스화 되어 있다. 서비스 검색기는 상황인식 해석기에서 추론된 의도와 관계를 설정하고 있는 서비스 목록을 모두 검색하여 hasService 프로퍼티로 관계를 설정한다.

본 시스템에서 정의된 의도는 다음과 같다.

의도	설명
SecureNot	가정 내에 위험요소 관리 및 안전 제어가 가능한 대상이 존재
SecureKeep	이전 상태의 가정 내 안전 시스템을 유지
SecureAll	가정 내에 서비스 대상이 존재하지 않음에 따라 시스템이 자율적으로 가정 내 안전을 제어
SecureKids	가정 내에 위험요소 관리 및 안전 제어가 어려운 유아가 존재 시스템이 가정 내 안전을 제어

### 4. 실험 시나리오

실험의 시나리오는 가정을 대상으로 한다. 가정 내에서 구성원에 따라 적합한 안전관리 서비스를 실행해 줄 수 있는가가 실험의 목표이다. 서비스를 받는 대상은 가정의 구성원으로 성인과 어린아이이다. 가정 내의 현재 구성원이 누구인가에 따라 각기 다른 서비스가 실행된다. 구성원에서의 어린아이는 가정 내의 장치에 대한 작동이 서투르

거나 오작동을 할 가능성이 높은 유아로 제한한다. 본 실험에서 유아의 정의는 10세 미만으로 한정하였다. 또한 외출을 2가지로 구분하여 단기외출과 장기외출로 정의하였다. 단기외출은 잠시 가까운 곳(집 근처의 슈퍼마켓 등)을 방문하기 위한 외출이다. 단기외출과 장기외출의 구분은 외출 후의 시간을 측정하여 판단한다. 서비스는 안전과 관련된 것들로 출입문의 잠금장치, 가스 제어기, 보일러 제어기이다. 가스 제어기와 보일러 제어기는 두 가지의 서비스를 제공한다. 가스 제어기는 잠금장치의 Open/Close와 조작을 제어하는 Lock/Unlock의 서비스를 제공한다. 보일러 제어기는 보일러의 On/Off와 조작을 제어하는 Lock/Unlock의 서비스를 제공한다.

아래의 (표 1)은 시나리오와 그에 따른 예상 서비스이며, (표 2)는 각 시나리오별 예상 의도 추론 결과이다.

<표 1> 시나리오별 예상 서비스

	시나리오1	시나리오2	시나리오3	시나리오4
출입문 잠금장치	Unlock	Lock	Lock	Lock
가스 밸브 제어기	Open	이전상태 유지	Close	Close
가스 조작 제어기	Unlock	이전상태 유지	Lock	Lock
난방 작동 제어기	계절에 따라	계절에 따라	계절에 따라	계절에 따라
난방 조작 제어기	Unlock	이전상태 유지	Lock	Lock

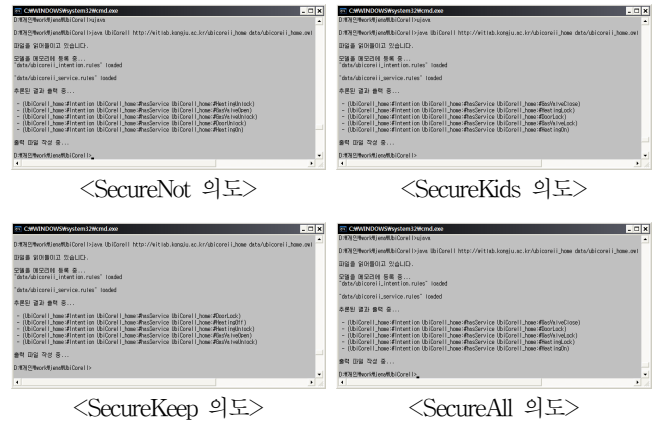
<표 2> 각 시나리오별 예상 의도 추론 결과

	환경	의도
시나리오1	어른, 아이	secureNot
시나리오2	어른 단기외출, 아이	secureKeep
시나리오3	어른 장기외출, 아이 장기외출	secureAll
시나리오4	어른 단·장기외출, 아이	secureKids

5. 실험 및 실험결과 고찰

실험을 위해 기초 온톨로지 구조를 작성하고 의도 규칙과 서비스 규칙을 작성하여 실험하였다[2]. 기초 온톨로지 구조의 작성은 OWL DL을 준수하는 구조로 작성하였다[3]. 상황해석 처리를 위한 프로그램은 Java2 SE 1.5.2를 기반으로 작성하였으며, 온톨로지의 처리 및 추론을 위

해 HP Lab에서 개발한 Jena2 Framework를 사용하였다[4].



(그림 3) 실험 결과 출력 화면

(그림 3)은 실험용 프로그램에서 명령 프롬프트를 통해 추론된 서비스가 나타남을 보여준다.

5.1 비교평가

온톨로지를 이용한 상황해석 구조와 온톨로지를 사용하지 않은 상황해석 구조는 다양한 상황을 적용하기 위한 프로그램의 수정에 대한 복잡성으로 비교할 수 있다. 이러한 비교를 위해 상황의 변화에 대한 각각의 수정 사항을 비교하였다.

온톨로지를 사용하지 않은 상황해석 구조에서는 상황의 변화에 대응하기 어려운 단점을 가지고 있다[5]. <표 3>는 새로운 서비스 대상이 추가됨에 따라 온톨로지를 이용한 경우와 이용하지 않은 경우의 프로그램 수정 사항의 간략한 비교이다.

<표 3> 온톨로지 이용에 따른 비교 평가

온톨로지 이용	<ul style="list-style-type: none"> <li>○ 사용자 클래스에 인스턴스 추가</li> <li>○ 대상에 따른 키워드 추가</li> <li>○ 새로운 상황에 대한 의도 해석 규칙 추가</li> </ul>
온톨로지 미 이용	<ul style="list-style-type: none"> <li>○ 새로운 사용자에 대한 새로운 변수 선언(혹은 변수 타입 변경)</li> <li>○ DB의 사용자 테이블 구조 변경(필요 시)</li> <li>○ 새로운 사용자에 대한 코드 값 추가</li> <li>○ 대상 판단 알고리즘 추가</li> <li>○ 상황해석 알고리즘 수정(새로운 대상에 대한 의도 추가)</li> </ul>

5.2 온톨로지 평가

2가지 이상의 상황이 충돌하여 발생할 수 있는 상황해석의 모호성은 시스템이 얼마나 데이터의 특성을 정확히 파악하고 데이터의 관계를 설정하느냐에 따라 해결될 수

있다. 보다 정확한 상황의 파악을 위해서는 더 많은 데이터가 필요하다. 하지만 데이터가 많아질수록 온톨로지의 구조는 복잡해지고, 결과적으로 명확하지 못한 온톨로지 구조가 생성될 수 있다. 이러한 온톨로지 구조는 결국 추론 상에서의 오류와 두 가지 이상의 상황에서 상황 추론을 결정짓지 못하는 추론의 모호성을 가질 수 있다.

따라서 온톨로지의 구조상에 오류가 없는지, 온톨로지를 구현한 구문에 문제는 없는지를 검사하는 것은 온톨로지를 이용한 상황 추론에서 중요한 평가가 될 수 있다. 온톨로지 개발 툴을 이용하여 온톨로지의 Validation을 수행할 수 있는데, 이 Validation은 구문 검사, 순환성, 일관성 등을 검사할 수 있다. 온톨로지의 기술적 평가를 위해 온톨로지 개발 툴인 Protégé 2000의 plug-in을 이용하여 온톨로지 구조의 Validation을 수행하였다.

Validation을 통해 온톨로지 구조가 추론엔진을 통해 정상적인 추론을 할 수 있는 구조임을 확인할 수 있다. 추론엔진은 온톨로지 구조에 의해 영향을 받을 수 있다. OWL DL을 준수하는 온톨로지 구조는 추론엔진을 통해 명확한 추론이 가능하다.

본 논문에서 제안하는 온톨로지 구조는 OWL DL을 준수하여 작성하였으므로 명확한 추론이 가능하며, 이를 온톨로지 개발 툴의 Validation을 통해 검증하였다.

<표 4> 각 환경에서의 서비스 추론

시간	대상의 상태			의도추론 결과
	어른1	어른2	유아	
10:30:28	O	O	O	SecureNot
10:36:44	X	O	O	SecureNot
10:37:51	X	X	O	SecureKeep
11:07:51	X	X	O	SecureKids
11:32:13	O	X	O	SecureNot
11:35:17	X	X	O	SecureKeep
11:37:19	O	X	O	SecureNot
11:49:25	X	X	X	SecureAll
11:55:19	X	O	X	SecureNot
12:06:44	O	O	X	SecureNot
12:07:08	O	O	O	SecureNot

온톨로지의 기술적 평가와 함께 완성도를 평가하였다. 완성도 평가를 위해 실험의 시나리오를 통해 적절한 서비스가 추론되는지 실험하였다. 상황해석 시스템을 정량적으로 평가하기는 매우 어렵다. 본 평가에서는 시나리오에 따라 주어진 데이터와 상황에서 적절한 의도가 파악되고 서비스가 추론되는지 평가하였다.

실험에서는 다중 사용자를 고려한 온톨로지 구조를 변경 없이 사용하였으며, 추론 규칙 또한 변경 없이 사용하

였다.

위의 (표 4)는 각 환경에서의 추론 결과표이다. 표를 통해서 각기 다른 환경에서도 정상적인 추론이 발생함을 알 수 있다. 서비스 대상이 1명에서 2명으로 바뀌더라도 서비스 대상에 따른 의도가 추론된다. 또한 서비스 대상이 2명인 환경에서도 대상에 따라 의도가 추론된다. 어른이 2명인 환경에서 어린아이에 대한 의도인 SecureKids가 추론되지 않는 것을 볼 수 있다.

이는 규칙을 통해 사용자가 어른인지 어린아이인지 파악하고, 그에 따라 추론을 하기 때문이다. 이에 따라 어린아이를 대상으로 하는 의도인 SecureKids는 어른이 1명인 환경과 어른이 2명인 환경에서 해당 대상인 어린아이가 없기 때문에 추론되지 않는다.

실험을 통해 본 논문에서 제안하는 온톨로지 구조가 서비스 대상이 바뀌더라도 일관성 있는 추론을 할 수 있음을 알 수 있다.

## 6. 결론

본 논문에서는 다중 사용자가 존재하는 가정에서의 상황인식 기술을 이용한 안전관리 시스템을 제안하였다. 이 안전관리 시스템은 가정 내에서의 사고가 발생할 수 있는 장치들을 제어한다. 다중 사용자 환경에서의 상황인식 시스템은 상황을 추론함에 있어 모호성을 가질 수 있고, 충돌하는 상황을 추론할 수도 있다. 이를 위해 온톨로지 기반의 추론엔진을 이용하였으며, 우선순위 방식의 규칙을 작성하여 문제점을 해결하고자 했다.

상황인식 기술은 점점 많은 사용자를 대상으로 하는 서비스를 위한 기술로 발전해 나가야 할 것이다. 본 연구는 다중 사용자 환경에서 좀 더 정확한 추론을 위한 연구로 진행되어야 할 것이다.

## 참고문헌

- [1]. S. S. Yau and F. Karim, "An Adaptive Middleware for Context-Sensitive Communications for Real-Time Applications in Ubiquitous Computing Environments", Real-Time Systems, 26(1):29-61, January 2004.
- [2]. A. Harth and S. Decker. "OWL Lite-Reasoning with Rules", WSML Working Draft, 2005.
- [3] <http://www.w3.org/TR/owl-features/>
- [4] <http://jena.sourceforge.net/>
- [5] Christopoulou, E., Goumopoulos, C., Zaharakis, I., Kameas, A., "An Ontology-based Conceptual Model for Composing Context-Aware Applications", Research Academic Computer Technology Institute (2004).