

의도추론의 모호성 해결을 위한 온톨로지 기반

상황해석 구조의 설계 및 구현

이승철^o 김치수 임재현

국립공주대학교 컴퓨터공학과

scin21c@kongju.ac.kr, cskim@kongju.ac.kr, defacto@kongju.ac.kr

Design and Implementation of Context-Aware Architecture

based on Ontology to solve the ambiguity of an Intention Reasoning

Seung-Chul Lee^o Chi-Su Kim Jae-Hyun Lim

Dept. of Computer Engineering, Kongju National University

요 약

상황인식 시스템의 추론 기능은 점점 더 중요해져가고 있다. 정확하고 오류 가능성이 적은 상황 추론 기능은 상황인식 시스템에서 중요한 역할을 한다. 온톨로지는 규정된 규칙을 이용한 추론엔진을 지원한다. 이러한 추론엔진을 이용함으로써 상황 추론에 있어 추론의 모호성을 해결할 수 있다. 본 논문에서는 온톨로지 기반 상황 인식 시스템을 제안한다. 본 논문에서 제안하는 상황 인식 시스템은 상황 해석에서 발생할 수 있는 모호성을 줄일 수 있다. 온톨로지 기반 상황 인식 시스템의 효용성을 확인하기 위해 가정을 대상으로 한 구현과 실험을 실시하였다.

1. 서론

기존의 상황인식 시스템은 새로운 상황에 대응하기가 쉽지 않다. 기존의 상황인식 시스템은 상황해석을 프로그램에서 수행하고 있다. 새로운 상황에 적용하기 위해서는 프로그램의 수정이 필요하고, 상황이 복잡해짐에 따라 프로그램 또한 복잡해진다. 이렇게 프로그램의 복잡도가 상승하게 되면, 프로그램은 상황에 더욱 대응하기 힘들어진다. 변화하는 상황을 상황해석 시스템의 프로그램을 수정함으로써 대응하는 방법은 프로그램이 복잡해질수록 더욱 프로그램 구조를 이해하기 어렵고, 수정하기 어렵게 만들기 때문이다.

온톨로지는 데이터 사이의 관계를 생성하고, 컴퓨터가 데이터를 이해할 수 있도록 도와준다[1]. 상황을 해석하기 위해서는 수많은 데이터가 필요하며 데이터는 인간에 의해 입력되거나, 센서를 통해 입력된다. 하지만 데이터는 인간이 이해하기 쉬운 문자로 구성되어 있어 컴퓨터가 단어의 의미를 이해한다는 것은 불가능하다. Gruber는 온톨로지를 “공유된 개념화(shared conceptualization)에 대한 정형화되고 명시적인 명세(formal and explicit

specification)”라고 정의하였다[2]. 즉, 온톨로지는 해당 영역의 개념과 이들 개념들 사이의 관계를 설정하는 것으로써, 컴퓨터가 상황을 이해하고 데이터를 해석하는데 도움을 줄 수 있다. 또한 온톨로지의 특징 중 하나인 규칙(rule)을 이용한 추론을 함으로써 스스로 관계를 정의하고, 필요한 정보를 추론할 수 있다[3]. 온톨로지 기반 상황인식 시스템에서는 온톨로지 구조와 규칙을 재 정의하고 배포함으로써 변화된 상황 혹은 전혀 다른 상황에서도 상황인식 시스템의 프로그램 수정 없이 적용이 가능하다[4]. 온톨로지 기반 상황인식 시스템은 프로그램 위주의 상황인식 시스템에 비해 관리, 유지, 보수에 편의성이 높다고 볼 수 있다. 이러한 온톨로지의 특징과 이점은 상황인식 시스템에서 많이 이용되고 있다. 메릴랜드 대학에서는 Harry Chen의 주도하에 CoBra Ontology라는 온톨로지 구조가 개발되었다[5]. 온톨로지 도메인의 특성과 확장성을 이용한 SOUPA는 유비쿼터스 환경에서 사용할 수 있는 개념을 정리하고 있다[6]. IBM에서는 모바일 기기에서 이용할 수 있는 Celadon Ontology를 개발하였으며[7], 국립 타이완 대학에서는 SOA라고 불리는 이벤트 상황에서 실시간으로 온톨로지의 구조를 변경할 수 있는 온톨로지 구조를 개발하였다[8].

본 연구는 산업자원부와 한국산업기술재단의 지역혁신 인력양성 사업으로 수행된 연구과제임

상황을 추론함에 있어 하나의 상황을 두 가지 이상으로 해석하는 추론의 충돌이 발생할 수도 있다. 이러한 상황해석의 문제를 온톨로지를 이용하여 줄일 수 있다. 온톨로지는 구조화된 포맷을 가진다. 따라서 구조가 어느 환경에도 대응할 수 있도록 명확하고 범용 적으로 작성된다면 프로그램의 수정을 최소화하면서 새로운 상황에 대응할 수 있다. 온톨로지는 추론엔진의 지원을 받을 수 있다. 추론엔진은 정해진 문법에 따라 작성된 규칙을 이용하여 추론을 한다. 온톨로지 기반 추론엔진을 이용함으로써 추론의 충돌 등으로 인한 상황해석의 모호성을 줄일 수 있다.

본 논문에서는 온톨로지를 이용하여 프로그램의 복잡성을 낮추고, 해석의 모호성을 보완할 수 있는 온톨로지 기반 상황해석 구조를 제안한다. 온톨로지 기반 상황해석 구조의 실험을 위해 온톨로지 구조와 규칙을 작성하고 실제 프로그램을 개발하여 실험하였다. 또한 온톨로지의 기술적 평가를 위해 Protégé 2000을 이용한 온톨로지 유효성(validation)을 평가하였다.

2. Ontology 기반의 상황해석구조

2.1. 상황해석 프로세스

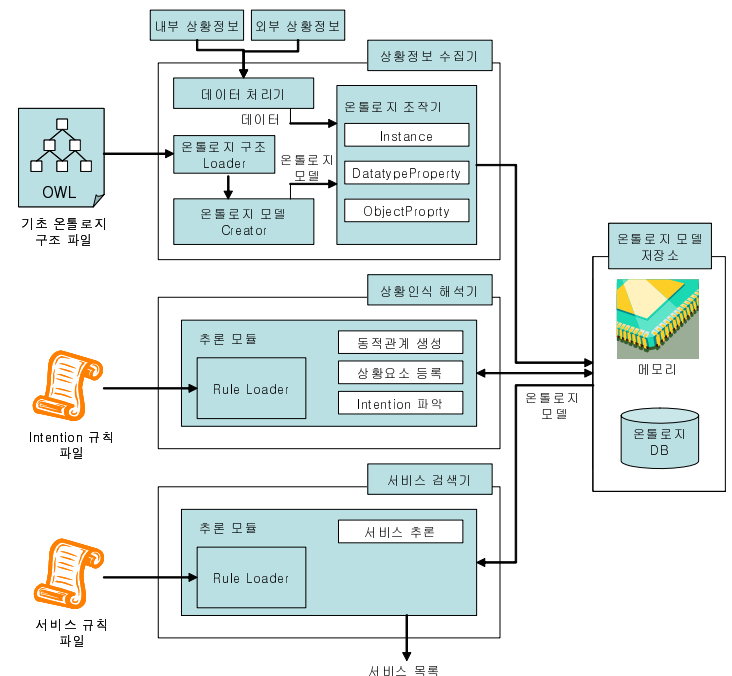
상황해석을 위해서는 여러 가지 데이터가 필요하다. 이러한 데이터는 상황인식 서비스가 필요한 상황에 따라 달라진다. 또한 데이터는 센서로부터 입력되거나, 사용자에 의해 입력될 수 있으며, 기존의 데이터를 사용할 수도 있다. 데이터의 구분과 데이터의 성격을 위해 5가지의 요소를 사용한다. 이 5가지의 요소는 데이터를 수집하고 상황을 해석하기 위해 필요한 요소으로써, "Object", "Time", "Location", "Action" 그리고 "Intention"이다[9]. Object, Time, Location, Action은 입력된 데이터를 위한 상황요소으로써 최종적으로 Intention을 파악하기 위한 기초 데이터가 된다.

2.2. 온톨로지 기반 상황해석 프로세스

온톨로지 기반 상황해석은 (그림 1)과 같은 과정을 가진다. 상황인식 시스템은 기초 온톨로지 구조를 읽어 들이고 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 구조를 온톨로지 모델이라 칭한다. 이렇게 데이터가 입력된 온톨로지 모델은 1차 추론을 거쳐 완전한

온톨로지 모델로 형성된다. 1차 추론에서는 각 데이터(클래스, 인스턴스) 간의 하위 관계가 생성되고, Intention 규칙에 의한 데이터 간의 관계가 생성된다. 또한 1차 추론을 거침으로써 상황해석에 필요한 5가지 요소의 값이 생성되고, Intention이 파악된다. 1차 추론을 거친 온톨로지 모델은 서비스 규칙을 이용한 2차 추론을 거쳐 서비스 검색이 이루어진다.

추론이 2회 일어나야 하는 이유는 1차 추론을 통해 온톨로지 모델이 데이터, 의도 규칙 등을 통해 완전한 하나의 구조로 만들어지고, 이렇게 만들어진 구조를 서비스 규칙을 통한 2차 추론을 통해 서비스 검색을 하기 위함이다.



(그림 1) 온톨로지 기반 상황해석 구조의 상세 프로세스

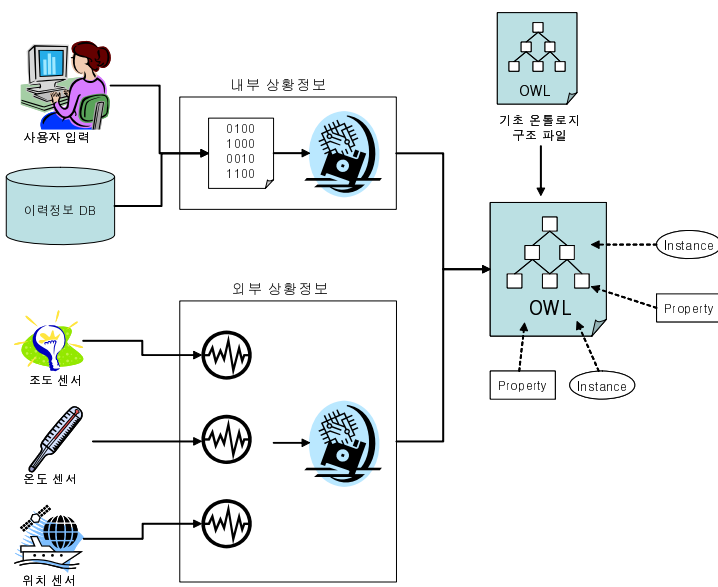
온톨로지 기반 상황해석 구조는 상황정보 수집기, 상황인식 해석기, 서비스 검색기의 세 가지 구조로 구성된다.

2.2.1 상황정보 수집기

상황정보 수집기는 데이터를 처리하고 온톨로지 모델에 입력하는 역할을 한다. 상황정보 수집기는 내부 상황정보와 외부 상황정보를 수집한다. 내부 상황정보는 데이터베이스, 파일, 메모리 등에 저장되어 있는 데이터 혹은 상황에 따라 사용자가 입력하게 되는 입력 데이터 등이 있다. 내부 상황정보는 미리 입력된 사용자의 신원정보, 서비스되었던 이력정보, 사용자 취향정보 등으로 기존에 입력되어 있어야 하거나, 누적되어야 할 데이터 들

이다. 외부 상황정보는 센서 등에 의해 입력되는 데이터이다. 상황인식에 필요한 모든 센서가 대상이 될 수 있다. 이러한 센서들은 사용자를 인식하거나 사용자의 위치를 추적하고, 사용자의 행위를 인지할 수 있다.

상황정보 수집기에서는 기초 온톨로지 구조를 읽어 들여, 데이터를 입력한다. 기초 온톨로지 구조는 클래스와 프로퍼티 만으로 구성되어 있으며, 추상적인 선언으로만 구성되어 있다. 데이터가 입력된 기초 온톨로지 구조는 온톨로지 모델로 생성되어 메모리에 등록된다. 등록된 온톨로지 모델은 이후 데이터의 입력과 추론 등을 통해 몇 차례에 걸쳐 구조가 변경된다. 기초 온톨로지 구조가 메모리에 등록된 후 상황정보 수집기는 데이터를 수집한다. 데이터는 앞서 언급한 내부 상황정보와 외부 상황정보이다. 센싱 등을 통해 이벤트 시점이 되면, 입력된 데이터는 온톨로지 모델에 입력된다. 우선 내·외부 상황정보들이 온톨로지 모델에 정의된 각 클래스들의 인스턴스나 DatatypeProperty로 생성된다. 인스턴스는 클래스의 특징을 받은 하나의 개체로써 사용자, 서비스 등이 어떠한 대상이다. DatatypeProperty는 이러한 대상들 혹은 클래스들이 가지게 되는 실제 값으로써 사용자의 이름, 나이 등 대상의 특징들이다. 이들이 이후 관계의 선언, 추론 등에 사용되는 실제 데이터들이다. 데이터가 입력된 후 인스턴스와 데이터, 인스턴스와 클래스, 인스턴스들 간의 관계가 ObjectProperty를 통해 연결된다. ObjectProperty는 관계로써 클래스들의 하위관계, 서로 다른 클래스에 속해 있는 인스턴스들 간의 관계 등이다.



(그림 2) 상황정보 수집기의 프로세스

(그림 2)는 상황정보 수집기의 전체 프로세스를 보여준다.

2.2.2 상황인식 해석기

상황인식 해석기의 최종적인 목표는 사용자 Intention의 파악이다. 이를 위해 상황인식 해석기는 온톨로지 구조에 의도 규칙(Intention Rule)을 적용한다. 의도 규칙은 동적인 상황에 따른 관계의 생성과 Object, Time, Location, Action 요소의 값을 생성하고, 최종적으로 Intention을 파악한다.

상황인식 해석기는 작성된 의도 규칙을 읽어 들인다. 의도 규칙에는 세 종류의 규칙이 담겨 있다. 첫 번째 규칙은 상황에 따른 관계의 생성으로 추론에 의해 생성될 동적인 관계를 담고 있다. 두 번째 규칙은 Object, Time, Location, Action 요소의 값 생성이다. 값들은 온톨로지 구조에 포함된 데이터들에 기반 하여 생성된다. 생성된 값들은 최종적으로 Intention을 파악하는데 사용된다. 마지막 세 번째 규칙은 Intention 해석을 위한 규칙이다. 두 번째 규칙을 통해 각 요소에 생성된 값들을 통해 Intention이 해석된다.

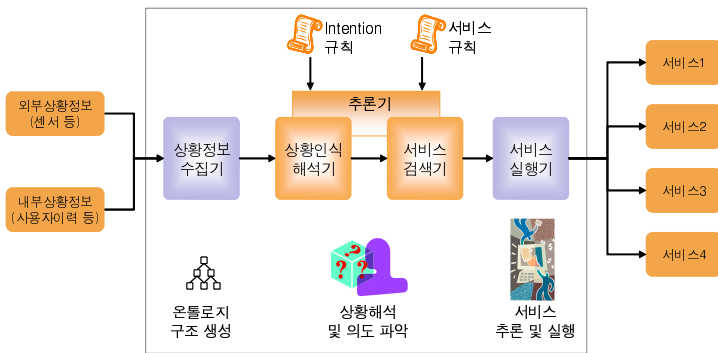
상황인식 해석기는 메모리에서 온톨로지 모델을 불러 오고, 파일로 저장된 의도 규칙 파일을 읽어 들인다. 의도 규칙 파일의 첫 번째 규칙에 따라 각 대상들의 동적인 관계를 설정한다. 동적인 관계는 상황에 따라 관계가 바뀌어야 하는 것들이다. 봄, 여름, 가을, 겨울과 같은 계절은 날짜에 따라 변경된다. 이렇게 특정한 상황에 따라 변경되는 것을 동적인 관계의 대상이라고 볼 수 있다. 센서에 의해 입력되는 시간 정보, 기상 정보, 사용자의 행위 정보, 위치 정보 등은 수시로 변경되지만 데이터 자체가 항상 변화하게 되므로, 동적인 관계의 대상이라 볼 수 없다. 동적인 관계가 생성된 후 Object, Time, Location, Action의 요소 값이 생성된다. 이 네 가지의 요소에 입력되는 데이터는 상황인식 서비스가 필요한 곳에 따라 달라질 수 있다. 요소의 값은 단어로 입력된다. 기존의 상황인식 시스템이 데이터 검색을 위해 특정 코드 값을 사용했던 것과 달리 온톨로지를 사용한 상황해석 시스템에서는 관계를 통해 데이터를 연결하므로 컴퓨터 중심의 코드 값이 아닌 프로그래머, 관리자 등 인간 중심의 단어를 사용할 수 있다. Object는 서비스를 받는 대상이 될 수 있지만, 서비스를 실행하는 주체가 될 수도 있다. 또한 Object가 서비스를 받는 대상으로 설정된다 하더라도 대상의 이름을 사용할 것인지, 성별을 사용할 것인지 등을 선택해야 한다. 본 논문에서 제안하는 온톨로지 기반 상황해석 구조에서는 상황의 해석에 키워

드를 사용한다. Object, Time, Location, Action의 상황요소에 사용할 수 있는 키워드를 정의해 둔 후 규칙을 통해 적합한 키워드와 연결한다.

세 가지의 규칙을 통해 추론된 내용은 다시 온톨로지 모델에 적용된 후 메모리에 저장된다. 이렇게 저장된 온톨로지 모델은 서비스 검색기에서 적절한 서비스를 검색하기 위해 공유된다.

2.2.3 서비스 검색기

서비스 검색기는 Intention에 의해 판단된 상황에 따른 적절한 서비스를 검색한다. 서비스들은 온톨로지 모델에 등록되어 있다. 각각의 서비스는 특정한 그룹을 가지고 있지 않으며 개별적으로 등록된다. Intention이 판단된 후 서비스 검색기는 서비스 규칙 파일을 읽어 들인다. 서비스 규칙 파일은 실행 가능한 모든 서비스의 적용 범위가 작성되어 있으며, 판단된 Intention에 적합한 서비스들이 검색된다. (그림 3)은 온톨로지 기반 상황해석 구조를 보여준다.



(그림 3) 온톨로지 기반 상황해석 구조

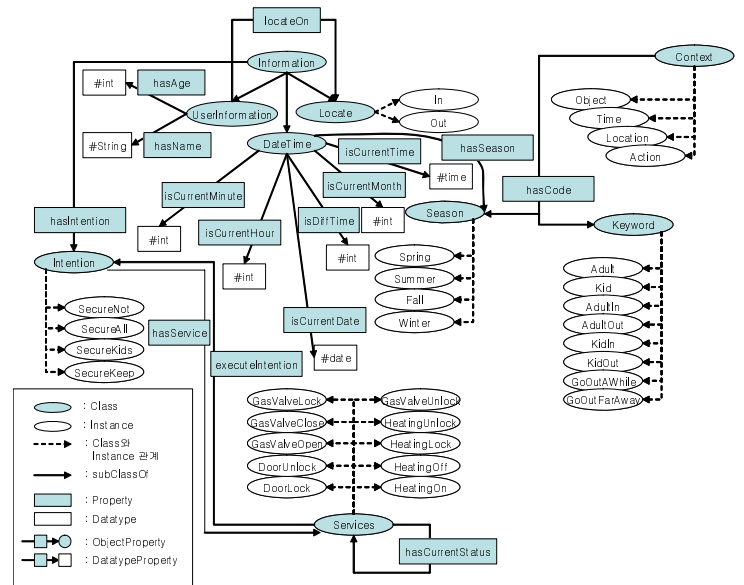
3. 실험

본 논문에서는 모호성 해결을 위한 온톨로지 기반 상황해석 구조를 제안한다. 상황해석 구조에 온톨로지가 유용하게 사용될 수 있음을 증명하기 위해 실제 기초 온톨로지 구조를 작성하고 Intention 규칙과 서비스 규칙을 작성하여 실험하였다. 기초 온톨로지 구조의 작성은 OWL DL을 준수하는 구조로 작성하였다[10]. 상황해석 처리를 위한 프로그램은 Java2 SE 1.5.2를 기반으로 작성하였으며, 온톨로지의 처리를 위해 HP Lab에서 개발한 Jena2 Framework를 사용하였다[11].

3.1. 시나리오

실험의 시나리오는 가정을 대상으로 한다. 가정 내에서 구성원에 따라 적합한 상황인식 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 서투르거나 오작동을 할 가능성이 높은 유아로 제한한다. 본 실험에서의 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. 외출 후 30분 이내는 단기외출로 정의하며, 30분 이후는 장기 외출로 정의한다. 서비스는 안전과 관련된 것들로 출입문의 잠금장치, 가스 제어기, 보일러 제어기이다. 가스 제어기와 보일러 제어기는 두 가지의 서비스를 제공한다. 본 실험에서 사용된 기초 온톨로지 구조는 (그림 4)와 같은 구조를 가진다.



(그림 4) 실험에서 사용된 기초 온톨로지 구조

기초 온톨로지 구조에는 클래스와 Property만이 정의되어 있다. 클래스들은 사물, 서비스 대상, 서비스 등을 개념적으로 정의하며, Instance를 통해 실제로 사용된다. 정의된 Property들은 클래스 혹은 Instance가 가질 수 있는 속성 값이나 특성을 정의하고 있다.

선언된 클래스는 실험환경에서 필요한 최소한의 정보들을 담고 있다. Information 클래스는 사용자 정보와 환경에 대한 최상위 클래스를 UserInformation DateTime Locate 클래스를 하위클래스로 가지고 있다. 각 클래스들은 필요한 인스턴스들을 가지고 있거나

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

실험을 위해 구현된 온톨로지 구조에서는 사용자에 대한 정보를 주로 다루고 있다. 사용자의 현재 상태(외출, 귀가)와 시간과 관련된 것들을 클래스로 정의하고 있다. 또한 개별적인 서비스들을 Instance로 정의하여 현재의 의도에 따라 다수의 서비스가 실행될 수 있다.

클래스 혹은 Instance는 Property를 통해 관계를 정의하며, 규칙 기반의 추론엔진을 통해 동적으로 관계가 생성될 수 있다. 이렇게 생성된 관계들은 의도를 추론하고, 추론된 의도에 적합한 서비스를 선택한다.

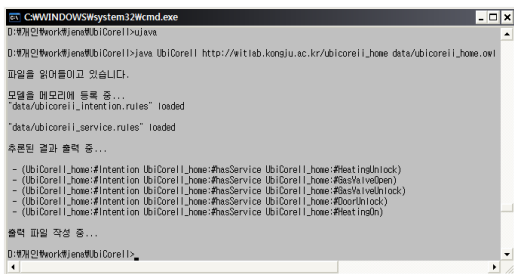
3.2. 실험 결과 고찰

시나리오에 따라 여러 가지 상황을 통해 실험을 실시하였다. 총 4가지 세부적인 시나리오가 발생할 수 있으며, 해당 시나리오들에서 적절한 추론이 이루어지는 실험하였다. 4가지의 세부적인 시나리오는 아래의 (표 1)과 같다.

(표 1) 4가지의 세부적인 시나리오

시나리오 1	어린아이 귀가, 어른 귀가
시나리오 2	모두 외출, 단기 외출
시나리오 3	모두 외출, 장기 외출
시나리오 4	어린아이 귀가, 어른 외출

위의 4가지 세부적인 시나리오에 따라 실험을 실시하여 실제 각 시나리오에 필요한 의도가 추론됨을 확인하였다.



(그림 5) 서비스 출력 화면

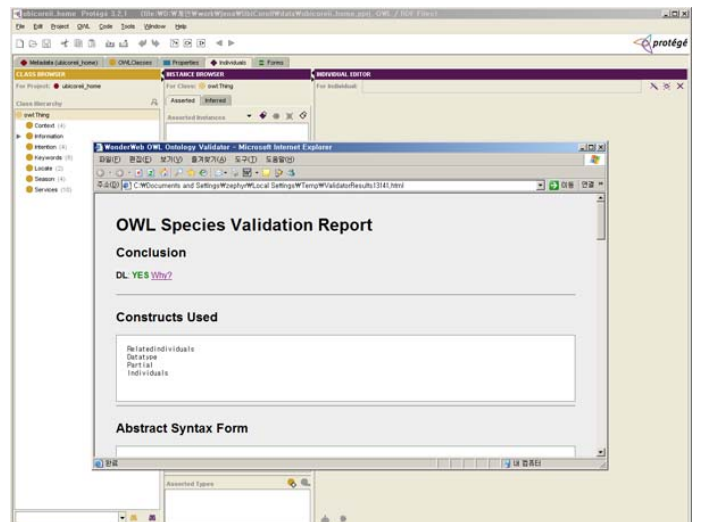
위의 (그림 5)는 의도가 추론되어 서비스가 결정된 출력 화면이다.

3.3. 실험 평가

온톨로지 기반 상황해석 구조는 상황해석의 모호성을

줄여줄 수 있다. 온톨로지 개발 툴을 이용하여 온톨로지의 Validation을 수행할 수 있는데, 이 Validation은 구문 검사, 순환성, 일관성 등을 검사할 수 있다[12]. 온톨로지의 기술적 평가를 위해 온톨로지 개발 툴인 Protégé 2000의 plug-in을 이용하여 온톨로지 구조의 Validation을 수행하였다.

현재의 추론엔진에서 OWL DL을 준수하는 온톨로지 구조는 추론엔진을 통해 명확한 추론이 가능하지만, OWL Full은 현재로써는 완전히 지원되는 추론엔진은 존재하지 않는다. 따라서 OWL DL을 준수하는 온톨로지 구조는 추론엔진의 기능을 이용할 수 있지만, OWL Full은 구조에 따라 추론에 오류가 발생할 가능성이 있다.



(그림 6) Protégé 2000 plug-in을 이용한 온톨로지 Validation 결과

본 논문에서 제안하는 온톨로지 구조는 OWL DL을 준수하여 작성하였으므로 명확한 추론이 가능하며, 이를 온톨로지 개발 툴의 Validation을 통해 검증하였다. 위 (그림 6)은 Protégé 2000 plug-in을 이용한 온톨로지 Validation 결과이다.

실험에서 사용된 온톨로지 구조가 다양한 상황에 쉽게 적용될 수 있는지를 확인하기 위해 서비스를 받는 대상에 변화를 주어 실험하였다. 실험의 상황은 아래 (표 2)와 같다.

(표 2) 실험 상황 변화

실험 상황 변화(서비스 대상)
어른 1명
어른 1명, 어린아이 1명
어른 2명

온톨로지 구조 및 추론 규칙과 프로그램의 수정없이 사용자만을 바꾸어 실험을 하였다. 아래 (표 3)는 실험의 결과를 나타내는 추론 결과표이다. 온톨로지 기반 상황 해석 구조는 큰 변화 없이 새로운 상황에 적용할 수 있음을 알 수 있다.

(표 3) 각 환경에서의 서비스 추론

환경	조건 (사용자, 상태, 상태변화 경과시간)	의도 추론 결과
어른	어른, 외출, 00:19:00	SecureKeep
	어른, 외출, 00:32:00	SecureAll
	어른, 귀가, 00:13:00	SecureNot
어른, 어린이	어른, 외출, 00:32:00 어린이, 외출, 00:45:00	SecureAll
	어른, 외출, 00:12:00 어린이, 외출, 00:25:00	SecureKeep
	어른, 외출, 00:54:00 어린이, 귀가, 00:11:00	SecureKids
	어른, 귀가, 00:05:00 어린이, 귀가, 00:35:00	SecureNot
	어른, 귀가, 00:47:00 어린이, 외출, 00:12:00	SecureNot
어른1, 어른2	어른1, 외출, 00:19:00 어른2, 외출, 00:45:00	SecureKeep
	어른1, 외출, 00:32:00 어른2, 외출, 00:58:00	SecureAll
	어른1, 귀가, 00:02:00 어른2, 외출, 01:08:00	SecureNot
	어른1, 귀가, 00:13:00 어른2, 귀가, 00:01:00	SecureNot

4. 결론

본 논문에서는 온톨로지 기반 상황해석 구조를 제안하였다. 온톨로지 기반 상황해석 구조는 프로그램의 수정을 최소화함으로써 새로운 상황에 쉽게 대응할 수 있다. 또한 온톨로지 추론엔진을 이용함으로써 정해진 문법 내에서 규칙을 작성한다. 추론엔진을 이용함으로써 프로그램 구조에 기반을 둔 상황해석보다 상황해석의 모호성을 줄일 수 있다. 실제 가정을 대상으로 하는 상황인식 시스템을 개발하고 실험을 실시하였다. 또한 실험결과를 통해 프로그램의 복잡도를 낮추고, 추론을 통한 상황해석의 모호성을 어느 정도 줄일 수 있음을 확인하였다.

참고문헌

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web", Scientific American, May 2001.
 [2] Gruber, T., "A Translation Approach to Portable

Ontologies", Knowledge Acquisition, Vol. 5, No. 2, pp.199-220, 1993.
 [3] A. Harth and S. Decker. "OWL Lite- Reasoning with Rules", WSML Working Draft, 2005.
 [4] Harry Chen, Tim Finin Anupam Joshi, "A Context Broker for Building Smart Meeting Rooms", in the Proceeding of the Knowledge Representation and Ontology for Autonomous Systems Symposium, 2004 AAAI Spring Symposium, Stanford University, CA, March 2004.
 [5] H. Chen, T. Finin, A. Joshi, "An Intelligent Broker for Context-Aware Systems", InCollection, Adjunct Proceeding of Ubicomp 2003, October 2003.
 [6] Chen, H., Perich, F., Finin, T., and Joshi A., "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications". In Proceedings of the First Annual International Conference on Mobile and Ubiquitous systems: Networking and Services (MobiQuitous 2004), Boston, MA, 2004.
 [7] Lee, M.C., Jang, H.K., Paik, Y.S., Jin, S.E. and Lee, S., "Ubiquitous Device Collaboration Infrastructure: Celadon", The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance, 2006.
 [8] Cheng, S.Y., Jin, W.R. and Hsu, Y.J., "Context-aware Policy Matching in Event-Driven Architecture", Technical report, AAAI Press, Menlo Park, California, pp. 140-141, 2005.
 [9] 임재현, "지능형 서비스를 위한 상황인식 해석 구조 설계 및 구현", 인터넷정보학회 논문집 10월호, 2006. 10.
 [10] <http://www.w3.org/TR/owl-features/>
 [11] <http://jena.sourceforge.net/>
 [12] A. Gomez-Perez, and M.C. Suarez-Figueroa, "Results of Taxonomic Evaluation of RDF(S) and DAML+OIL ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms import services", EON 2003 2nd International Workshop on Evaluation of Ontology-based Tools, 2003.