유비쿼터스 시스템을 위한 시맨틱 다중 에이전트
(Semantic Multi-agents Framework for Ubiquitous Systems)

최 정 화 † 박 영 택 ††
(Jung-Hwa Choi) (Young-Tack Park)

요 약 지난 10여년 동안 유비쿼터스 컴퓨팅 연구는 '언제나, 어디에서나, 어느 것이나 컴퓨팅 환경' 이라는 신기술 체계의 확립을 목표로 진행되었다. 이를 통하여 칩의 소형화 기술, 장치기술, 네트워킹 기술, 인간 중심의 인터페이스 기술, 응용기술 등의 기술혁신의 필요성을 알게 되었다. 본 논문에서는 인간 중심의 인터페이스 기술 측면에서 인간과 에이전트와의 상호작용 및 에이전트 스스로 인간이 원하는 서비스를 처리하고 제공하기 위하여, 다음 네 가지 단계를 제안한다. 첫째, 유비쿼터스 서비스를 사용자 요구에 맞게 서비스하기 위하여 정보자원 간에 의미적 연관성을 고려한 시맨틱 웹 기술을 이용한다. 둘째, 시맨틱 웹 기술 개발에 핵심이 되는 온톨로지를 구축하여 컴퓨터가 웹 문서의 내용을 인식할 수 있도록 한다. 셋째, W3C에서 표준화 작업 중인 차세대 웹 OWL 온톨로지 언어를 이용하여 컴퓨터와 컴퓨터 간에 메시지를 교환한다. 넷째, 컴퓨터로 전송된 메시지 분석 및 정보 수집을 위하여 FIPA의 JADE를 이용하여 다중 에이전트를 기능별로 구축한다. 본 논문에서 제안하는 시맨틱 다중 에이전트는 온톨로지 기반의 시맨틱 웹 기술을 적용하여 시맨틱 상황을 추론한다. 다중 에이전트가 OWL 온톨로지 언어를 이용하여 커뮤니케이션 함으로써 같은 상황이더라도 사용자에 따라서 제공되는 서비스가 다르게 추론된다. 따라서 기존의 상황 정보 시스템보다 에이전트에게 보다 의미 있는 정보의 예측을 가능하게 하였다.

키워드 : 유비쿼터스 시스템, 시맨틱 웹, 온톨로지, 에이전트, 에이전트 커뮤니케이션 언어, OWL

Abstract For the past ten years, the goal of ubiquitous computing research has been the establishment of a new technology system with the aim 'Anytime, Anywhere, and Any form'. The needs for agent technology innovations such as ontology-based structure, ontology-based agent communication language, and multi-agents frameworks have been identified. This paper proposes a noble multi-agents architecture for ubiquitous systems. We suggest four major steps in the interaction between human and agents which enable ubiquitous agents to process by themselves to provide adaptive service to meet human's needs. First, we propose a semantic web technology to represent the association between information resources more explicitly. Second, we construct a semantic ontology so that agents can recognize web contents. Third, we propose a method to communicate between agents using OWL ontologies. Finally, we suggest a multi-agents structure based on the JADE of FIPA to analyze messages and get information. The semantic multi-agents framework proposed in this paper infers semantic situations using semantic web technology based on ontologies. A service provided is inferred differently according to user state because the multi-agents communicate by using OWL ontology language. Therefore, our system better infers context information than other without ontologies.

Key words : Ubiquitous System, Semantic Web, Ontology, Agent, Agent Communication Language, OWL

1. 서 론

유비쿼터스 환경에서는 언제 어디서든 컴퓨터에 접근이 가능하기 때문에, 사람의 위치를 파악하고 현 상황에 맞는 적절한 서비스를 제공해 주게 된다[1]. 이러한 적절한 서비스를 제공하기 위해서는 스스로 판단하여 행동할 수 있는 에이전트를 통한 커뮤니케이션이 필요하다. 에이전트는 사용자나 다른 에이전트의 직접적인 지

· 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌습니다.
† 학생회원 : 숭실대학교 컴퓨터학과
cjh79@ailab.ssu.ac.kr
†† 종신회원 : 숭실대학교 컴퓨터학과 교수
park@comp.ssu.ac.kr
논문접수 : 2004년 3월 29일
심사완료 : 2005년 1월 28일

시나 간섭 없이 능동적으로 작업수행을 진행한다. 그리고 사용자가 제공받고자 하는 서비스나 목적달성을 위한 세부절차 등을 맡게 된다. 에이전트가 작업을 수행하다 다른 에이전트의 도움이 필요하면 에이전트 통신언어(ACL)를 이용하여 에이전트간의 메시지 교환에 의존하게 된다[2]. 하지만 모든 사물이 네트워크로 연결되기 때문에 개개인의 일거수 일투족이 다른 사람에게 노출될 수 있는 단점을 갖고 있다.

본 논문에서는 인간의 활동을 컴퓨터가 중개할 수 있도록 하기 위해서 에이전트간의 커뮤니케이션과 지식공유 및 전달기능을 정의하고, 시맨틱 웹(Semantic Web)[3]의 의미적 연관성을 이용하여 컴퓨터에 존재하는 에이전트를 네트워크화 한다. 시맨틱 웹은 월드와이드 웹을 창시한 팀 버너스리가 창안한 차세대 웹 기술로, 현재의 웹처럼 사람이 눈으로 보고 이해하는 웹이 아닌, 컴퓨터가 처리할 수 있는 웹을 의미한다. 또한 웹의 표준을 담당하고 있는 W3C를 중심으로 한 RDF, RDF-S (RDF Schema)와 DAML+OIL, OWL(Ontology Web Language) 등의 온톨로지(Ontology) 언어 기반의 온톨로지 기술로 특정 자원에 대한 메타 데이타를 기술한다. 본 논문에서는 시맨틱 에이전트 사이에 커뮤니케이션이 가능하게 하기 위해서 네 가지 단계에 중점을 두고 다중 에이전트 프레임워크를 제안한다. 유비쿼터스 서비스를 사용자 요구에 맞게 서비스하기 위해서 시맨틱 웹의 메타 데이타의 개념을 통해서 웹 문서에 시맨틱 정보를 덧붙이고, 정보를 컴퓨터가 이해할 수 있도록 한다. 또한 웹 데이타에 의미를 부여할 수 있는 표현 방법으로 시맨틱 웹 기술 개발에 핵심이 되는 온톨로지를 구축한다. 온톨로지는 특정 자원에 대한 메타 데이타를 기술하는 OWL[4]을 사용한다. OWL은 온톨로지 언어의 최신 버전으로 자원 속성 표현의 세분화로 인해 자원에 대한 정교한 표현이 가능하다. 본 시스템의 에이전트는 FIPA의 JADE를 이용하여 정보자원의 의미를 자동으로 추출하도록 기능별로 구축하였다. 에이전트들은 OWL 기반의 메시지를 통해서 커뮤니케이션 한다. 이와 같은 방식은 시맨틱 에이전트를 통하여 사용자 요구에 맞는 정보자원 의미 분석에 도움을 주고, 온톨로지에 표현된 속성과 관계에 의해서 정보를 공유하므로 사용자의 프라이버시 노출의 문제점을 해결한다. 또한 서비스를 사용자 메타 데이타와 온톨로지를 통해서 예측을 하므로 사용자 요구에 맞는 정보기기의 에이전트가 동작하게 된다.

본 논문은 2장에 관련 연구를 기술하고 있으며, 3장에 본 논문에서 제안하는 구조에 적용되는 시나리오를 설명하고, 4장에는 유비쿼터스 환경에서의 시스템 구조, 5장에는 OWL 컨텐트 메시지 전송을 위한 에이전트 간의 메시지 흐름, 6장에는 상태 인식을 위한 OWL 인스턴스 자동 생성 및 다중 에이전트 프레임워크 시뮬레이션 결과에 대해서 기술한다. 끝으로 7장에서 결론 및 향후연구 방향에 대해서 기술한다.

2. 관련 연구

오늘날의 과학기술로 인터넷의 부상은 더 쉽고, 더 나은 방향으로 발전하고 있다. 인터넷을 사용하는 사용자들은 좀더 인간적인 컴퓨터로부터 수많은 양질의 컨텐츠를 찾고, 컴퓨터와 더 편안한 커뮤니케이션을 하기를 원한다. 이 같은 요구를 해결해 줄 수 있는 연구는 현재 활발히 진행되고 있다. 본 논문에 대한 관련연구로써 두 가지 측면으로 나누어 살펴본다. 다음은 유비쿼터스 환경 기반에서 정보자원을 웹과 연결한 HP(Hewlett Packard)의 Cooltown[5]과, 시맨틱 웹 기반에서 에이전트 커뮤니케이션을 목적으로 한 UMBC의 ITTalk 프로젝트에 대해서 설명한다.

2.1 HP의 Cooltown

Cooltown 프로젝트는 현실 속의 사람, 사물, 공간이 동시에 웹상에도 존재하는 '현실 같은 월드와이드웹'을 구축하기 위해 개발되었다. Cooltown은 정보기기 간 또는 정보기기와 이용자간에 URL을 생성 및 교환하여 웹상의 한 HTML 페이지(웹 가상공간)에 링크한다. 그리고 모든 개체(사람, 사물, 장소 등)가 웹으로 통합되어 정보를 교환하는 방식으로 수행된다[6].

Cooltown은 URL 정보를 화면, 인쇄장치 등에 전송하여 각 구성요소를 상호 통합하기 위한 플랫폼으로, 정보기기 간의 컴퓨팅으로 구현되기 때문에 정보기기의 호환성을 중시한다. 현재는 HP제품만을 이용하여 웹 서비스가 지원되며, 타 정보기기의 호환성을 위한 연구를 진행하고 있다.

유비쿼터스 컴퓨팅이 궁극적으로 실세계와 가상세계와의 원활한 통합이라고 생각할 때, 이것은 매우 적합한 연구 방향이라고 판단된다. 하지만 제품마다 유비쿼터스를 지원하는 프로그램을 개발하여 설치하는 것은 유비쿼터스 컴퓨팅 사회의 실현을 늦추는 작용을 한다. 본 논문에서도 개체가 이동하여 만나게 되는 사람, 사물, 장소의 여러 변화에 따라 적절한 서비스를 제공한다. 하지만 각 제품의 응용프로그램을 통한 서비스가 아닌, 예를 들어 프린터의 기능을 제어하는 에이전트, 텔레비전의 기능을 제어하는 에이전트 등의 기능별 에이전트를 두어 메인 에이전트와의 커뮤니케이션을 통해서 서비스하는 구조를 제안한다.

2.2 UMBC의 ITTalk

ITTalk 프로젝트는 컴퓨터상의 에이전트와 웹을 연결하여 사용자와 컴퓨터간의 편리한 커뮤니케이션을 제공하기 위해서 개발되었다. ITTalk는 MS Outlook과

Yahoo Calendar를 에이전트를 통하여 자동으로 관리한다. 그래서 에이전트의 역할을 나누고, 커뮤니케이션 하여 사용자에게 알려야할 정보는 Outlook을 통해서 통보하고, 추가나 변경된 일정은 Yahoo Calendar의 사용자 스케줄을 조정한다[7].

ITTalk은 웹 온톨로지 언어인 DAML을 메시지 내용(content)으로 하여 FIPA의 ACL(Agent Communication Language)[8] 메시지로 에이전트 간에 커뮤니케이션을 제공한다. 또한 XSB[9]로 구축한 추론엔진을 이용하여 사용자 메타 데이타를 통한 새로운 데이타를 예측한다. 하지만 DAML의 속성으로는 유비쿼터스 환경을 표현하기에 부족하다. 본 논문은 유비쿼터스 환경에서 지능적인 웹 서비스의 제공을 목적으로 한다. 따라서 컴퓨터에 에이전트를 두고 DAML보다 속성의 표현이 풍부한 OWL을 사용하여 에이전트 간에 커뮤니케이션을 한다.

3. 적용 시나리오

사람들은 일상생활 속에서 집, 사무실, 백화점 등의 공간을 이동하면서 다른 사람과 만나 대화하고 주변 사물을 사용한다. 본 논문에서는 사람, 사물, 장소가 웹으로 통합된 유비쿼터스 컴퓨팅을 구현하기 위해서 다음과 같은 시나리오를 적용한다. 이 시나리오는 대학교의 한 건물을 하나의 플랫폼으로 구성하고, 하나의 방을 하나의 컨테이너로 구성한다.

213호 연구실에 있는 학생("사람" 자원)이 301호 교수실("장소" 자원)에 들어온다. 이때 학생과 교수실에 부착된 전자태그로부터 ID를 얻어 웹 사이트 URL로 변환되고, 이미지와 같은 웹 표현과 동적인 링크가 형성된다. 교수실의 에이전트는 플랫폼의 메인 에이전트로부터 학생 ID가 현 플랫폼에서 서비스 가능한 사용자인지 인식한다. 서비스 가능한 사용자라면, 교수실에 들어온 이유를 알기 위해 213호 연구실의 학생 에이전트에게 학생의 스케줄 정보를 요청한다. 학생의 스케줄을 확인한 학생 에이전트는 학생이 "교수님과의 A 프로젝트 회의"를 하기위해 갔음을 교수실 에이전트에게 통보한다. 교수실 에이전트는 A 프로젝트 회의를 위한 관련된 파일이 있는지 213호 학생 에이전트에게 다시 요청하게 되고, 학생 에이전트는 관련된 파일의 URL을 교수실 에이전트에게 통보한다. 교수실 에이전트는 분석된 정보를 메인 에이전트에게 통보하고, 메인 에이전트는 학생에게 서비스할 항목을 찾는다. 그리고 관련 디바이스 에이전트에게 통보하여 디바이스를 작동시키게 된다. 이 시나리오에서는 회의에 필요한 파일을 보여줄 컴퓨터가 필요하다. 메인 에이전트는 교수 에이전트로부터 학생과의 관계를 파악하고, 교수실에 있는 교수 컴퓨터에 학생의 회의 자료를 출력하게 된다.

사용자가 최초의 환경에서 다른 환경으로 이동할 때, 사용자가 계속해서 작업을 수행할 수 있도록 하기 위해서 사용자의 환경을 새로운 환경으로 변경해야 한다. 그러나 에이전트가 사용자 요구를 파악하지 못하면 어떠한 결정도 불가능하다. 본 논문에서는 사용자가 한 곳에서 다른 곳으로 이동한 것을 인식하고, 위치정보를 파악하여 서비스하기 위해 위와 같은 시나리오를 바탕으로 유비쿼터스 환경에서의 시스템 구조를 제안한다.

4. 유비쿼터스 환경에서의 시스템 구조

유비쿼터스 환경에서 사용자의 다양한 서비스 요구를 지능적으로 처리하기 위해서는 물리적 자원(physical resource)과 논리적 자원(logical resource)의 시맨틱 정보를 표현, 저장, 추론, 검색하는 기능이 필요하다. 본 절(節)에서는 자동화 에이전트로 하여금 의미 있는 정보를 추출하여 사용자가 요구하는 유비쿼터스 서비스를 효율적으로 수행할 수 있는 다음과 같은 온톨로지 기반의 유비쿼터스 시맨틱 웹 에이전트 시스템 구조를 설명한다. 본 절에서 설명하는 유비쿼터스 기반의 전체적인 시스템 구조는 그림 1과 같다.

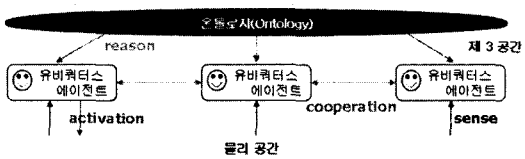

그림 1 유비쿼터스 시스템 구조

그림 1에서와 같이 학교, 직장과 같은 삶의 공간인 물리공간으로부터 사람과 디바이스 등의 정보자원을 네트워크 상의 에이전트가 감지(sense)한다. 그리고 감지된 자원에 대한 에이전트들이 생성되어 제3의 공간에 존재하게 된다. 제3의 공간이란 물리공간에 칩이 심어지고 네트워크로 연결되어(전자공간이 이식되어) 서로 교신하게 될 때 이루어지는 공간이다. 에이전트들은 구축된 온톨로지를 통해서 추론(reason)하고, 에이전트 간에 커뮤니케이션을 통해서 해당 서비스를 제공하여야 할 에이전트에게 메시지를 전달한다. 메시지를 전달받은 에이전트는 메시지를 분석하여 물리공간에 반영한다.

4.1 시맨틱 에이전트 모듈 설계

기존의 HTML로 작성된 웹은 사람을 중심으로 작성되어 있어서 에이전트가 처리할 수 없다. 또한 웹상의 데이타를 자동화하는 것이 어렵고, 웹상에 존재하는 정보량이 많기 때문에 수작업으로 데이타를 처리하는 것은 불가능하다. 이에 대한 해결책으로 시맨틱 웹의 요소

인 메타 데이타를 사용하여 감지한 정보를 기술하게 된다.

물리공간에서 센서(sensor)를 통해서 감지된 사용자의 현재 상황(context)은 메타 데이타의 스키마인 온톨로지를 통하여 의미가 분석되어 컨텍스트(context) 컨테이너에 추가된다. 온톨로지는 개념과 관계들로 구성되어 특정 도메인에 관련된 객체들의 계층적 구조를 표현하고, 유비쿼터스 에이전트는 온톨로지를 기반으로 표현되지 않은 객체들을 추론할 수 있는 추론규칙을 포함한다. 컨텍스트 컨테이너에 존재하게 되는 메타 데이타는 웹 표현을 유비쿼터스 에이전트가 이해할 수 있게끔 유비쿼터스 컨텍스트에 추가되어 에이전트에게 전달된다. 물리공간에 반영된 시맨틱 에이전트의 제3의 공간에서의 구조는 그림 2와 같다.

본 논문에서는 하나의 고유 공간을 하나의 컨테이너로 구성한다. 컨테이너에는 각각 다른 기능을 수행하는 에이전트들이 존재한다. 에이전트들은 컨텍스트 컨테이너에 존재하는 메타 데이타를 공유한다. 메타 데이타는 데이타의 호환성을 유지하고, 네트워크 자원의 기술에 필요한 일련의 데이타요소를 규정하며, 이들 자원을 신속하게 검색할 수 있는 정보이다. 각 에이전트들은 자신의 도메인 지식베이스(Domain Knowledge Base)와 온톨로지를 기반으로 수행해야 하는 작업(task)를 예측한다. 예측한 작업이 자신이 서비스할 수 있는 작업에 속하지 않으면 에이전트 간에 커뮤니케이션을 통해서 다른 에이전트에게 전달되고 물리공간에 반영된다.

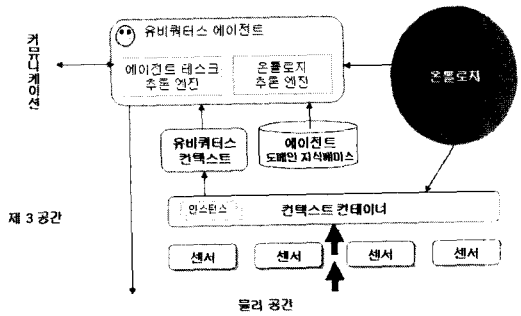

그림 2 시맨틱 에이전트 구조

센서로부터 감지된 데이타는 온톨로지 인스턴스로 구성되어 온톨로지를 기반으로 분석된다. 분석된 인스턴스는 그 기능에 맞는 에이전트에게 전달된다. 유비쿼터스 에이전트는 두가지 추론 엔진으로 구성된다. 첫째, 에이전트 테스크 추론엔진은 유비쿼터스 컨텍스트의 메타 데이타 기능에 맞는 에이전트가 수행해야 할 작업을 추론한다. 추론된 작업은 에이전트 간에 커뮤니케이션을 통해서 수행해야 할 에이전트에게 전달된다. 둘째, 온톨로지 추론 엔진은 각각의 에이전트들이 가지고 있는 도메인 지식을 공유하고 재사용하기 위하여 존재한다. 그리고 에이전트들이 자동적이고 지능적인 기능을 수행하기 위해서 에이전트 도메인 지식베이스를 개념화하고 명세화한다. 그리고 에이전트 간에 이해할 수 있는 정보의 의미를 부여하고 정보간의 관계를 설정한다.

4.2 에이전트 사이의 커뮤니케이션 구조

유비쿼터스 환경에서의 서비스는 특정사물에 한정하지 않고 사람의 동적인 움직임에 따른 공간의 다른 사물을 이용하여 제공하게 된다. 즉, 사람의 생활을 중심으로 사람이 의식하지 않게끔 여러 에이전트가 존재하여 환경과 상호작용을 통한 컴퓨팅이 이루어지게 된다. 본 절에서는 유비쿼터스 서비스를 사용자 요구에 맞게 서비스하기 위하여 각 자원에 존재하는 에이전트들을 네트워크로 연결하여 플랫폼을 설계하고, 에이전트간의 메시지 교환방식을 설명한다. 에이전트 커뮤니케이션 구조는 그림 3과 같다.

그림 3에서와 같이 모든 자원은 네트워크로 연결되고, 유사 환경에 존재하는 자원들은 한 플랫폼으로 구성된다. 그리고 플랫폼 안의 에이전트 간에 공유할 수 있는 블랙보드(Black Board)가 존재한다. 블랙보드에는 에이전트들이 온톨로지를 기반으로 OWL 메시지를 주고받으며 처리한 인스턴스들이 저장된다. 예를 들어 사용자 이름, 사용자 소유 방 번호, 스케줄 정보를 저장해 놓는다. 블랙보드를 이용하면 정보의 재사용이 가능하고, 에이전트 간의 많은 메시지를 전송하면서 생기게 되는 부하를 줄일 수 있다.

4.2.1 플랫폼과 컨테이너

모든 자원은 네트워크 상에서 에이전트와 사용자를 이어주고 정보를 공유하는 플랫폼들로 구성된다. 플랫폼의 예로는 학교, 하나의 건물, 하나의 방 등을 들 수 있다. 하나의 플랫폼은 유사 환경을 고유공간으로 세분화하여 컨테이너로 구분되고 하나의 메인 컨테이너를 갖는다. 메인 컨테이너는 플랫폼에 존재하는 모든 컨테이너를 관리한다. 각 컨테이너에는 각기 다른 서비스를 하는 에이전트들이 존재하며, 각 에이전트들은 블랙보드의 지식(knowledge)을 공유하여 커뮤니케이션 한다.

본 시스템은 플랫폼을 기반으로 여러 컨테이너로 구분하고, 컨테이너에는 특정기능을 하는 에이전트들이 존재하게 된다. 이러한 기능을 위해서 메인 컨테이너에는 두 가지 요소가 존재한다. 첫 번째 요소는 'White Page'인 AMS(Agent Management Services)이다. AMS는 에이전트가 활동하면 ID를 할당하여 레지스터에 등록하고, 활동을 멈추면 레지스터에서 삭제한 후 ID를 소멸한다. 이와 같은 에이전트의 life-cycle 기능을 제공

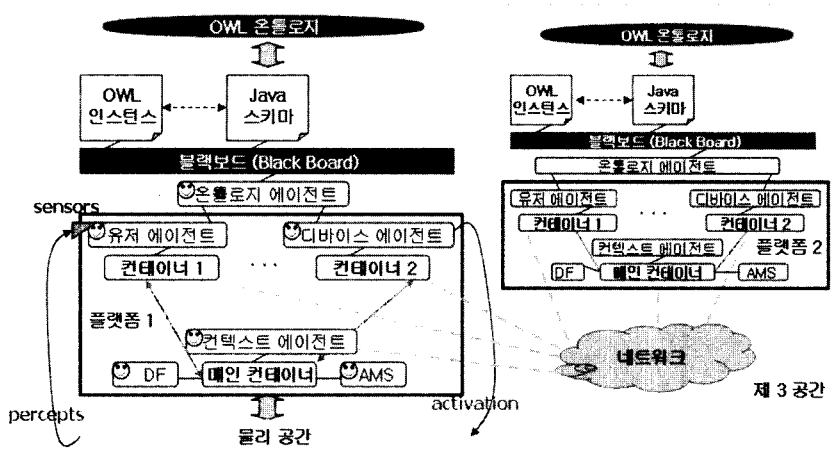

그림 3 시맨틱 에이전트 커뮤니케이션 구조

하는 요소가 AMS이다. 두 번째 요소는 'Yellow Page'인 DF(Directory Facilitator)이다. 자원의 서비스를 제공하는 에이전트의 서비스 타입과 서비스 이름, 서비스에 접근하기 위한 언어와 온톨로지 및 속성을 등록한다. 그리고 해당 서비스를 하는 에이전트가 제 3공간에 존재하게 되면, DF에 등록된 서비스 명세서를 참조하여 사용자가 원하는 서비스를 공급하는 에이전트가 있는지 검색한다. 이와 같이 디렉토리를 관리하는 기능을 제공하는 요소가 DF이다.

4.2.2 시맨틱 에이전트

본 연구에서는 컨텍스트 에이전트, 유저 에이전트, 디바이스 에이전트, 온톨로지 에이전트로 시맨틱 에이전트를 나눈다. 컨텍스트 에이전트는 메인 컨테이너에서 물리공간의 사용자 인증과 제 3공간의 에이전트와의 매개체 역할을 한다. 즉, 사람이 장소를 이동하여 특정 사물에 동적으로 변화하면, 수집된 정보를 가지고 지능적 의사 결정을 내리게 된다. 따라서 물리공간에 적용시켜야 할 서비스를 결정하고, 유저 에이전트로부터 사용자 서비스에 관련된 사용자가 선호하는 디바이스의 정보를 받아서 규칙(rule)을 만들어 관리한다. 그리고 프라이버시 규칙을 적용하여 서비스할 디바이스 에이전트에게 통보한다. 즉, 에이전트를 식별할 수 있도록 새로운 서비스의 기능을 가진 에이전트가 생성되면 AMS에 등록한다. 그리고 DF에 등록된 서비스 명세를 참조하여 원하는 서비스를 제공하는 에이전트를 찾는다. 검색된 에이전트에게 ACL Message Type(REQUEST, AGREE, INFORM, PROPOSE 등)에 따라서 메시지를 보내고 디바이스 에이전트를 작동한다. 더 이상 그 기능의 에이전트가 필요 없으면, AMS에 등록된 에이전트 정보를 삭제한다.

유저 에이전트는 사람을 추적하고 각 개인을 식별한다. 또한 사용자의 행동 및 그 상황에 맞는 다양한 정보에 근거하여 사용자를 대신하여 작업을 수행하고, 자동적으로 상황을 판단한 후 다양한 처리를 하게 된다. 즉, 메인 컨테이너의 컨텍스트 에이전트에 연결한 후, 센서로부터 입력이 들어오면 메인 컨테이너의 DF로부터 해당 서비스를 할 컨텍스트 에이전트의 AID(Agent ID)를 검색하여 탐지하고 커뮤니케이션 한다.

디바이스 에이전트는 장치(device)기능에 따라 컨텍스트 에이전트로부터 장치 작동 내역을 OWL 형식으로 전달받아 작동한다. 디바이스 에이전트는 OWL 파일을 분석하여 실행 가능한 데이타형식으로 전환하고 장치를 작동한다.

온톨로지 에이전트는 온톨로지를 이용하여 에이전트들이 주고받은 메시지의 내용으로부터 재사용이 가능한 메타데이타들을 블랙보드에 저장한다. 또한 OWL 인스턴스(instance)를 자바(java) 스키마(schema) 형식으로 변환하고, 역변환 하는 역할을 수행한다. 그리고 온톨로지의 추가적인 생성과 온톨로지 관리 및 사용자들의 프라이버시 보호를 위한 서비스 측면의 관계정보를 제공하게 된다.

5. OWL 컨텐트 메시지 전송을 위한 에이전트 간의 메시지 흐름

에이전트들은 ACL로 작성된 메시지를 MTP(Message Transport Protocol)를 이용하여 전달한다. 본 논문에서 제안한 에이전트 간의 메시지 전달은 물리공간으로부터 자원을 감지하여 에이전트들 간에 OWL 메시지 내용을 에이전트에게 통보하고 웹 서비스 하게 된다.

그림 4는 물리공간의 사람을 인식하고 사람이 요구하

는 서비스를 인지하여, 디바이스를 작동시키는 에이전트 간의 메시지 흐름 구조이다. 이 구조는 3장에서 언급한 시나리오에 적용해 볼 수 있다. 유저 에이전트가 301호에 들어온 학생("사람" 사용자)을 감지하고, 감지한 학생 에이전트로부터 학생의 정보(knowledge)를 알아내어, 컨텍스트 에이전트에게 웹 서비스를 받아도 되는 사람인지 인증을 요청한다. 컨텍스트 에이전트는 온톨로지 에이전트를 통하여 그 사람에게 부여할 권한과 정보, 서비스내역을 유저 에이전트에게 알려준다. 또한 회의 자료 파일을 스크린에 출력하게 될 교수님의 컴퓨터 디바이스 에이전트에게 출력할 파일의 URI와 교수실에서 이용할 수 있는 서비스를 OWL 인스턴스로 생성하여 전달한다. 디바이스 에이전트는 전송받은 OWL 인스턴스를 분석하여 장치를 작동하게 된다. 이와 같은 에이전트간의 메시지를 통해 장치가 스스로 적절한 서비스를 감지하여, 사람의 개입 없이 다른 서비스 제공자에게 장치 자신의 역할을 등록하고 자동 서비스 하게 된다.

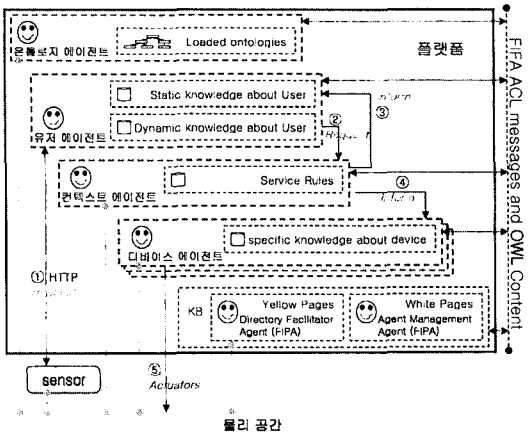

그림 4 시맨틱 에이전트 메시지 구조

6. 상태 인식을 위한 OWL 인스턴스 메시지 구조

6.1 인터페이스

한 컴퓨터에서 실행되던 프로그램을 에이전트들의 메시지 교환을 통해 다양한 플랫폼을 가로질러 다른 컴퓨터로 실행상태를 옮길 수 있다. 이러한 처리는 에이전트가 스스로 처리하게 되는데, 이전 컴퓨터의 에이전트로부터 OWL 인스턴스를 받아오면 된다. 온톨로지 에이전트는 OWL 인스턴스를 온톨로지 스키마구조를 통해서 컴퓨터 사용자의 이름, 위치정보, 프라이버시, 개인적 취향에 대한 정보를 분석한다. 따라서 사람이 한 장소에서 다른 장소로 이동하면, 다른 장소와 그 사람의 위치정보에 기입된 장소를 연결하는 에이전트가 존재하고, 이 에이전트로부터 OWL 인스턴스 메시지를 받아 처리하게 된다. 에이전트들은 일반적으로 특정 환경에서 일어나는 업무를 관리할 수 있는 프로그램이다. 따라서 방이나 복도를 걷는 사람이 가장 가까운 장치를 자동으로 이용하게 하는 기능들이 포함되어 있다.

6.2 OWL 컨텐츠 메시지 전송을 위한 온톨로지 구조

자동적 행위를 발생시키거나 행위에 대한 관계를 위해 OWL을 기반으로 온톨로지를 구축한다. OWL 속성(attribute)을 이용하여 인스턴스 간의 특징을 파악하고 사용자 프라이버시를 보장한다. 주요 온톨로지의 구성은 다음과 같다.

- physical 컨텍스트 온톨로지 : location, time, room
- environmental 컨텍스트 온톨로지 : light, sound
- device 컨텍스트 온톨로지 : media device 및 기타 device
- personal 컨텍스트 온톨로지 : person, schedule

위와 같은 컨텍스트 온톨로지를 OWL 온톨로지 언어로 표현한다. 또한 사용자와 유비쿼터스 디바이스 온톨로지를 계층적 구조로 설계하고, 이를 에이전트가 활용할 수 있도록 시스템을 설계하였다.

본 연구에서는 서비스하고자 하는 상황을 인지하고자 기존의 상황 인지 시스템보다 매우 구체적이고 풍부한 시맨틱 상황을 추론할 수 있는 온톨로지 기반 시스템을 구축하였다. OWL로 작성된 온톨로지 스키마를 위의 그림과 같이 java 클래스 형식으로 변환해서 java기반인 JADE로 구축된 에이전트들이 온톨로지를 기반으로 상황에 따른 인스턴스 생성이 가능하도록 하였다. 3장에서 언급한 시나리오를 예로서 살펴보면, 기존의 상황 정보에서는 학생이 301호에 들어왔다는 것만을 제공할 수 있지만 본 시스템의 시맨틱 상황 정보는 위의 온톨로지를 활용하여 학생이 자신의 방이 아닌 다른 공간에 들어왔다는 것을 추론할 수 있다. 따라서 사용자에 따른 차별화된 서비스 제공이 가능하다. 그리고 교수와의 관계를 파악하여 교수님 방에서는 교수가 설정한 범위 안에서의 디바이스만 작동 가능하게 된다. 또한 교수가 방에 없을 시 학생에게 제공되는 교수의 위치정보에도 제한을 두어 사용자의 프라이버시를 침해하지 않게 된다.

에이전트 간에 전송하게 되는 메시지 내용은 그림 5과 같이 OWL로 작성된 인스턴스를 전송하게 된다. 메시지를 전송받은 디바이스 에이전트는 온톨로지를 기반으로 메시지를 분석하여 장치를 작동하게 된다.

본 시스템은 Java class로 구성된 온톨로지를 이용하여 Person, Device, Operation등의 정보를 인식하고 수행된다. 모든 온톨로지 스키마가 정의된 Person-Ontology 파일을 통하여 에이전트 프로그램은 온톨로지

```
<?xml version='1.0'?>
<rdf:RDF xmlns='http://ailab.ssu.ac.kr/ontology/ubiProject/ontologies#'
         xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
         xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
         xmlns:owl='http://www.w3.org/2002/07/owl#'
         xml:base='http://ailab.ssu.ac.kr/ontology/ubiProject/ontologies#'>
<Person rdf:ID='choi'>
         <hasName>Jung-hwa </hasName>
</Person>
<Schedule rdf:ID='choi'>
         <byPerson><owl:Thing rdf:ID='Jung-hwa'/></byPerson>
         <duringTime rdf:resource='2'/>
         <hasContent rdf:resource='#uT Project Seminar'/>
</Schedule>
</rdf:RDF>
```

그림 5 디바이스 에이전트가 전송받은 OWL 인스턴스

기반의 OWL 메시지를 전달받게 된다. 온톨로지는 크게 Concept, Predicate, AgentAction 클래스로부터 상속받아 사용된다. Concept으로 정의된 스키마는 블랙보드에 들어갈 메타 데이타를 정의하고, 각 스키마의 인스턴스를 기능별로 구분하여 AgentAction 클래스에 스키마로 정의하고, 에이전트 프로그램이 동작되게 된다. Predicate 클래스로부터 상속받은 클래스는 에이전트의 동작을 명시하게 된다.

6.3 시스템 동작의 원리

본 시스템은 강력한 보안 기능과 풍부한 네트워크 관련 기능이 탑재되어있는 Java를 기반으로, 특정 OS에 의존하지 않고 어떤 플랫폼에서도 애플리케이션을 실행할 수 있는 JADE(Java Agent Development Framework)를 사용하였다. JADE는 다중 에이전트 시스템 개발에 유용한 미들웨어(middleware)로서 FIPA 표준에 따른 지능형 에이전트를 구현하기 위한 소프트웨어 개발 프레임워크이다[10]. 본 시스템은 이 플랫폼을 이용하여 에이전트간의 네트워크가 형성되게 된다. 그리고 디바이스 간에 상호작용을 위해서 FIPA 표준 온톨로지 언어인 OWL을 사용하여 에이전트 간에 메시지를 전송한다.

3장에서 언급한 시나리오를 통해서 에이전트간의 메시지 전달에 의한 시스템 동작을 살펴보겠다. 유비쿼터스 네트워크로 연결된 플랫폼은 학교 또는 각 건물이고, 건물의 교실, 연구실, 교수실, 화장실 등은 컨테이너로 구분된다. 이 중 하나를 메인 컨테이너로 하고, AMS, DF, MTP기능을 포함하여 에이전트를 관리하고 적절한 서비스를 부여하게 된다. 유비쿼터스 플랫폼이 형성되면, 메인 컨테이너에는 플랫폼 안의 에이전트와 AMS, DF, MTP 서비스를 연계할 컨텍스트 에이전트가 생성된다. 그리고 각 컨테이너에 사람을 감지하고 요구사항을 분석할 유저 에이전트와, 컨테이너 안에 존재하는 장치마다 디바이스 에이전트들이 생성된다.

그림 6은 OWL 사람이 감지되어 서비스가 제공되기까지의 흐름이다. 에이전트는 JADE API를 이용하여 생성한 프로그램이며, OWL로 구성된 온톨로지를 JADE에서 제공하는 'Ontology Bean Generator'를 이용하여 생성한 Java Class를 이용하여 지식을 공유한다. 그래서 301호 교수실에 학생이 들어오면, 유저 에이전트가 학생에게 제공할 서비스를 알기 위해서 학생의 ID를 메인 컨테이너의 컨텍스트 에이전트에게 보내게 된다. 컨텍스트 에이전트는 온톨로지 에이전트를 통해서 학생에 대한 Person과 Location 스키마의 메타 데이타를 기반으로 213호의 학생 에이전트와 통신을 하게 된다. 따라서 온톨로지 정보와 학생 에이전트가 보유한 학생 스케줄의 메타 데이타를 받아 오게 되는 것이다. 이런 메타 데이타의 전달은 에이전트 통신 언어인 ACL을 사용하고, FIPA에서 제공하는 메시지 타입(REQUEST, AGREE, INFORM 등)에 따라 에이전트들이 반응한다. 메시지 내용은 본 시스템에서 구성한 OWL 온톨로지 스키마를 기반으로 한 OWL 인스턴스를 포함하게 된다.

에이전트가 다른 에이전트로부터 ACL 메시지를 받게 되면, ACL 메시지의 OWL로 작성된 내용(content)만을 추출한다. 추출한 OWL 인스턴스를 Jena를 사용하여 기계가 추론할 수 있는 FOL(First Order Logic)을 기반으로한 KIF(Knowledge Interchange Format)[11]형식으로 변환한다. 변환한 KIF 형식의 인스턴스를 XSB Prolog를 사용하여 사용자 메타 데이타를 통하여 제공할 서비스를 예측한다. 이때 학생 스케줄을 받은 컨텍스트 에이전트는 회의에 필요한 자료들을 교수실 컴퓨터 에이전트에게 ACL 메시지로 보내게 된다. 그리고 교수실의 빔 프로젝터가 있음을 인식하고, 빔 프로젝터 에이전트에게 스크린에 투사할 파일을 메시지로 통보한다. 또한 파일을 스크린에 투사하기 위해서 주변 환경이 어두워야 함을 인식하고, 교수실 형광등 에이전트에게 소등하라는 메시지를 보내게 된다. 이러한 서비스는

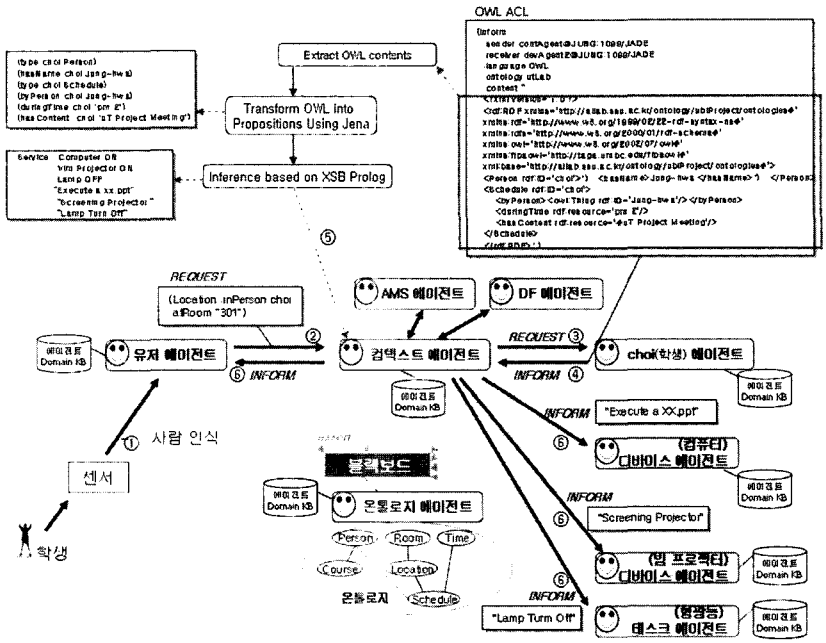

그림 6 OWL ACL 추론

Prolog로 작성한 OWL에서 제공하는 속성들의 관계 규칙을 기반으로 예측하게 된다. 이러한 메시지 전달을 통해서 유비쿼터스 환경에 제공되는 서비스가 결정된다.

본 시스템은 자료 공유를 대용량의 데이타베이스를 이용하는 것이 아니라 에이전트를 기능별로 구분하여 지식을 보유하고 메시지를 통해 공유함으로써, 더 정확하고 빠른 서비스제공을 목적으로 하였다.

6.4 구 현

그림 7,8은 본 시스템에서 제안한 다중 에이전트 커뮤니케이션 프레임워크의 시뮬레이션 화면이다. 3장의 시나리오를 기반으로 메인 컨테이너의 컨텍스트 에이전트와 301호의 유저 에이전트간의 커뮤니케이션을 통해서 디바이스 에이전트에게 서비스를 알리는 화면이다.

본 연구는 한 공간에 두 명의 사용자가 들어온 경우, 같은 상황에서 다른 서비스를 추론하도록 하였다. 두 명의 사용자는 서비스 공간의 소유자인 교수와 uT Project 회의를 위해서 현 공간에 들어온다. 교수 에이전트는 두 사용자와 관련되는 현재 시간의 교수 일정을 확인하고 현 공간에서 사용자들에게 제공할 서비스를 추론한다. uT Project는 Project1과 Project2로 나뉜다. choi(학생)는 Project1에 속하고 lee(학생)는 Project2에 속한다. 에이전트간의 통신을 통한 서비스 추론 후, 교수 에이전트는 회의에 필요한 파일 URI를 알기 위해서 choi 에이전트와 lee 에이전트를 연결하여 커뮤니케이션 한다.

그림 7은 301호에 id001이 들어오면 유저 에이전트는 사용자 아이디를 메인 컨테이너의 컨텍스트 에이전트에게 보내고 서비스 받을 사용자인지 인증절차를 거친다. 유효한 사용자라면 컨텍스트 에이전트는 사용자 소유의 방에 있는 개인(choi) 에이전트를 연결하고, 사용자가 301호에 들어온 목적을 알게 된다. 그리고 301호 소유주와의 관계를 파악하여 301호 소유주의 프라이버시를 해치지 않는 범위에서 서비스를 제공한다. 컨텍스트 에이전트는 추론 엔진을 통하여 서비스를 추론 한 후, 디바이스 에이전트에게 서비스를 통보한다. 디바이스 에이전트는 서비스를 기능별로 구분하여 해당 서비스를 제공하는 에이전트에게 통보하게 된다. 이 때, 에이전트간의 커뮤니케이션은 사용자가 자신의 소유가 아닌 다른 공간에 존재할 때의 상황 인지를 통한 서비스 제공을 시뮬레이션 하도록 하였다. 그림 8은 id002에 대한 서비스 추론 과정이다. 그림 7과 같은 상황에서 301호 유저 에이전트는 소유자와 lee와의 관계를 파악하고 다른 서비스를 추론함을 볼 수 있다.

본 연구는 유비쿼터스 환경에서의 시나리오를 정하고 일어날 수 있는 상황을 스크립트 파일로 구성하였다. 스크립트 파일로 메뉴를 구성하고 사용자가 메뉴 선택 시, 온톨로지 기반의 추론엔진과 다중 에이전트 간의 통신으로 상황에 따른 서비스 메시지를 출력한다. 한번 추론된 정보들은 블랙보드에 저장되어 에이전트들이 공유하고, 사용자별 개인 에이전트들의 지식을 통해서 상황이

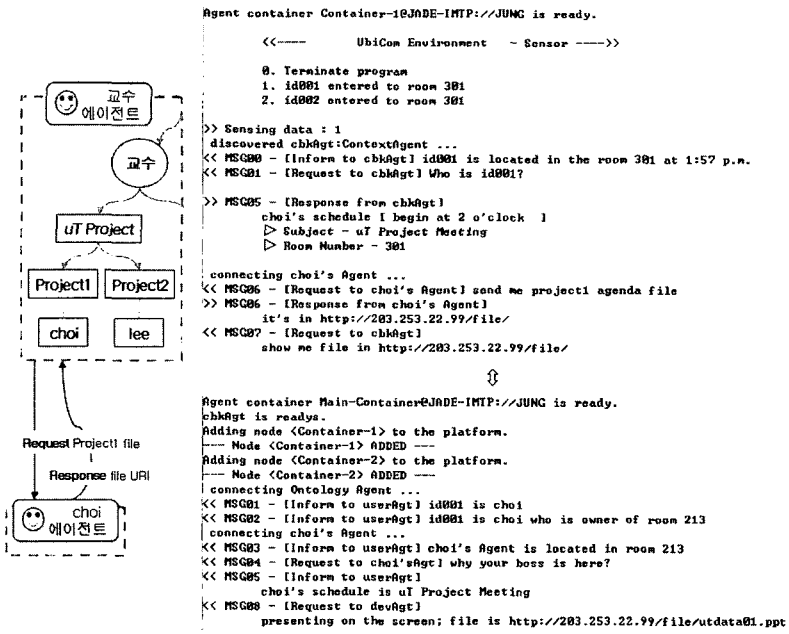

그림 7 id001이 301호에 들어온 경우
301호 유저 에이전트(상)와 컨텍스트 에이전트(하) 간의 커뮤니케이션

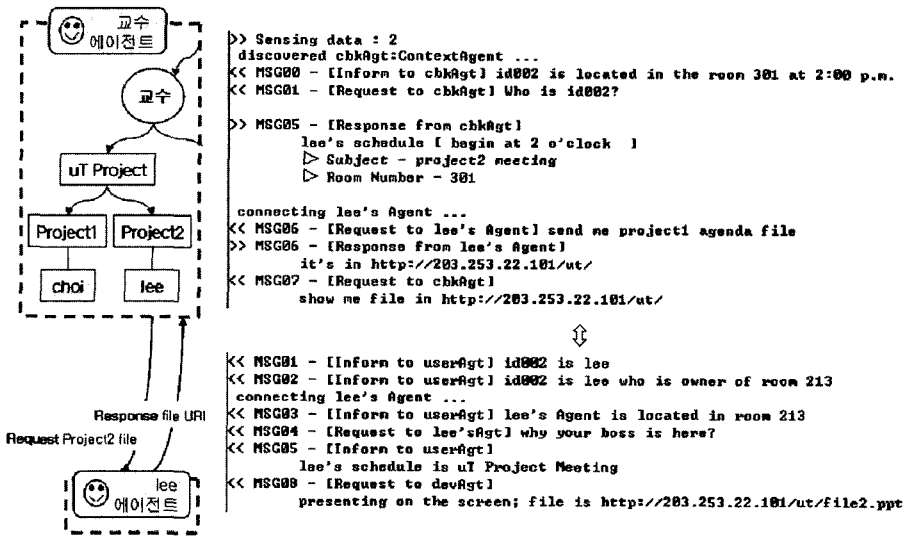

그림 8 id002가 301호에 들어온 경우
301호 유저 에이전트(상)와 컨텍스트 에이전트(하) 간의 커뮤니케이션

같더라도 사용자에 따른 다른 서비스를 추론하게 된다.

7. 결론 및 향후 연구 과제

본 논문은 지금까지 개발된 센서 네트워크 기술을 토대로 인간과 에이전트와의 상호작용 및 에이전트 스스로 인간이 원하는 서비스를 처리하고 제공하기 위한 커뮤니케이션 구조를 제안하였다. 물리공간의 사람이라고 하는 자원이 장소를 이동하여 특정 사물에 관하여 동적으로 변화하는 정보를 제공받도록 함으로써, 물리적 공간과 에이전트가 존재하는 제3의 공간이 통합되어 유비

쿼터스 서비스를 가능하게 한다. 본 논문에서는 유비쿼터스 환경에서 사용자의 행위를 분석하여 요구되는 서비스를 제공하기 위해서 시맨틱 웹의 메타 데이타 개념을 통하여 웹 문서에 시맨틱 정보를 덧붙이고, 에이전트가 시맨틱 정보를 통해서 사용자 요구를 분석하고 추론하여 서비스 할 수 있도록 하였다. 또한 모든 공간의 디바이스를 공유함으로써 침해될 수 있는 프라이버시 문제를 온톨로지에 표현한 속성과 관계에 의하여 자원의 공유를 제한하였다. 그리고 에이전트 간에 공유하는 지식정보를 블랙보드에 추가하여 커뮤니케이션 함으로써, 사용자에게 제공해야 할 서비스 예측의 정확성을 높이도록 하였다.

앞으로, 한 개인이나 다른 여러 사용자, 다른 에이전트들의 행동 패턴들로부터 학습(learning)을 통하여 계획(planning), 추론(reasoning)기능을 추가한 에이전트 간의 메시지 분석 연구를 계속 수행해 나갈 것이다. 이때, 블랙보드 상에 존재하는 메타데이타가 아닌 사용자의 지속적인 관심을 이용함으로써 유비쿼터스 환경에서의 지능적인 서비스에 더 다가가고자 한다.

참 고 문 헌

[1] W. Edwards and R. Grinter, "At home with ubiquitous computing: seven challenges," Ubicomp 2001: Ubiquitous Computing, LNCS 2201, pp. 256-272, 2001.

[2] Michael Schalk, Thorsten Liebig, Torsten illmann, Frank Kargl, "Combining FIPA ACL With DAML+OIL - A CASE Study," OAS2002 Workshop Proceedings, 2002.

[3] Tim Berners-Lee, James Hendler and Ora Lassila, "The Semantic Web," Scientific American, May 2001.

[4] Jeff Heflin. "Web Ontology Language (OWL) Use Cases and Requirements," W3C Candidate Recommendation 18 August 2003.

[5] HP Cooltown project, http://www.cooltown.com.

[6] W. Chan, "Using CoolBase to Build Ubiquitous Computing Applications," HP Technical Report, HPL-2001-215, 2001.

[7] R. Scott Cost, Tim Finin, Anupam Joshi, Yun Peng, Charles Nicholas, Harry Chen, Lalana, Filip Perich, Youyong Zou, Sovrin Tolia, and Ian Soboroff, "ITtalks: A Case Study in the Semantic Web and DAML+OIL," IEEE Intelligent Systems, 17(1), pp. 40-46, 2002.

[8] "FIPA SL Content Language Specification," Foundation for Intelligent Physical Agents, 2001, http://www.fipa.org/specs/fipa00008/XC00008D.html.

[9] Kostantinos Sagonas, et. al., "XSB as an Efficient Deductive Database Engine," In ACM Conference on Management of Data (SIGMOD), pp. 442-453, 1994.

[10] Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, "JADE - FIPA-compliant agent framework," In Proceedings of PAAM'99, pp.97-108, 1999.

[11] Michael R. Genesereth, "Knowledge Interchange Format," draft proposed American National Standard (dpANS), NCITS.T2/98-004, http://logic.stanford.edu/kif/dpans.html

최 정 화

2004년 숭실대학교 컴퓨터학부(학사) 2004년~현재 숭실대학교 대학원 컴퓨터학과 석사과정. 관심분야는 유비쿼터스 컴퓨팅, 시맨틱 웹, 온톨로지 추론, 다중 에이전트 시스템

박 영 택

1978년 서울대학교 전자공학과(학사) 1980년 KAIST 전산학(석사). 1992년 Univ. of Illinois at Urbana-Champaign(박사). 1981년~현재 숭실대학교 컴퓨터학과 교수. 관심분야는 인공지능, 에이전트, 전문가 시스템