

URC를 위한 상황 정보 관리 기술

Context Management Technology for Ubiquitous Robotic Companion

지능형 로봇 특집

홍충성 (C.S. Hong)	지능형로봇서버연구팀 연구원
조준면 (J.M. Cho)	지능형로봇서버연구팀 선임연구원
이강우 (K.W. Lee)	지능형로봇서버연구팀 선임연구원
서영호 (Y.H. Seo)	지능형로봇서버연구팀 선임연구원
김형선 (H.S. Kim)	지능형로봇서버연구팀 책임연구원
김 현 (H. Kim)	지능형로봇서버연구팀 팀장

목 차

-
- I. 서론
 - II. 상황 인식 컴퓨팅 기술 동향
 - III. CAMUS 상황 정보 모델
 - IV. 결론

URC는 언제 어디서나 나와 함께 하며 나에게 필요한 서비스를 제공하는 로봇이라는 의미의 새로운 네트워크 기반의 지능형 서비스 로봇 패러다임이다. URC 환경에서 로봇은 사용자의 현재 상황을 이해한 뒤 적절한 정보와 서비스를 지능적이고 능동적으로 제공할 수 있어야 하며, 이를 위하여 상황 인식 시스템의 개발이 필요하다. 상황 인식 시스템의 핵심 기능 중의 하나는 상황 인식 응용에서 이용 가능한 사용자의 주변 환경에 관한 정보를 표현하기 위한 형식적인 상황 정보 모델 및 이를 관리하고 이용하는 방법을 제공하는 데 있다. 본 고에서는 상황 인식 컴퓨팅의 기술 동향을 상황 정보 모델 중심으로 소개하고, URC를 위한 상황 인식 미들웨어인 CAMUS의 상황 정보 모델 및 상황 정보 관리 모듈에 관하여 소개한다.

I. 서론

정보기기의 소규모화, 고성능화, 가격 하락과 더불어 네트워크 서비스의 질적 향상은 사용자가 언제 어디서나 정보에 접근하고 서비스 받을 수 있는 정보통신 환경을 마련해 주었다. 이러한 기술의 발전은 로봇 분야에도 영향을 미치고 있다. 기존의 산업용 로봇 시장이 포화 상태에 이르면서 새로운 시장 창출이 요구되었고, 이를 위하여 서비스 로봇에 대한 연구가 시작되어 현재 활발히 진행되고 있다. 서비스 로봇이 가전기구나 정보 단말기와 같이 일상 생활 속에 들어오기 위해서는 사용자가 저가로 쉽게 로봇을 구입하여 다양한 서비스를 받을 수 있도록 되어야 한다. 하지만 현재의 서비스 로봇은 가격에 비해 제공하는 기능과 서비스가 단순할 뿐만 아니라 로봇 자체의 분명한 킬러 응용(killer application)을 갖고 있지 않다.

이러한 배경 하에서 정보통신부에서는 로봇의 가격은 낮추는 반면 그 기능과 서비스는 다양화될 수 있도록 하기 위하여 “언제 어디서나 나와 함께 하며 나에게 필요한 서비스를 제공하는 로봇”을 의미하는 URC라는 새로운 네트워크 기반의 지능형 서비스 로봇 패러다임을 제시하고 있다. 기존 로봇에 네트워크 및 정보 기술 인프라를 최대한 활용함으로써, 단품 로봇과의 차별성과 더불어 다양한 서비스를 제공하는 것이 목표이다.

상황 인식은 URC 서비스에서 수행되어야 할 가장 중요한 기능 중의 하나이다. URC에서 지능형 서비스 로봇은, 사용자가 명시적으로 정보나 서비스를 요청할 때 현재의 상황을 이해해서 그 상황에 맞는 서비스를 제공할 뿐만 아니라 사용자의 명시적인 요청이 없는 경우에도 필요한 시점에 적절한 정보와 서비스를 지능적이고 능동적으로 제공할 수 있어야 한다. 이 때 상황 인식 시스템은 사용자의 주변 환경과 컴퓨팅 장치들의 특성에 관한 정보를 획득하고, 해석할 뿐만 아니라 사용자가 원하는 정보와 서비스를 제공하는 시스템을 의미한다.

상황 인식 시스템의 핵심 기능 중의 하나는 상황

인식 응용에서 이용 가능한 사용자의 주변 환경에 관한 정보를 표현하기 위한 형식적인 상황 정보 모델 및 이를 관리하고 이용하는 방법을 제공하는 것이다. 상황 인식 시스템에서는 다양한 상황 인식 응용 개발을 지원하기 위하여 상황 정보의 표현 능력, 표현 형식성의 수준, 사용 및 적용 편리성, 추론을 통한 높은 수준의 상황 정보 해석 지원 등을 고려하여 상황 정보 모델을 개발하고 제공해야 한다.

본 고에서는 네트워크 기반의 지능형 서비스 로봇을 위한 URC 환경을 구현하기 위한 필수 요소인 상황 인식 컴퓨팅의 기술 동향을 상황 정보 모델 중심으로 소개한다. 또한 URC 환경을 위한 상황 인식 미들웨어인 CAMUS[1]의 상황 정보 모델 및 상황 정보 관리 모듈에 관하여 소개한다.

II. 상황 인식 컴퓨팅 기술 동향

1. 상황 정보의 정의

최초의 상황 인식 시스템인 “Active Badge Location System”[2]과 같은 초기의 상황 인식 시스템들은 주로 사용자의 위치를 상황 정보로 이용하였다. 초기의 상황 정보 시스템들은 파악된 사용자의 현재 위치를 기반으로 적절한 정보를 사용자에게 제공함으로써 상황 인식 컴퓨팅 환경을 구현하고자 하였다.

상황 인식 컴퓨팅 환경에서는 사용자의 위치 이외에도 다양한 상황 정보들이 존재하고 있다. 많은 연구자들은 이러한 상황 정보를 이용하기 위하여 다양한 상황 정보를 열거시킴으로써 상황 정보에 관한 정의를 내리고자 노력하였다. 그러나 이러한 정의들은 각각의 시스템에서 사용된 상황 정보의 종류를 묘사하고는 있지만, 사용자의 의도나 프로파일과 같이 응용 분야에 따라 추가될 수 있는 다른 종류의 상황 정보는 고려하지 못하고 있다. 이러한 문제를 해결하기 위하여, Dey는 상황 정보의 정의는 형식적(formal)이며 일반적(general)이어야 한다고 주장하였으며, 그의 연구에서 상황 정보란 특정 엔티티

(entity)의 상황을 묘사할 수 있는 모든 정보이며, 이때 엔티티는 사용자와 응용 사이의 상호작용에 관련 되는 모든 객체를 포함한다고 정의하였다[3].

2. 상황 인식 응용

상황 인식 컴퓨팅에서 상황 정보를 사용하는 방법은 크게 두 가지로 나눌 수 있다. 첫번째는 상황 인식 응용에서 상황 정보를 이용하여 제공할 서비스를 선택하는 것이며, 두번째는 제공될 서비스를 상황에 맞게 적응시키는 것이다.

상황 인식 컴퓨팅 환경에서 상황 정보는 사용자와 환경의 변화를 감지하여 자동적으로 발생하거나 사용자 혹은 응용의 필요에 의해 추출된다. 초기의 상황 인식 응용은 사무실 환경 내에서 사용자의 위치와 식별자를 이용하여 적절한 서비스를 수행하는데 초점이 맞추어졌다. 최초의 상황 인식 응용으로 알려진 ORL에 의해 수행된 call forwarding 응용은 액티브 배지(active badge) 시스템을 기반으로 사용자의 현재 위치를 파악한 뒤, 사용자에게 걸려오는 전화를 가장 가까운 곳에 위치한 전화기로 연결해 주는 서비스를 제공하였다[4]. 사무실 환경 내에 적용되던 상황 인식 응용은 이후 AT&T Bell 연구소에서 개발한 shopping assistant 응용[5]과 같이 쇼핑몰, 박물관, 전시장 등 다양한 장소에 적용되었다.

사무실, 박물관과 같은 가정 외부에서 주로 적용되던 상황 인식 응용은 상황 정보를 이용한 가전 제어 및 온도, 습도 조절과 같은 거주 환경 조절을 목적으로 AwareHome 프로젝트[6]와 EasyLiving 프로젝트[7] 같이 가정 내부에 적용되기도 하였다.

장소 및 사용자 인식과 같은 비교적 제한적인 상황 정보만을 이용하는 초기의 상황 인식 응용과는 다르게, 1990년대 후반부터는 다양한 상황 정보를 조합하여 지능적인 서비스를 제공하는 상황 인식 응용들이 발표되기 시작하였다. Schmidt 등은 TEA 프로젝트에서 사용자의 위치가 아닌, 사용자의 행동과 빛의 크기 등의 주변 환경에 관한 정보를 바탕으로 PDA에서 보여지는 글자의 크기가 적절하게 변

화되는 응용과 사용자의 현재 행동 및 주변 상황을 고려하여 모바일 폰의 벨 종류 및 소리의 크기를 능동적으로 변화시키는 응용을 개발하였다[8].

상황 인식 응용에서 복잡한 상황 정보를 이용하여 지능적인 서비스를 제공하고자 하는 노력은 다양한 스마트 공간을 구현하고자 하는 노력으로 이어지고 있다. MIT의 인공 지능 연구실에서 수행하고 있는 Oxygen 프로젝트에서는 분산 에이전트 로직(logic)을 사용하여 사용자의 음성 인식을 포함한 다양한 상황 인식을 통한 지능형 서비스를 제공할 수 있는 공간(agent-based intelligent reactive environments)을 만들었으며, Stanford University에서는 PDA와 빔 프로젝터와 같은 다양한 정보 장치들 간의 상호 운용성(inter-operability)을 높일 수 있는 스마트 작업 공간을 제공함으로써 사용자 사이의 정보 공유를 보다 효율적으로 할 수 있는 응용을 개발하였다[9],[10].

최근 상황 인식 컴퓨팅에서 가장 활발하게 개발되고 있는 스마트 공간으로는 지능형 회의실(intelligent meeting room)과 스마트 홈(smart home)이 있다. 지능형 회의실에서 회의 도우미 응용은 회의 일정에 따라 회의 참석자들의 위치를 파악한 후, 그들이 지니고 있는 이동 장치를 통하여 회의 내용 및 진행 상황을 알려주고 회의 진행 상태를 파악하여 다양한 지능형 회의실 내의 장치들을 조정함으로써 회의 진행을 도와주는 응용이다.

3. 상황 정보 모델

가. 개요

상황 인식 컴퓨팅 환경에서 응용들은 상황 정보 모델을 기반으로 개발되고 실행된다. 상황 인식 시스템에서는 상황 정보를 가공하고, 저장하며, 공유하기 위한 효율적이며 통합된 상황 정보 모델을 필수적으로 제공할 수 있어야 한다. 이를 위하여, 상황 인식 시스템은 적절한 상황 정보 모델을 구성하고 이를 관리할 수 있는 기능과 능동적이고 지능적인 서비스를 위하여 상황 정보 모델에 명시적으로 표현

되지 않은 암묵적인 상황 정보를 추론할 수 있는 기능을 제공해야 한다. 기존의 상황 인식 시스템에서 제공하는 상황 정보 모델은 데이터 구조 및 응용과의 관계를 고려하여 세 가지로 분류할 수 있다.

첫번째는 상황 인식 응용 중심의 상황 정보 모델링 접근 방법이다. 초기의 상황 인식 시스템은 특정 응용들을 위하여 상황 정보를 표현하고 모델링하였다. 이러한 상황 정보 모델은 일반적으로 응용에 의존적일 뿐만 아니라 표현 방식에 있어 형식성이 결여되게 되는데, Context Toolkit[3]의 key-value의 튜플은 대표적인 응용 중심의 모델링 접근 방법의 예이다.

두번째 상황 정보 모델링 방법은 모델 중심의 접근 방법으로 상황 정보를 표현하기 위하여 개념적인 모델링 접근 방법을 사용한다. 관계 데이터베이스로 쉽게 구현이 가능한 ER 모델을 기반으로 하는 형식적 상황 정보 모델이 대표적인 예이다. Henriksen 등은 상황 정보뿐만 아니라 부가적인 특징들을 ER과 UML 다이어그램을 이용하여 모델링 하였으며 [11], 이 모델은 확장된 ORM을 이용하여 다시 재구성되기도 하였다. 모델 중심의 접근 방법은 상황 정보 모델이 지식 추론을 쉽게 할 수 있는 기반을 제공하지 못하기 때문에 상황 인식 응용들이 추론을 통한 높은 수준의 상황 정보 해석을 하는 데 어려움을 겪게 되었다.

이러한 단점을 극복하기 위하여 최근에는 상황 정보 모델을 온톨로지와 같이 엄밀한(formal) 지식 표현 스킴(scheme)을 사용하여 표현하는 접근 방법이 사용되고 있다. 온톨로지 기반 접근 방법은 분산된 시스템 간의 지식 공유의 목적을 달성하기 위

하여 특정 도메인에서 상황 정보를 표현하는 온톨로지를 모델링하는 것에 주로 초점을 맞추고 있다. Ranganathan 등은 DAML+OIL을 이용하여 일차 술어식(first-order predicate)으로 표현된 상황 정보 모델을 스마트 공간 상의 상황 인식 에이전트들에게 제공하는 분산 미들웨어 기반구조인 Gaia 시스템에서 개발하였다[12]. Chen 등은 지능형 회의실 환경에서 수행되는 브로커 중심의 상황 인식 에이전트 기반 구조인 CoBrA 시스템을 개발하였다 [13]. CoBrA 시스템에서는 유비쿼터스 에이전트들을 지원하기 위한 OWL 기반의 상황 정보 온톨로지가 정의되었다. 또한 Chen 등은 다양한 응용 개발을 위한 표준 온톨로지인 SOUPA를 정의하였다[14]. Wang 등은 SOCAM 기반구조에서 상황 정보 의미 표현, 상황 정보 추론 및 공유, 상황 정보 분류, 상황 정보 종속, 그리고 상황 정보 품질 표현과 관련된 문제들을 해결하기 위하여 OWL을 이용한 엄밀한 상황 정보 모델을 제시하였다[15].

나. 온톨로지 기반 상황 정보 모델

사물의 존재론적 의미, 즉 ‘존재들과 존재들의 속성들에 관한 설명’을 기술하기 위한 온톨로지는 정보 과학 분야에서 특정 영역의 용어들과 그들간의 관계를 명시적으로 기술하기 위하여 사용된다. 온톨로지는 타 도메인의 지식을 재사용하고, 도메인 지식의 상속 및 부분 지식의 결합, 그리고 컴퓨터를 이용한 방대한 지식베이스에 대한 정형화된 분석을 할 수 있게 해준다.

온톨로지를 이용하여 상황 정보 모델을 기술하는 장점은 다음과 같다.

● 용어해설 ●

온톨로지: 공유된 개념화(shared conceptualization)에 대해 명시적이고 정형화된 명세(formal and explicit specification)로 정의되는 온톨로지는 시맨틱 웹의 등장으로 새롭게 주목받고 있다. 온톨로지는 일반적으로 개념(concepts), 관계(relations), 개념의 계층(concept hierarchy), 관계를 통한 개념과 개념의 연결로서의 함수(function), 공리(axioms) 등으로 구성된다.

● 용어해설 ●

OWL: OWL (Web Ontology Language)은 풍부한 어휘(vocabulary)와 형식적 의미론(formal semantics)을 포함하고 있기 때문에 기계 해석이 가능한 웹 콘텐츠를 저작하는 데 있어 XML, RDF 및 RDF 스키마(RDF-S)보다 뛰어나다. OWL은 표현력 및 추론 기능의 차이에 따라 OWL Lite, DL, Full과 같은 서로 다른 세 개의 하위 언어로 구성된다.

- 표현력 증가: 온톨로지 언어는 객체 기반 모델링 접근 방법을 이용하고 있다. 따라서 다양한 종류의 상황 정보의 특성들을 획득하고 이를 표현하기 위한 충분한 표현력을 가지게 된다.
- 상황 지식 공유 가능: 상황 정보 표현을 위한 온톨로지의 사용은 상황 인식 컴퓨팅 환경에 존재하는 에이전트들 혹은 서비스들과 같은 컴퓨팅 엔티티들이 상호 작용하는 과정에서 공통의 개념을 가질 수 있게 도와줄 뿐만 아니라 개념에 관한 의미를 이해할 수 있도록 도와준다.
- 추론: 온톨로지를 기반으로 할 경우, 상황 인식 시스템은 다양한 로직 기반의 추론 메커니즘을 사용할 수 있게 된다. 원시 상황 정보 혹은 이들로 부터 해석된 낮은 추상화 수준의 상황 정보로부터 높은 수준의 상황 정보를 유도해 낼 수 있으며, 이렇게 얻어진 상황 정보들은 학습이나 추천 등과 같은 다양한 서비스를 제공할 수 있도록 도와준다.
- 지식 재사용: 서로 다른 도메인의 잘 정의된 온톨로지를 재사용할 경우, 큰 비용의 지불 없이

큰 규모의 상황 정보 온톨로지를 쉽고 효율적으로 구축할 수 있다.

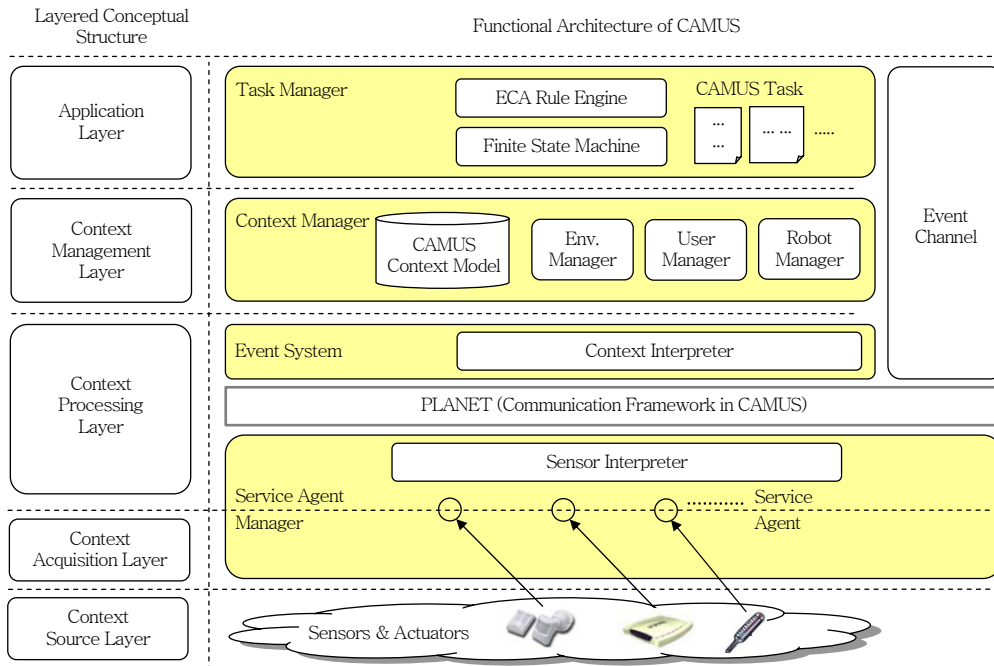
- 확장성: 상황 정보 온톨로지의 개념은 일반적으로 용어 혹은 용어들의 계층 구조들로 많이 구성된다. 따라서 새로운 상황 정보 온톨로지들은 계층적인 방법들을 이용하여 추가 및 확장이 가능하다.

III. CAMUS 상황 정보 모델

1. CAMUS 개요

ETRI에서 개발하고 있는 CAMUS는 URC 환경 내에서 획득된 상황 정보를 기반으로 환경 내에 있는 사용자에게 적절한 서비스를 제공할 수 있도록 상황 기반 응용의 개발과 실행을 지원하는 상황 인식 미들웨어이다[1].

CAMUS의 관점에서의 실세계(real world)는 물리공간(physical space)과 가상공간(cyber space)으로 구성된다. 물리공간은 실제적인 물리객체(phy-



(그림 1) CAMUS 기능적 구조도

sical object)들로 구성되어 있으며, 사람은 이러한 물리객체와의 상호작용을 통해 일을 한다. 가상공간은 물리공간을 추상화한 것으로서, 물리객체들을 매핑한 전산 자원(electronic resource)들로 모델링된다. 물리공간과 가상공간은 매핑된 수단을 통해 상호작용이 가능하다. CAMUS의 기능적인 구조는 (그림 1)과 같다.

서비스 에이전트 관리자(service agent manager)는 물리 공간의 센서를 가상 공간으로 매핑하고, 이들 센서 정보로부터 상황 정보를 추출하여 상황 관리자와 작업 관리자에게 제공함으로써 상황 기반 응용, 즉 CAMUS 태스크(task)가 능동적으로 서비스를 제공할 수 있도록 지원한다. 이 계층에서는 상황 정보의 추출뿐만 아니라 해석, 타임 스탬핑, 필터링 등을 통한 상황 정보의 가공도 이루어진다. 또한 서비스 에이전트 관리자는 로봇 플랫폼, 가전기기 등의 장치 제어, 그리고 음성/영상 처리, 일정 관리와 같이 CAMUS 태스크가 이용할 서비스들에 대한 인터페이스와 이들의 구현 코드를 관리한다. 이 계층은 CAMUS 태스크에 의해 요구되는 서비스를 탐색하고 실제 구현 모듈을 호출하는 역할도 한다.

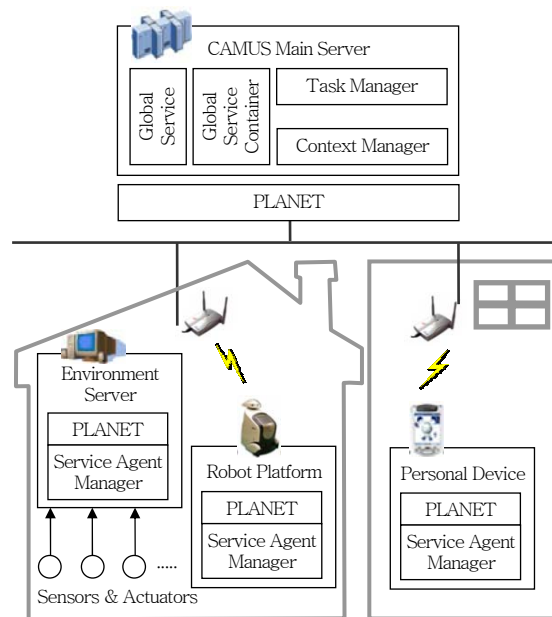
이벤트 시스템(event system)은 네트워크를 통해 물리적으로 분산된 환경으로부터 생성되는 다양한 이벤트를 관리하고 CAMUS의 각 구성 요소들 사이의 메시지 교환을 담당한다. 특히, 분산된 여러 환경에 설치된 센서로부터 생성되어 전달되는 이벤트를 사전에 기술된 객체로 전달하여 상황 관리자가 상황 정보 모델을 갱신할 수 있도록 하고, 상황 기반 응용이 상황 변화를 인지하여 사용자에게 능동적이고 지능적인 서비스를 제공할 수 있도록 한다.

상황 정보 관리자(context manager) 계층은 서비스 에이전트 관리자로부터 이벤트를 통해 전달된 상황 정보와 이 정보를 바탕으로 암묵적인 지식을 추론하여 저장하고 관리한다. 이 계층에서 관리하는 상황 지식은 추후 CAMUS 태스크가 실행될 때 참조된다. 따라서 상황 정보 관리자는 상황 지식 표현을 위한 상황 정보 모델과 함께 상황 정보를 추가하거나 수정하는 기능, 상황 지식 검색 기능을 제공한다.

작업 관리자(task manager) 계층은 개별 CAMUS 태스크를 기동시키고 수행중인 태스크 프로세스를 관리 또는 제어하는 역할을 한다. 태스크는 다시 다수의 작업 규칙으로 구성된다. 각 작업 규칙은 ECA의 표현으로, 이벤트와 조건을 통해 환경 및 상황의 변화와 사용자 또는 응용의 요구를 기술하고, 해당 상황이 발생될 때 수행할 서비스를 행동으로 기술한다. ECA 작업 규칙이 수행될 때 필요한 상황 정보를 상황 정보 관리자를 통하여 참조한다.

CAMUS에서 이벤트는 기본적으로 공표-예약(publish-subscribe) 모델에 기반하여 처리된다. 이벤트는 서비스 에이전트 관리자 등에 의해 공표되고 공표된 이벤트는 이벤트 시스템에 의해 장소별, 사람별, 그리고 로봇 스테이션별로 마련된 이벤트 채널(event channel)로 보내지게 된다. 이벤트 채널은 CAMUS 태스크 또는 상황 관리자와 같이 이벤트 처리를 담당하는 객체에 해당 이벤트를 전달한다.

이러한 기능적인 구조 하에서, CAMUS는 (그림 2)와 같이 물리적 환경에서 실장되는 형태를 기반으로 CAMUS 메인 서버와 서비스 에이전트 관리자로 구분된다. CAMUS 메인 서버와 서비스 에이전트



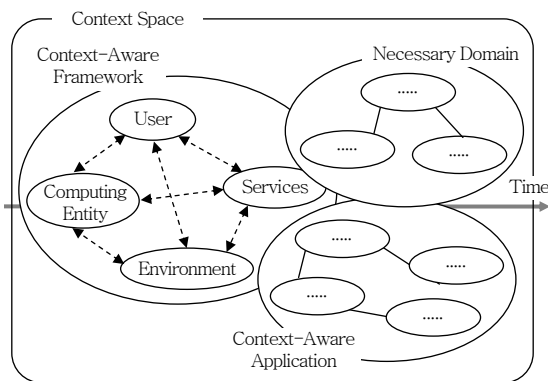
(그림 2) 물리적 환경에서 CAMUS 시스템 구성도

관리자 사이의 통신은 통신 프레임워크인 PLANET을 통하여 이루어진다.

2. 상황 공간

CAMUS에서는 URC 환경에서 지능적 서비스들을 제공하기 위하여 상황 정보를 표현하는 개념적 공간으로 (그림 3)과 같이 상황 공간(context space)을 사용한다. 상황 공간에는 상황 정보를 구성하는 기본적인 요소들인 사용자, 컴퓨팅 엔티티, 환경, 서비스와 이들 사이의 상호 작용들이 포함된다. 상황 공간에서 표현되는 상황 정보는 상황 정보 획득 지원을 위한 시스템 개발의 주체와 재사용성을 기준으로 프레임워크 상황 정보, 도메인 상황 정보, 그리고 응용 상황 정보로 분류된다. 프레임워크 상황 정보는 여러 상황 인식 응용에서 공통적으로 필요하며 빈번하게 사용되기 때문에 항상 상황 인식 시스템에서 상황 정보의 획득을 지원해야 하는 상황 정보이며, 도메인 상황 정보는 일부 상황 인식 응용의 필요에 의하여 특정 도메인에 관한 정보 혹은 지식을 기술하는 상황 정보이다. 여기서 상황 정보 도메인은 시스템 혹은 응용 개발자들이 특별한 목적으로 관심을 가지고 표현하는 상황 정보의 개념적인 경계이다. 마지막으로 응용 상황 정보는 프레임워크 상황 정보와 도메인 상황 정보에 포함되지 않지만 상황 인식 응용에 필요한 상황 정보를 의미한다.

상황 공간에는 시간의 개념이 포함된다. 시간은



(그림 3) CAMUS의 상황 정보 공간

많은 상황 인식 응용에서 중요한 속성으로, 시간의 변화에 따라 사용자, 컴퓨팅 엔티티, 환경, 그리고 서비스들과 그들의 상호 작용의 실제 내용들이 변화하게 된다.

3. 계층적 상황 정보 모델

CAMUS에서는 상황 인식 응용을 개발하는 데 있어서 앞서 제시한 상황 정보 공간을 기초로 하여 개념적인 계층적 상황 정보 모델을 구성하고 사용한다. CAMUS에서의 상황 정보 모델은 (그림 4)와 같이 상황 정보의 내용에 따라 네 개의 계층으로 구성된다.

첫번째 계층은 상황 정보 표현 형식의 개념화에 관한 계층으로 common ontology, domain ontology, application ontology, 그리고 instances들의 표현 형식 및 표현에 필요한 기본 요소들을 제공하는 표현 계층(representation layer)이다. RDF, RDFS, 그리고 OWL DL과 같은 온톨로지 및 규칙 표현 언어들이 이 계층에 위치한다.

두번째 계층에 위치하는 일반 계층(common layer)에는 상황 인식 시스템에서 제공하는 상황 정보들이 기술되고 제공된다. 일반 온톨로지(common ontology)는 다양한 응용에 공통적으로 사용되는 온톨로지 개념들, 즉 클래스들(classes)과 속성들(properties)이 표현된다. 일반 온톨로지는 상황 인식 응용들에 관한 높은 수준의 지식을 제공하며, 세번째 계층에 위치하는 도메인 온톨로지들과 네번째 계층에 위치하는 응용 온톨로지와 통합되어 하나의 상황 정보 모델을 구성할 수 있는 단서를 제공한다.

도메인 계층(domain layer)에는 특정 분야와 관련된 재사용 가능한 상황 지식 및 정보들이 기술된다. 이 계층에 위치하는 도메인 온톨로지는 일반 온톨로지를 이용하여 특정 도메인에 맞도록 상세화시킨 온톨로지이다. 지역 정보 온톨로지 혹은 날씨 온톨로지 등이 이 계층에 포함된다. 일반적으로 도메인 온톨로지는 상황 인식 응용 개발시 필요에 의해 재사용된다.

응용 계층(application layer)에는 특정 상황 인식 응용을 개발하는 데 필요한 상황 정보 및 지식이 기술된다. 일반 계층과 도메인 계층에서 묘사된 온톨로지를 바탕으로 특정 응용에 필요한 요소들을 추가함으로써 상세화시킨 온톨로지가 위치한다. 일반적으로 응용 온톨로지는 다른 응용의 개발 시에 재사용이 불가능한 온톨로지이다.

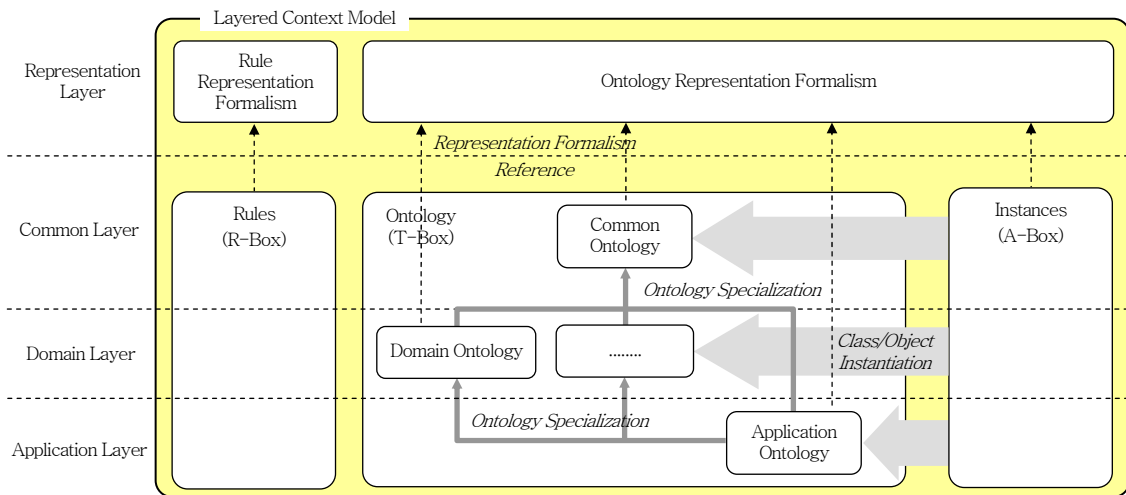
계층적 상황 정보 모델에는 앞서 설명한 온톨로지 이외에도 각각의 온톨로지와 관련된 인스턴스들과 규칙들이 포함된다. 인스턴스는 상황 정보 모델에서 온톨로지로 표현되는 개념들에 대응되는 실제 값들을 의미한다. 일반적으로 인스턴스들은 상황 인식 시스템이 기동된 후 생성되며, 갱신 정책에 의하여 주기적 혹은 특정 이벤트에 의해 갱신된다.

이러한 네 개의 계층으로 구성된 상황 정보 모델은 필요할 때 동적으로 구성되어짐으로써, 상황 인식 기반구조 내에서 관리되는 상황 정보뿐만 아니라 다양한 정보 소스에서 얻을 수 있는 상황 정보를 이용한 상황 인식 응용을 쉽게 개발할 수 있도록 도와준다.

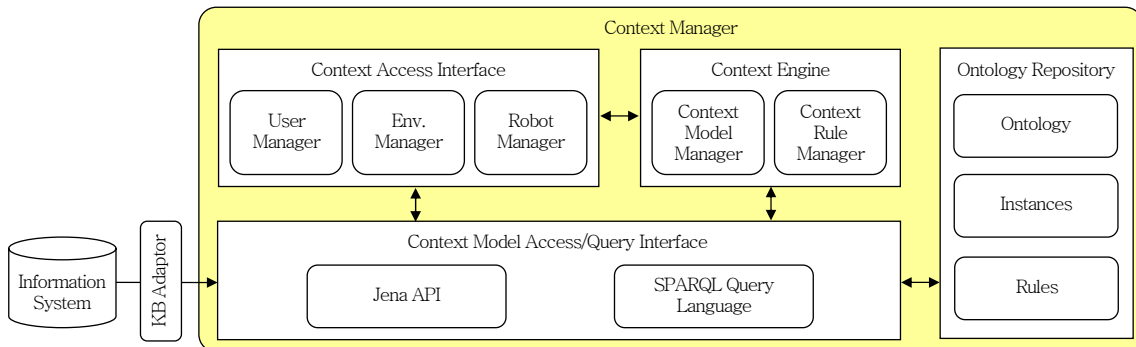
4. 온톨로지 기반 상황 정보 관리자

CAMUS 프레임워크에서 온톨로지 기반의 상황 정보 모델을 지원하는 상황 정보 관리자를 어떻게 개발하였는지에 관하여 기술한다.

(그림 5)와 같이 CAMUS 상황 정보 관리자는 크게 네 부분으로 구성된다. 상황 정보 관리자의 저장



(그림 4) CAMUS의 계층적 상황 정보 모델



(그림 5) CAMUS의 상황 정보 관리자 구조도

장치에는 (그림 4)의 계층적 상황 정보 모델에서 설명한 바와 같이 온톨로지, 인스턴스, 그리고 규칙들이 저장된다. 이 중 온톨로지와 인스턴스는 Jena 도구에서 제공하는 영속적 저장 장치인 RDB를 이용하게 되며, 규칙들의 경우에는 파일 단위로 저장된다.

상황 정보 관리자의 context engine에는 context model initializer, context model manager, context rule manager가 위치한다. Context model initializer는 DB에 저장된 상황 정보 모델을 초기화하고, CAMUS가 구동되는 시점에 상황 정보가 불필요한 인스턴스들을 삭제하는 역할을 한다. Context model manager는 상황 정보 모델 및 context rule manager를 생성하고 관리하는 역할을 한다. Context rule manager는 사용자 정의 규칙을 파일로부터 가져온 후, 환경 내의 상황의 변화로 인하여 상황 정보 모델의 내용이 변경되었을 때 이를 적용하는 역할을 한다.

상황 정보 관리자는 두 가지의 인터페이스를 제공한다. 첫번째는 상황 정보 관리자 개발에 사용된 Jena 도구에서 제공하는 Jena Ontology Model API와 쿼리 언어인 SPARQL Query Language이다. Jena API와 SPARQL Query Language를 사용하여 상황 정보 모델에 접근할 경우에는 상황 정보 모델 전체에 접근이 가능하다.

두번째는, 상황 인식 응용 구현시 빈번하게 사용되는 일반 상황 정보 온톨로지에서 제공하는 클래스와 속성들에 직접 액세스가 가능하도록 도와주는 context access interface이다. 여기에는 사용자 관리자(user manager), 환경 관리자(environment manager), 그리고 로봇 관리자(robot manager)가 위치한다. 상황 인식 응용 및 상황 정보 해석을 위한 상황 정보 해석기에서는 각각의 관리자에서 제공하는 메소드를 사용하여 보다 쉽게 상황 정보 모델을 검색, 참조, 그리고 변경할 수 있다.

IV. 결론

본 고에서는 네트워크 기반의 지능형 서비스로

봇을 개발하는 데 중요한 기술적 요소 중의 하나인 상황 인식 기술에 관하여 고찰하고, URC에서 상황 인식을 지원하는 CAMUS 미들웨어를 상황 정보 모델 및 상황 정보 관리 모듈 중심으로 설명하였다.

CAMUS에서는 높은 수준의 상황 정보 추상화와 다양한 상황 인식 응용 개발시 상황 정보의 확장성과 재사용성을 높이기 위하여 온톨로지 기반의 계층적 상황 정보 모델을 사용하고 있으며, 이를 지원할 수 있는 상황 정보 관리 모듈을 제공하고 있다. 제시된 상황 정보 모델의 구조 및 상황 정보 관리 모듈은 높은 수준의 상황 정보 해석을 위한 추론 기능을 제공할 뿐만 아니라 응용 개발시 필요한 사용자 정의 규칙들을 모델에 반영시킴으로써 상황 인식 응용 코드가 간결해지고 개발이 쉬워지는 장점을 제공하고 있다.

약어 정리

CAMUS	Context-Aware Middleware for URC Systems
CoBrA	Context Broker Architecture
ECA	Event-Condition-Action
ORL	Olivetti Research Ltd.
ORM	Object Role Modeling
SOCAM	Service-Oriented Context-Aware Middleware
SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications
TEA	Technology for Enabling Awareness
UML	Unified Markup Language
URC	Ubiquitous Robotic Companion

참고 문헌

- [1] H. Kim, Y.J. Cho, and S.R. Oh, "CAMUS - A Middleware Supporting Context-aware Services for Network-based Robots," *IEEE Workshop on Advanced Robotics and Its Social Impacts*, Nagoya, Japan, 2005.
- [2] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol.10, No.1, 1992.

- [3] A.K. Dey, "Providing Architectural Support for Building Context-Aware Applications," *PhD Thesis, Georgia Institute of Technology*, 2000.
- [4] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol.10, No.1, 1992.
- [5] A. Asthana, M. Cravatts, and P. Krzyzanowski, "An Indoor Wireless System for Personalized Shopping Assistance," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, 1994, pp.69-74.
- [6] I. Essa, "Ubiquitous Sensing for Smart and Aware Environments," *IEEE Personal Communications*, 2000, pp.46-49.
- [7] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, and D. Robbins, "The New EasyLiving Project at Microsoft Research," *Joint DARPA/NIST SmartSpace Workshop*, 1998.
- [8] A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K.V. Laerhoven, and W.V. Velde, "Advanced Interaction in Context," *In Proc. of First Int'l Symp. on Handheld and Ubiquitous Computing*, 1999, pp.89-101.
- [9] B. Johanson, G. Hutchins, T. Winograd, and M. Stone, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms," *IEEE Pervasive Computing Magazine*, Vol.1, No.2, 2002.
- [10] J. Borchers, M. Ringer, J. Tyler, and A. Fox, "Stanford Interactive Workspaces: A Framework for Physical and Graphical User Interface Prototyping," *IEEE Wireless Communications, Special Issue on Smart Homes*, 2002.
- [11] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems," *Proc. of the First Int'l Conf. on Pervasive Computing*, LNCS 2414, Zurich, Aug. 2002.
- [12] A. Ranganathan and R.H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," *In ACM/FIP/USENIX Int'l Middleware Conf.*, 2003.
- [13] H. Chen, "An Intelligent Broker Architecture for Pervasive Context-Aware Systems," *PhD Thesis, University of Maryland, Baltimore County*, 2004.
- [14] H. Chen, T. Finin, and A. Joshi, "An Ontology for Context-aware Pervasive Computing Environments," *The Knowledge Engineering Review*, Vol.18, No.3, 2003.
- [15] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, "Ontology Based Context Modeling and Reasoning Using OWL," *In Proc. of the Second IEEE Annual Conf. on Pervasive Computing and Communications Workshops*, 2004.