

# URC에서의 상황인식 컴퓨팅 기술

Context-Awareness Computing in Ubiquitous Robotic Companion

## 지능형 로봇 특집

정진미 (J.M. Jung)      소프트웨어로봇연구팀 연구원  
 이강우 (K.W. Lee)      소프트웨어로봇연구팀 선임연구원  
 김현 (H. Kim)          소프트웨어로봇연구팀 책임연구원, 팀장

## 목 차

- .....
- I . 서론
  - II . 상황인식 컴퓨팅 기술 동향
  - III . CAMUS
  - IV . 결론

URC는 기존 로봇에 네트워크 및 정보 기술을 접목한 지능형 서비스 로봇의 새로운 개념으로서, 언제, 어디서나 나와 함께 하며, 나에게 필요한 서비스를 제공하는 네트워크 기반 로봇이다. URC 개념이 구현되기 위해서는 유비쿼터스 네트워크 또는 센서 네트워크, 고성능 로봇용 서버 등과 같은 하드웨어 인프라가 구축되어 있어야 하며, 이러한 인프라 상에서 구동되는 소프트웨어 인프라가 필요하다. 본 고에서는 URC의 소프트웨어 인프라의 구현에 요구되는 많은 기술 중 특히 상황인식 컴퓨팅 기술에 대해 논의하고 ETRI에서 개발중인 상황인식 미들웨어인 CAMUS에 대해 소개하고자 한다.

## I. 서론

최근 정보 통신 기술의 발전에 따라 사람들은 이동중에도 컴퓨팅 파워와 네트워크 자원을 사용할 수 있게 되었다. 컴퓨터 메모리나 프로세서의 계산 능력은 엄청나게 높아졌고, 네트워크 대역폭 또한 계속 증가하고 있어 사용자가 언제 어디서나 정보에 접근하고 서비스를 받을 수 있게 되었다.

최근 이러한 기술의 발전은 로봇 분야에도 영향을 미치고 있다. 기존의 산업용 로봇 시장이 포화 상태에 이르면서 새로운 시장 창출이 요구되었고 이에 서비스 로봇에 대한 연구가 시작되어 현재 활발히 진행되고 있다. 서비스 로봇이 가전기구나 정보단말기와 같이 우리 생활 속에 들어오기 위해서는, 사용자가 저가로 쉽게 로봇을 구입하여 다양한 서비스를 받을 수 있도록 되어야 한다. 하지만 현재의 서비스 로봇은 가격에 비해 기능과 서비스가 단순하며 로봇 자체의 분명한 killer application을 갖고 있지 못하다. 이러한 배경 하에서 정보통신부에서는 로봇의 가격은 낮추는 반면 그 기능과 서비스는 다양해질 수 있도록 하기 위해 URC라고 하는 새로운 개념의 네트워크 기반 로봇을 개발중에 있다.

일반적으로 로봇은 외부 환경을 센싱하고, 이를 바탕으로 판단하고, 이 판단에 따라 행동하는 세 가지의 기능적 요소를 갖는다. URC는 로봇 자체에서 처리되던 이 세 가지 필수 기능을 외부 환경에 분산시키고 네트워크를 통해 외부의 센싱 기능과 프로세싱 기능을 충분히 활용할 수 있도록 하는 것이다. 즉, 로봇 자체의 센싱 기능을 늘려가기 보다는 외부 환경에 내재된 센서 기능을 활용할 수 있게 하고, 기존 로봇의 프로세싱을 높이기 보다는 원격지의 고성능 서버를 다수의 로봇이 공동으로 활용할 수 있게 하자는 것이다. 센싱 기능이 보완된다면, 외부 환경 및 사용자의 상황 인지가 훨씬 더 좋아질 수 있으며, 이에 따라 로봇이 보다 능동적으로 행동을 할 수 있게 된다. 또한 프로세싱의 제약이 극복됨으로써, 로봇의 기능이 그만큼 확장되고, 다양한 서비스가 가

능해지며, 궁극적으로는 로봇의 지능이 높아질 수 있게 된다. 나아가, 사용자는 로봇을 떠나 있어도 자신의 개인용 정보단말기를 가지고 원격지 서버에 접속하여 언제 어디서나 다양한 로봇 서비스를 받을 수 있게 된다. 결국 이러한 개념을 통해 최소한의 센싱 기능과 프로세싱 기능을 갖는 저가의 하드웨어 로봇만으로도 고도의 다양한 서비스를 받을 수 있으며, 단순 로봇 시장에 새로운 비즈니스 모델을 제시하여 이에 따른 부가가치 창출이 가능하게 된다.

URC 개념이 구현되기 위해서는 유비쿼터스 네트워크 또는 센서 네트워크 등의 네트워크 인프라, 고성능 로봇용 서버 등과 같은 컴퓨터 하드웨어 인프라가 먼저 구축되어야 한다. 또한, 로봇과 고성능 서버와의 네트워크 통신을 통해 언제 어디서나 다양한 서비스를 제공할 수 있는 소프트웨어 인프라가 필요하다.

본 고에서는 URC의 소프트웨어 인프라의 구현에 요구되는 많은 기술 중 특히 상황인식 기술에 대해 논의하고 ETRI에서 개발중인 상황인식 미들웨어인 CAMUS에 대해 소개하고자 한다.

## II. 상황인식 컴퓨팅 기술 동향

상황인식은 URC 서비스에서 수행되어야 할 가장 중요한 기능 중의 하나다. 즉, 사용자가 로봇에게 정보나 서비스를 요청할 때 현재의 상황을 이해해서 그 상황에 맞는 서비스를 제공하거나, 사용자의 요청이 없는 경우에도 필요한 시점에 필요한 정보와 서비스를 지능적이고 능동적으로 제공할 수 있어야 한다. 이를 위해서는 사용자 주변의 다양한 상황정보가 잘 모델링되고, 수집되고, 분배되어야 함은 당연하다. 이에 지금까지 정의된 상황(context)과 상황을 센싱하는 여러 가지 방법 및 상황인식 컴퓨팅(context-aware computing)에 대해 살펴보고 기 개발된 상황인식 응용(application)을 대략 기술한다.

## 1. 상황 정의

Merriam-Webster의 Collegiate Dictionary에 의하면 “context”는 “무엇인가가 존재하거나 발생하는 데 관련 있는 조건들”이라고 되어 있다. 이는 일반적인 정의이긴 하지만 컴퓨팅 환경에서의 개념을 이해하는 데는 도움이 되지 않는다. 사실 상황이란 단어의 사용은 좀 모호하다. 이 세상에서 일어나는 모든 일이 상황이 될 수 있기 때문이다. 지금까지 일반적인 정의에 만족하지 못한 많은 학자들이 구체적인 예를 열거하며 상황에 대한 정의를 내리고 있는데, Schilit은 상황을 다음과 같은 세 개의 범주로 나눈다[1].

- Computing context: 네트워크 연결 상태, 통신 대역폭, 그리고 프린터, 디스플레이, 워크스테이션과 같은 주변의 컴퓨팅 자원들
- User context: 사용자의 프로파일, 위치, 주변의 사람들 등
- Physical context: 조명, 소음 레벨, 교통 상태, 온도 등

Guanling Chen과 David Kotz는 시간 역시 많은 응용에서 중요시 여기는 본질적인 상황이라 생각하고 네번째 범주를 제안했다[2].

- Time context: 시간, 주(week), 달(month), 계절 등

컴퓨팅, 사용자, 물리적 상황 등을 시간 축을 따라 기록하면, 상황에 대한 이력을 응용에 이용할 수 있는데 이것 역시 유용한 상황정보가 된다.

또한, 여러 가지의 상황 정보를 융합함으로써 현재의 상태를 좀 더 확실하게 이해할 수도 있다. 이때 위치, 개체, 시간 등을 포함한 기본적인 상황들은 응용 시스템에 특별한 또 다른 상황들을 위한 인덱스 역할을 할 수 있다. 예로, 현재의 장소와 시간을 알고 사용자의 일정이 적힌 목록을 갖고 있으면, 응용은 사용자가 회의중인지, 수업을 듣고 있는지, 공항에서 대기중인지 등과 같은 사용자 현재 상태를 더욱 잘 파악할 수 있을 것이다.

몇몇 학자들은 상황에 대해 또 다른 정의를 내리고 있다. Schmidt 등은 “주위환경, 위치 등을 포함한 사용자와 정보기기들의 상태에 관한 지식”이라고 정의한다[3]. Dey는 “어떤 개체의 상태를 특징짓는 데 이용될 수 있는 임의의 정보. 여기서 개체란 사용자와 응용 그 자체를 포함해서 둘 사이의 상호작용에 관련된 사람, 장소, 객체 등이다.”라고 상황을 정의하고 있다[4]. Chen은 모바일 컴퓨팅의 관점에서 상황은 “응용의 동작을 결정짓거나 이벤트를 발생시키는 주위의 상태와 설정(settings)들의 집합 그리고 사용자가 관심을 돌만한 것”이라고 했다[2]. 더불어 응용의 동작에 영향을 주는 종류의 상황을 능동적(active) 상황이라 하고 응용에 관련은 있지만 그다지 중요하지 않은 종류의 상황을 수동적(passive) 상황이라 칭하였다. 하나의 특정 상황이 주어졌을 때, 그것이 능동적이냐 수동적이냐는 응용에서 어떻게 사용되는지에 따라 다르다.

## 2. 상황 센싱 및 모델링

응용이 상황을 사용하기 위해서는 먼저 현재의 상황이 센싱되고 이것이 응용에 전달돼야 한다. 따라서 종류가 다른 상황마다 이를 수행하는 메커니즘이 필요하다.

위치(location)는 가장 많이 이용되는 중요한 상황으로써 사용자가 움직일 때마다 값이 변하기 때문에 많은 상황인식 응용에 있어 믿을 만한 위치 추적 시스템이 요구된다. 만일 사용자가 기꺼이 그리고 잊어버리지 않고 항상 자신의 위치 정보를 시스템에게 알린다면 위치 상황을 획득하는 것이 쉬워진다. 전형적으로 사용자가 방을 출입하기 전에 자신의 배지 혹은 지문을 확인시켜 주는 방법이 있다. 또한 시스템이 사용자가 어느 워크스테이션에 로그인했는지 감시할 수도 있다. 하지만 이러한 방법들은 사용자의 적극적인 협조 하에 가능한 것이고 만일 사용자가 시스템에 알리는 것을 잊고 방을 떠난다면 정확성을 보장할 수 없게 된다.

위치는 실외와 실내에서 센싱하는 방법이 다르다.

실외에서는 단연 GPS를 사용한다. 최근 미국 정부는 GPS 신호에 대한 규제를 완화시켜 기존의 10배 이상인 10~20미터의 정확도가 가능하게 되었다. 다른 한편, GPS는 실내에서 제대로 작동하지 않는데 대부분 건물을 통과하기에는 신호의 세기가 미약하기 때문이다. 따라서 실내에서 위치를 잘 감지할 수 있는 값싸고 질 좋은 센서를 만드는 것은 중요한 문제이다. 이제까지의 많은 연구 과제를 보면 다양한 방법으로 위치 추적 시스템을 만들고 있다. 예로, Xerox ParcTab에서 수행한 Oivetti Active Badge 시스템의 경우에는 적외선(IR)을 이용하였고 AT&T에서 제안한 Personal Shopping Assistant는 RF 송신기와 수신기를 사용하여 핸드헬드 장치나 사람을 추적하였다. 이 외에 초음파와 무선 신호를 이용한 것들도 있다. 또한 네트워크 도메인을 바탕으로 구축된 몇몇 위치추적시스템은 실내와 실외를 특별히 구분하지 않는다. GUIDE 프로젝트와 같은 경우, IEEE 802.11에 기반한 셀룰러 시스템을 만들었다. 여기서 셀은 WaveLAN 802.11 기지국의 범위에 의해 정의된다. 아울러 위치 변화를 찾아내는 데 Mobile-IP 프로토콜을 사용하는 방법도 시도되고 있다.

위치뿐만 아니라 다른 많은 종류의 상황 역시 다양한 방법으로 센싱하는 것이 가능하다. 이 중에서 시간은 컴퓨터에 부착된 시계로부터 어렵지 않게 획득할 수 있다. 많은 응용에서 위치 정보와 타임스탬프를 연관시켜 사용하고 있다. 근처에 있는 객체 정보를 필요로 할 경우에는 시스템이 사람과 다른 객체의 위치를 기록해 두고 그 위치가 저장된 데이터베이스를 검색함으로써 역시 쉽게 정보를 얻을 수 있다. 네트워크 대역폭 또한 중요한 상황정보인데 시스템이 지원해 주지 않으면 응용이 대역폭 변화에 적응할 수 있는 손쉬운 방법은 없다. 이 외에 조명을 감지하는 광전다이오드, 경사와 진동을 측정하는 가속도계, 사람과의 근접도를 탐지하는 수동 IR 센서, 소리를 감지하는 마이크, 온도와 압력을 위한 센서 등을 통해 다종의 상황정보를 얻을 수 있다.

앞서 언급한 위치, 시간, 온도 등과 같이 가공되지 않은 상황정보 외에도 사용자의 “현재 행동”과 같은 고수준의 상황정보가 있다. 이것을 추출해 내는 것은 매우 복잡한 문제로 세 가지 접근 방법이 있다. 하나는 카메라 기술과 이미지 처리에 기반한 머신 비전을 이용하는 것이다. 다른 하나는 사용자가 특정 시간에 무엇을 할 것인지가 적혀 있는 일정표를 참조하는 것이다. 하지만, 이것은 사용자가 항상 일정에 따라 행동하는 것이 아니므로 쉽지가 않다. 마지막으로 몇 개의 저수준의 상황을 결합하여 복잡한 상황을 이끌어내는 인공지능 기술이 있다.

위와 같은 방법으로 센싱된 상황정보는 응용시스템이 사용하기 쉽도록 잘 모델링 돼 있어야 한다. 상황정보는 그 종류에 따라 서로 다른 속성을 가지기 때문에 속성을 표현하고 모델링하는 방법이 달라진다. 특히, 상황정보 중 가장 널리 사용되는 위치에 관해서는 객체의 이동을 제어하고 위치와 관련된 질의(예를 들면, 한 객체가 주어지면 그 객체의 현재 위치 정보를 반환하라 또는, 임의의 위치가 주어지면 거기에 있는 모든 객체 목록을 반환하라. 등등)를 처리하기 위해 효과적이고 이상적인 모델이 요구된다. GIS가 기하 데이터와 위상 데이터를 저장하고 처리하긴 하지만, 그것들은 기본적으로 정적인 지도 데이터베이스이며 실시간 객체 추적을 다루지 못한다.

위치 모델에는 전형적으로 위치를 추상화된 심벌로 표현하는 상징적(symbolic) 모델과 좌표로 표현하는 기하학적(geometric) 모델이 있다. 어떤 것을 선택할지는 주요 센서 시스템의 출력 양식에 달려 있는데, 대체적으로 GPS 기반 시스템은 기하학적 모델을 사용하고 액티브 배지 기반 시스템은 상징적 모델을 사용한다.

### 3. 상황인식 컴퓨팅

개발자가 응용 시스템을 개발할 때 상황정보를 어떻게 효율적으로 활용할 것인가 하는 것은 매우 중요한 이슈 중의 하나이다.

Schilit는 상황인식 시스템을 다음과 같이 분류하였다[1].

- Proximate selection: 사용자 근처의 객체가 쉽게 선택될 수 있도록 만드는 사용자 인터페이스 관련 응용
- Automatic contextual reconfiguration: 상황이 변경됨에 따라 새로운 컴포넌트를 추가하거나 기존의 컴포넌트를 제거하거나 컴포넌트간의 연결을 변경하는 것과 관련된 응용
- Contextual information and commands: 결과를 산출하는 데 있어 상황이 영향을 미치는 응용
- Context-triggered actions: 상황인식 시스템이 어떻게 변경되어야 하는지 명시하는 데 사용되는 단순한 IF-THEN 규칙들

Pascoe는 상황인식의 특징을 상황센싱(contextual sensing), 상황적응(contextual adaptation), 상황자원발견(contextual resource discovery), 상황증대(contextual augmentation) 등과 같이 분류할 것을 제안했다[5]. 여기에서 상황증대는 사용자의 상황에 디지털 데이터를 결합시키는 능력을 뜻한다. Dey는 이러한 아이디어를 묶어서 상황인식 응용이 지원할 수 있는 특징을 세 개의 일반적인 범주로 구분하였다[4]. 즉, 정보와 서비스를 사용자에게 주는 것, 서비스를 자동으로 수행하는 것, 나중의 검색을 위해 정보를 태깅하는 것이다. Chen과 Kotz는 그들이 내린 상황의 정의를 바탕으로 상황인식 컴퓨팅 역시 수동적인 것과 능동적인 것으로 구분하여 정의를 내리고 있다[2].

- Active context awareness: 새로운 상황이 발견되면 자동으로 그것에 맞게 동작을 변경하는 응용
- Passive context awareness: 새로운 또는 변경된 상황을 관심을 갖는 사용자에게 제공하거나 후에 검색하도록 유지하는 응용

#### 4. 상황인식 애플리케이션

상황인식 컴퓨팅은 약 10년 전에 제안된 이래 많은 학자들이 연구해왔다. 또한, 기술의 유용성을 입증하기 위해 여러 가지의 상황인식 응용이 구축되었다. 하지만 이러한 응용이 일반인에게 널리 이용되지는 못하였다. 다음은 이미 개발된 몇몇 상황인식 응용에 관한 대략의 설명이다.

- Call Forwarding[6]: Olivetti Research Ltd.에서 개발한 것으로 사용자의 위치를 상황 정보로 사용한다. 이것은 사용자의 위치를 인식해서 전화가 왔을 때 사용자로부터 가장 가까운 곳에 있는 전화기로 연결해 준다.
- Teleporting[7]: 이것 역시 Olivetti Research Ltd.에서 개발한 것으로 주변의 컴퓨터와 통신 시설에 사용자의 인터페이스를 동적으로 매핑시키는 툴이다.
- Active Map[8]: Xerox PARC에서 개발하였다. 방마다 각각의 무선 기지국을 두고 여기에서 위치 정보를 수집한다. 지도 상에는 사용자의 위치를 나타내는 개개의 면들이 존재하고, 이 정보는 몇 초 단위로 갱신된다.
- Mobisaic Web Server[9]: Washington 대학에서 만든 것으로 위치와 시간을 이용한다. Mobisaic은 표준 클라이언트 브라우저를 확장한 것으로 웹 페이지 저작자가 하이퍼텍스트에 환경 변수를 내포하고 있는 동적인 URL을 링크시킬 수 있도록 한다. 동적 URL이 환경 변수의 현재 값을 적용하여 확정되면 그에 해당하는 적절한 페이지가 반환된다. 환경 변수의 값이 바뀔 때마다 웹 페이지는 자동으로 갱신된다.
- Shopping Assistant[10]: AT&T Bell 연구소에서 개발하였다. 이 장치는 사용자에게 물품의 위치와 상세정보를 알려주고 세일중인 물품 목록을 제공하며 가격을 비교 분석하는 등 매장에서의 쇼핑을 안내한다. 매장이 고객의 프로파일을 관리해야 하므로 프라이버시 문제가 고려되어야 하는데 결과적으로 고객은 자신의 프로



파일을 제공하지 않고 익명으로 쇼핑하기를 원하는 부류와 자신의 프라이버시를 희생하는 대가로 추가적인 할인을 기대하는 부류로 나뉜다.

- Cyberguide[11]: Georgia 기술 연구소에서 개발한 것으로 관광객의 위치와 시간을 상황정보로 이용한다. 다양한 종류의 시스템이 사용자에게 현재 위치에서 이용 가능한 정보 서비스를 제공한다. 또한, 사용자가 이전에 여행한 이력을 자동으로 편집하여 새로운 여행지를 추천하는 데 이용한다. 외부의 위치정보는 GPS를 통해 획득하고 내부에서는 IR positioning 시스템을 사용한다.
- Conference Assistant[12]: Georgia 기술 연구소에서 개발하였다. 이 시스템은 회의 참석자를 돕기 위해 다양한 상황정보를 이용한다. 회의 일정, 사용자의 위치, 사용자의 관심 분야 등을 조사하고 사용자가 발표장에 들어가면 자동으로 발표자의 이름과 발표 주제 및 다른 관련 정보를 보여준다. 또한 나중의 검색을 위해 주변의 이용 가능한 오디오/비디오 장치가 자동으로 현재의 슬라이드, 의견, 질문을 기록한다.
- People and Object Pager[13]: Kent 대학에서 개발한 것으로 상황정보로 사용자의 위치, 근처의 사람과 사물을 이용한다. Pager는 방문객으로부터 가장 가까이에 위치한 사람에게 메시지를 라우팅함으로써 페이징 장치는 갖고 있지 않지만 액티브 배지를 가진 방문객에게 메시지를 보낼 수 있다.
- Adaptive GSM phone and PDA[3]: Star-lab의 TEA에서 개발한 것으로 PAD 시나리오에서 노트패드 는 사용자의 행동에 맞게 글자 크기를 조절한다. 즉 사용자가 걷고 있을 때는 크기를 크게 하고 정지해 있을 때는 작게 한다. 또한 조명과 같은 주변 조건에 따라서도 변경된다. 폰 시나리오에서는 모바일 폰의 프로파일이 인식된 상황에 따라 자동으로 선택된다. 다시 말해 폰이 손에 있는지, 테이블 위에 있는지, 가방 안에 있는지 혹은 밖에 있는지에 따라

자동으로 소리를 낼 것인지, 진동을 할 것인지를 선택하고, 소리의 경우 크기를 조절할 수도 있다.

앞서 보듯이 현재 상황인식에 관한 많은 연구가 진행되고 응용들도 구현돼 왔지만, 상황인식 시스템을 구현하는 데는 몇 가지 해결해야 할 이슈들이 있다. 상황정보는 전형적인 사용자 입력 형태와 달리 많은 다양한 소스로부터 정보가 입력된다. 따라서 입력정보인 상황정보에 대한 추상화가 요구된다. 또한 상황정보를 센싱하는 컴포넌트와 실제 응용 시스템이 결부되어 있지 않고, 완전히 분리되어 있다. 또한 상황정보를 처리하기 위한 서비스와 장치를 동적으로 찾을 수 있어야 한다. 이와 같은 문제점으로 인하여 이제까지 상황인식 시스템은 ad-hoc한 방법으로 만들어져 왔다[14].

### III. CAMUS

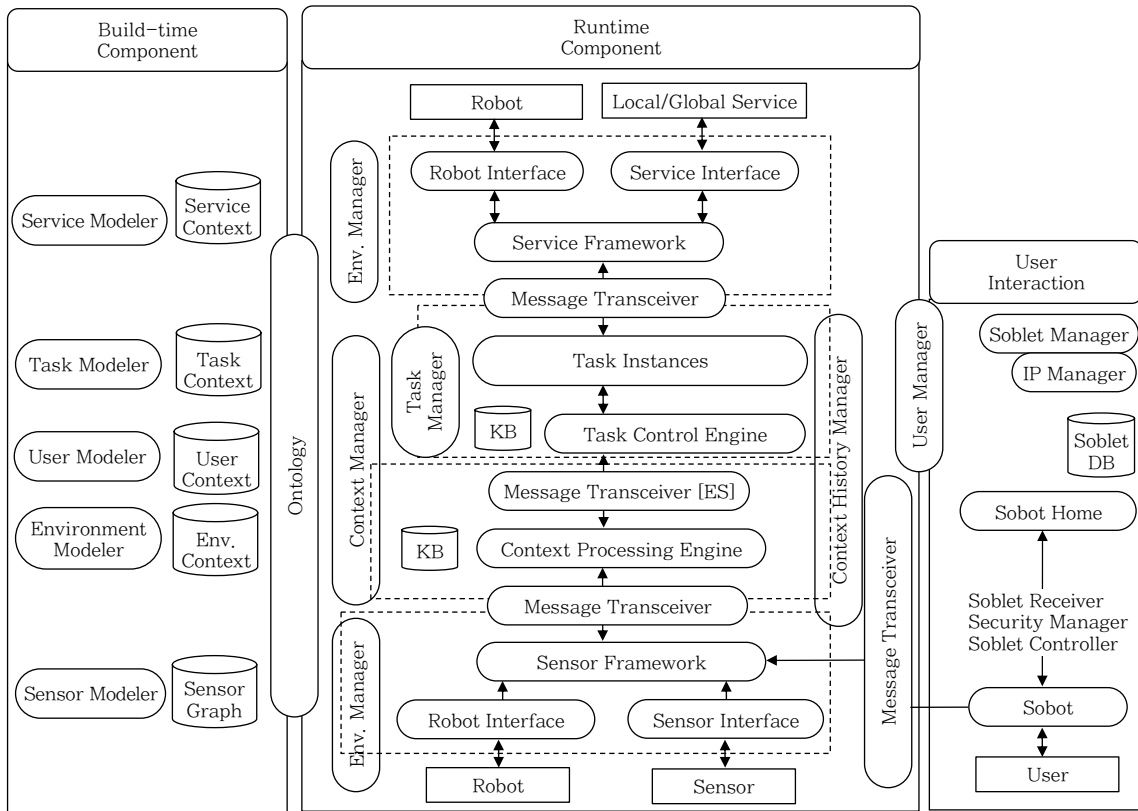
다음은 ETRI에서 개발중인 URC를 위한 상황인식 미들웨어인 CAMUS를 소개한다.

(그림 1)은 CAMUS의 시스템 구조도를 보여준다. CAMUS는 물리공간에 대한 가상공간 모델링 부분, 작업 실행 부분 및 사용자 인터페이스 부분으로 나뉜다.

#### 1. 가상공간 모델링

가상공간 모델링은 크게 센서 모델링(sensor modeling), 서비스 모델링(service modeling), 환경 모델링(environment modeling), 사용자 모델링(user modeling) 및 작업 모델링(task modeling) 등으로 구성된다.

센서 모델링은 물리 공간의 센서를 가상 공간으로 매핑하고, 이들 센서 정보로부터의 상황을 추출하여 수행 엔진(task engine)에 제공함으로써, 능동적인 서비스를 제공할 수 있도록 지원한다. 센서 모델링의 결과는 서비스로 작성되며, 작업에 의해 서비스로 호출된다.



(그림 1) CAMUS의 시스템 구조도

서비스 모델링은 장치 제어, 음성 처리, 일정관리 등과 같이 CAMUS의 작업 단위로부터 수행될 서비스(컴퓨터 프로그램 모듈)들에 대한 인터페이스와 이들 구현 코드에 대한 관리를 지원한다.

본 논문에서 CAMUS에 의해 호출되는 서비스는 CORBA 객체로 가정하였다. 따라서 서비스 인터페이스는 IDL로 정의되며, 구현 코드는 CORBA stub/skeleton code를 포함하는 프로그램 코드들의 집합이다.

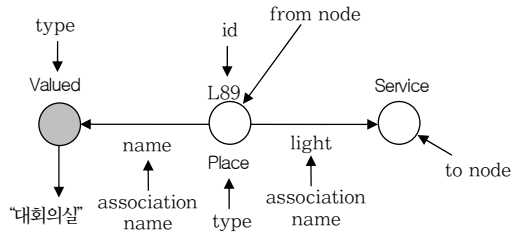
환경 모델링은 물리공간의 한정된 영역과 해당 영역에 존재하는 가용 자원에 대한 모델링이다. 네트워크로 통신할 수 없는 센서나 기기 등은 가상공간 상에서 의미가 없다. 따라서 이러한 센서나 기기는 궁극적으로 서비스의 형태로 시스템과 상호작용을 한다. 따라서 환경 모델링이 끝나고 나면, 서비스 모델링 시점에 등록된 코드들이 실행되어 서비스

를 준비하게 된다.

사용자 모델링은 CAMUS에 등록된 사용자 정보에 대한 모델링이다. 사용자 정보는 인적 프로파일 정보 및 선호사항 정보 등을 포함한다.

작업 수행을 위해서는 작업 모델링이 이루어져야 한다. 이는 기존에 모델링된 환경 정보를 이용하여, 특정 작업을 수행하기 위해 필요한 특정 상황정보를 모델링하고, 해당 작업 수행에 필요한 작업 규칙(task rule)을 작성하는 것이다.

CAMUS에서 사용하는 모든 상황 정보는 UDM을 통해 표현된다. UDM에서는 모든 상황 정보를 노드와 노드 사이의 연관으로 표현한다. 노드는 가상공간에서 개체를 표현하며 사람, 장소, 작업, 서비스 등이 된다. 노드들 중 값을 갖는 것은 “valued” 노드라 불리며, Java 객체를 값으로 갖는다. 이 객체는 작업 수행중에 참조되거나, 변경될 수 있다.



(그림 2) Universal Data Model

연관은 노드 사이의 관계를 기술하며, 방향성을 갖는 화살표로 표현되며 그 의미를 표현하는 이름을 갖는다. 화살표가 시작되는 노드를 시작 노드(from node)로, 화살표가 끝나는 노드를 종료 노드(to node)라 부른다. (그림 2)는 UDM을 통한 모델링의 예를 보여 준다.

본 그림은 “대회의실”이라는 이름을 갖는 장소에 “light”라는 이름의 서비스가 존재함을 모델링하였다. 그림을 통해 “L89”라는 장소 노드가 있고, 이 노드는 “name”이라는 연관을 통해 “대회의실”이라는 값을 갖는 valued 노드로 연결되었고, “light”라는 연관을 통해 연결된 서비스 노드를 볼 수 있다.

CAMUS에서의 작업은 다수의 작업 규칙으로 구성되어 있다. 각 작업 규칙은 ECA 규칙 표현으로, event와 condition을 통해 적절한 상황을 기술하고, 해당 상황이 발생될 때 수행할 일을 action으로 기술한다.

CAMUS에서는 상황인식 응용 개발자들이 보다 쉽게 작업 규칙을 작성토록 하기 위해 기존의 Java 언어를 확장한 PLUE라고 하는 작업 규칙 기술 언어를 제공한다. 다음은 PLUE로 기술된 간단한 작업 규칙의 예를 보여 준다.

```
on($place.temperature::ValueChanged e)
if(e.value>$place.resident.preferred_temp.high)
{
    $place.air_conditioner.turnOn();
}
```

“on” 절에는 작업 규칙의 수행을 유발하는 이벤트 연산식(event description)이 기술되고, “if” 절에서는 해당 작업 규칙이 수행되기 위한 조건을

기술하고, if 절 내부는 “action”에 해당하는 부분으로서, 명시된 조건이 만족하는 경우 수행될 서비스들을 기술한다. 이 예에서는 온도 값이 변화되고, 사용자가 원하는 온도보다 높으면, 에어컨을 작동시키는 간단한 규칙을 보여준다. 일반적으로 if 절에 UDM을 이용한 상황 정보의 참조가 경로 연산식(path expression)을 통해 이루어진다.

## 2. 작업 수행

(그림 3)은 CAMUS에서의 작업 수행에 관련된 정보의 흐름과 시스템 구성 모듈을 보여준다.

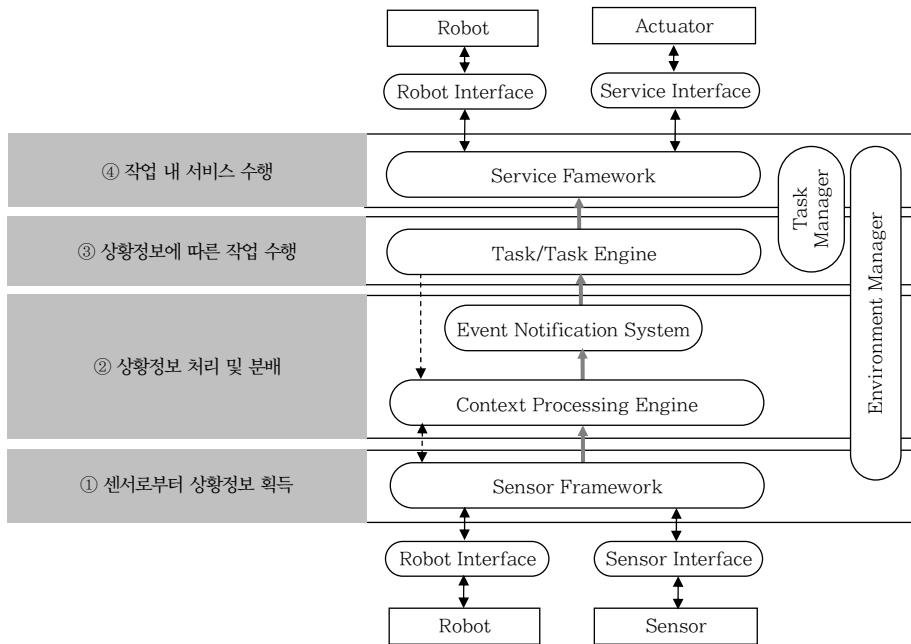
센서 프레임워크(sensor framework)는 환경 내의 물리센서, 응용 시스템 또는 사용자 명령 등의 입력 정보를 가공하여 상황 처리 엔진에게 전달하는 모듈이다. 사용자로부터의 음성정보, 영상정보, 온도/습도 센서, 사용자 일정정보 등이 모두 센서 정보가 될 수 있다. 예를 들어, 음성명령의 경우, 음성인식기(speech recognizer)를 통해 사용자의 음성명령을 문자열로 변환하고 상황정보에 맞추어 가공하여 이를 상황 처리 엔진(context processing engine)에게 넘겨 준다.

상황 처리 엔진은 센서 프레임워크로부터 전달된 상황정보로부터 상위 수준의 상황정보를 얻고 이를 별도의 저장소에 관리한다. 상위 수준의 상황정보는 작업 엔진에서 처리될 수 있는 symbolic data이다. 저장소에서 관리되는 정보는 추후 UDM을 통해 작업 수행중 참조된다. 또한 상황관리자는 상황 정보에 변경이 발생했을 때, 그 event를 이벤트 통지 시스템(event notification system)으로 전달하고, 작업엔진(task engine)에서 작업을 수행하는 데 필요한 상황정보를 제공하는 역할을 한다.

이벤트 통지 시스템은 센서나 기타 외부 서비스를 통해 상황 관리자에 의해 전달되는 이벤트를 이를 원하는 작업에게 전달하는 역할을 한다.

작업 관리자(task manager)는 개별 작업을 기동시키고, 수행중인 작업 프로세스를 관리 또는 제어하는 역할을 한다. 작업 엔진은 상황에 따른 실





(그림 3) Task Execution in CAMUS

제 작업을 수행시킨다. 작업 엔진은 해당 작업에 대한 상황 정보(fact)와 작업 규칙(rule)을 추론할 수 있는 추론 엔진을 갖는다. 본 시스템에서는 JESS를 추론 엔진으로 사용한다.

마지막으로, 서비스 프레임워크(service framework)는 작업 엔진에 의해 요구되는 서비스를 탐색하거나 실제 서비스를 호출하는 역할을 담당한다.

### 3. 사용자 인터페이스

사용자와 CAMUS 간의 상호작용은 소봇(sobot)에 의해 이루어진다. 소봇은 사용자의 명령을 받아 시스템에 전달하거나 시스템으로부터의 메시지를 사용자에게 전달하는 역할을 한다. 소봇은 소봇 플랫폼(sobot platform)이 탑재된 로봇을 포함한 다양한 정보단말기 상에서 구동하며, 소봇 코어(sobot core), 소블릿(soblet) 및 소봇 홈(sobot home)으로 구성된다. 소봇 코어는 아바타 형태의 소프트웨어 캐릭터, 음성 인터페이스 및 소블릿 실행 환경으로 구성되는 소봇 구동을 위한 최

소 핵심부이다. 소블릿은 특정 서비스와 연관되어 클라이언트에서 요구되는 각종 프로그램 모듈들로서, 소봇 코어에 의해 필요한 시점에 다운로드 된다. 소봇 홈은 소블릿을 관리하고, 필요한 시점에 소블릿 코드를 단말기에 전송하며, CAMUS 실행 엔진과의 통신을 담당하는 서버이다.

## IV. 결론

본 고에서는 URC에서의 가장 중요한 기술적 요소 중의 하나인 상황인식 기술에 대해 고찰하고 URC에서 상황인식을 지원하는 시스템 미들웨어인 CAMUS를 소개하였다. CAMUS는 로봇이 서비스하는 환경을 모델링하기 위한 방안을 제시하며, 환경 내 상황정보를 이용하여 로봇이 보다 능동적인 서비스를 하기 위한 시스템 하부 엔진을 제공한다. 현재는 개발된 프로토타입 시스템을 이용해 여러 가지 로봇을 대상으로 가정 환경에서의 상황 인지 기반 서비스를 위한 몇 가지 시나리오를 구현중에 있다.

## 약어 정리

CAMUS	Context-Aware Middleware for URC Systems
ECA	Event Condition-Action
GIS	Geographical Information System
GPS	Global Positioning System
IDL	Interface Definition Language
IR	Infrared
PLUE	Programming Language for Ubiquitous Environment
RF	Radio Frequency
UDM	Universal Data Model
URC	Ubiquitous Robotic Companion

## 참고 문헌

- [1] Bill Schilit, Norman Adams, and Roy Want, "Context-aware Computing Applications," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, Dec. 1994, pp.85-90.
- [2] Guanling Chen and David Kotz, "A Survey of Context-Aware Mobile Computing Research," Dartmouth Computer Science Technical Report TR2000-381.
- [3] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde, "Advanced Interaction in Context," *In Proc. of First Int'l Symp. on Handheld and Ubiquitous Computing*, Karlsruhe, Germany, Sep. 1999, pp.89-101.
- [4] Anind K. Dey and Gregory D. Abowd. "Towards a Better Understanding of Context and Context-awareness," Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999.
- [5] Jason Pascoe. "Adding Generic Contextual Capabilities to Wearable Computers," *In Proc. of the Second Int'l Symp. on Wearable Computers*, Pittsburgh, Pennsylvania, Oct. 1998.
- [6] Roy Want, Andy Hopper, Veronica Falcão, and Jona than Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, Vol.10, No.1, Jan. 1992, pp.91-102.
- [7] Frazer Bennett, Tristan Richardson, and Andy Harter. "Teleporting-making Applications Mobile," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, Dec. 1994, IEEE Computer Society Press, pp.82-84.
- [8] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser, "The ParTab Ubiquitous Computing Experiment," *Mobile Computing*, Chapter 2, Kluwer Academic Publishers, 1996.
- [9] Geoffrey M. Voelker and Brian N. Bershad, "Mobisaic: An Information System for a Mobile Wireless Computing Environment," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, Dec. 1994, IEEE Computer Society Press, pp.185-190.
- [10] Abhaya Asthana, Mark Cravatts, and Paul Krzyzanowski, "An Indoor Wireless System for Personalized Shopping Assistance," *In Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, Dec. 1994, pp.69-74.
- [11] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton, "Cyberguide: A Mobile Context-aware Tour Guide," *Wireless Networks*, Vol.3, No.5, Oct. 1997, pp.421-433.
- [12] Anind K. Dey, Masayasu Futakawa, Daniel-Salber, and Gregory D. Abowd, The Conference "Assistant: Combining Context-Awareness with Wearable Computing," *In Proc. of the 3rd Int'l Symp. on Wearable Computers*, San Francisco, CA, Oct. 1999, pp.21-28.
- [13] Peter J. Brown. "Triggering Information by Context," *Personal Technologies*, Vol.2, No. 1, Mar. 1998.
- [14] Patrick Fahy, Siobhan Clarke, "CASS - Middleware for Mobile Context-Aware Applications," *MobiSys 2004 Workshop on Context Awareness*.