

유비쿼터스 시스템 환경에서의 상황인지와 오토노믹 컴퓨팅†

한국과학기술연구원 강동훈 · 안상철 · 고희동* · 권용무 · 김형곤

아주대학교 조위덕*

1. 서 론

유비쿼터스 컴퓨팅(Ubiquitous Computing)은 사물에 컴퓨팅 기능이 내장되어 언제, 어디에서나, 어느 장치로도 편리하게 주변 환경으로부터 서비스를 사용할 수 있게 하는 정보기술의 패러다임이다. 이러한 패러다임의 변화에 대응하기 위한 정부 주도의 IT839전략은 미래 국가 비전으로서의 시너지가 높은 8대 정보통신 서비스, 이를 뒷받침하는 3대 첨단 인프라, 경쟁력이 있는 9대 신성장 동력으로 구성되어 있다. 이러한 IT839전략은 IT 산업 가치 사슬의 각 단계인 서비스-인프라-신성장 동력을 유기적으로 연결하는 전략으로서 u-Korea를 구현하기 위한 구체적인 산업정책이다. IT839 전략은 유비쿼터스 컴퓨팅을 기반으로 국가의 모든 자원을 네트워크화·지능화하여 궁극적으로는 센서 정보를 기반으로 시스템이 상황을 인지하여, 사람의 통제 없이 상황에 맞는 최적의 서비스를 찾아서 제공하는 자율화된 서비스(Autonomy Service) 시스템을 구현하는 데에 있다[1].

본 논문에서는 상황인지와 이를 기반으로 하는 지능형 서비스 시스템의 개념을 소개하고, 유비쿼터스 시스템 환경에서의 상황인지와 오토노믹 컴퓨팅과의 연계성을 살펴본다. 그리고, UPnP* 미들웨어를 적용한 홈 환경에서의 상황인지 기반의 지능형 서비스 구조를 제안한다.

2. 상황인지(Context-Awareness)

이 장에서는 상황인지의 개념을 제안하고, 상황을 추론하는 대상을 개별 작업(Task)을 그룹핑하여 묶는 하나의 액티비티(활동; Activity)로 지정하여 살펴본다.

2.1 상황인지의 개념 제안

상황인지는 시스템이 다양한 센서 정보를 바탕으로

스스로 상황(Context)을 인지하는 것으로서, 그 정의에 있어서, 아직 논란이 많으나 지능형 서비스를 위한 중요한 개념이다[2]. B. Schilit와 M. Theimer는 “사람과 그 사람에 인접한 사물의 Location, Identity 그리고 그 사물의 변화”를 상황으로 정의하였다. 그리고 그는 시간의 변화와 더불어, 사용자의 위치가 바뀔 때 따라 그 사용자의 위치에 인접한 사물들에게 소프트웨어가 적용할 수 있을 때에 이를 상황인지라고 하였다[3]. A.K. Dey는 “사용자와 Application 사이에 인터랙션이 있는 개체(Entity)를 Person, Place, Object라고 두게 되면, 그 개체의 Situation을 특징화할 수 있는 정보”를 상황이라고 두었다. 이 때 시스템이 사용자의 작업(Task)에 관련된 정보나 서비스를 제공하기 위해 상황정보를 사용한다면, 이 시스템은 상황인지를 하고 있다고 정의하였다[4]. B. Schilit의 개념을 발전시켜 특징화할 수 있는 정보에 주목을 한 A.K. Dey의 상황에 대한 정의는 사람과 사물이 일정한 시간에 어떤 특정한 장소에서 인터랙션이 있을 때, 사물의 변화를 특징화할 수 있는 정보를 말하는 것으로, 본질적으로는 B. Schilit와 같은 개념을 설명하고 있다. 여기에서 사용자 외에도 장치 간에 인터랙션이 있게 되면, 그 장치의 특징적인 변화를 상황정보로 고려할 수 있다.

본 논문에서는 상황인지의 개념을 “센서 정보를 복합하여 미리 설정된 상황정보를 귀납적으로 추론하는 논리적 연산과 그 결과”로서 정의한다. 그리고 상황정보는 “시간이 흐름에 따라 사용자의 위치가 변화하였을 때에 이에 따른 사용자의 행위나 장치의 동작으로 인해, 대상이 되는 사물의 변화가 수반되었을 경우, 그 특징적인 변화를 표현하는 정보”로 정의한다.

상황을 표현하는 기법으로서, 지식(Knowledge)을 표현하는데 사용되고 있는 온톨로지와 같은 메타 데이터를 적용하여 개념(Concept)과 개념 간의 관계를 설정하며 어떤 분류 체계를 정하는 모델링 기법이 활발히 연구되고 있다[5]. A. Ranganathan과 R.H. Campbell은 유비쿼터스 컴퓨팅에서의 상황정보를 First-order Logic으로

† 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천기술개발사업의 지원에 의한 것임.

* 종신회원

표현을 하는 모델을 제시하였다[6]. 뒤의 장에서 자세히 살펴보겠지만, 본 논문은 상황정보를 상위 수준과 하위 수준으로 구분하여, 모델링 하는 접근 방법을 따른다.

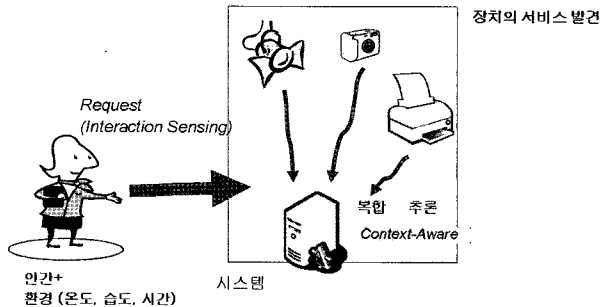


그림 1 복합·추론에 의한 상황인지

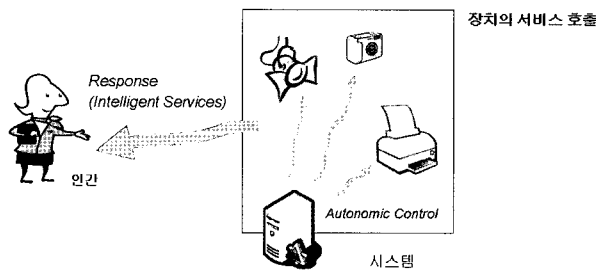


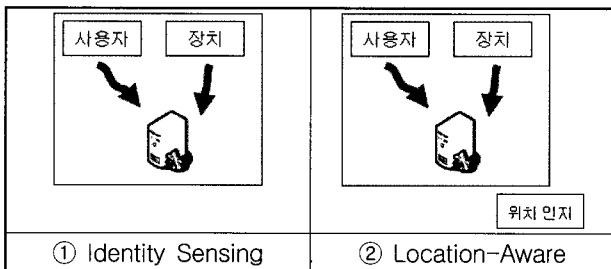
그림 2 상황인지 기반의 서비스

2.2 상황인지를 위한 센싱 정보

상황인지가 다양한 센서 정보를 복합하여 하나의 상황정보로 추론을 하는 것이라면(그림 1 참조), 지능형 서비스는 시스템이 이러한 상황정보를 기반으로 일련의 서비스를 묶어서 제공을 하는 서비스로 정의할 수 있다(그림 2 참조).

상황인지를 위한 센싱 정보는 사용자와 장치의 Identity를 인식하는 일이 우선적이 되며, 이를 기반으로 사용자나 장치가 있는 위치를 인식한다(표 1 참조).

표 1 하위 수준의 상황인지



여기에서 시스템은 시간 정보를 사용자, 장치 그리고 위치와는 독립적으로 획득할 수 있는데, 이러한 상황인지는 “하위 수준”의 상황인지로 정의할 수 있다. 사용자를 인지하는 방법은 Vision 기반의 얼굴인식, 지문인식과 같은 패턴인식이나 RFID와 같은 전자식별 장치 등을 활용한 인식 방법 등이 연구되고 있다. 그리고 장치

를 인지하는 방법으로는 UPnP[7]나 Jini[8]와 같은 미들웨어를 통하면, 네트워크를 통해 장치를 발견하고 해당 서비스 목록을 호출할 수 있다.

사용자와 장치의 위치를 인지하는 방법은 크게, (1) Triangulation, (2) Proximity, (3) Scene analysis의 세 가지 기술이 사용되고 있다. Triangulation은 이미 위치를 알고 있는 점을 기준으로 거리를 이용하거나 각도를 이용하여 원하는 지점의 위치를 계산하는 방법이다. 그리고 Proximity는 알고 있는 점들의 집합에서 가장 가까운 곳을 측정하는 기술이다. Scene analysis는 View를 분석하기 좋은 유리한 지점에서 해당 위치를 알아내는 방법이다(자세한 내용은 www.cs.washington.edu/research/portolano/papers/UW-CSE-01-07-01.pdf를 참조)[9].

시간 정보는 시스템이 독립적으로 획득이 가능한 센싱 정보이다. 따라서 사용 가능한 장치를 발견하고, 사용자와 관심이 있는 장치의 위치를 획득하는 부분은 시간 정보와는 서로 독립적이다. 이상에서 하위 수준의 상황정보를 요약하면, (1) 사용자와 장치의 Identity Sensing, (2) 사용자와 장치의 위치 인지, (3) 시간 정보로 구성이 된다. 이러한 하위 수준의 상황정보에서는 시간 정보와 함께, 센서를 포함한 인식 장치가 어떤 사용자나 장치의 특정 위치에서의 존재를 인지하고 있다고 할 수 있다.

2.3 상황인지의 대상: Activity

상위 수준의 상황인지는 사용자나 장치의 Identity와 위치를 인지한 후에 동작과 연관이 있는 상황정보를 인지하는 것이다. 따라서, 미리 상황정보를 설정을 해 둔다면, 일련의 동작이 어떠한 목적을 위해 수행되고 있는지를 추론할 수 있다. 이 때, 개별 동작을 시스템 입장에서는 작업 (Task)로 정의를 하고, 같은 목적을 위해 수행되는 일련의 행위, 곧 작업의 묶음을 Activity로 정의를 한다. 결국, 상황인지는 특정한 장소에서 일정한 시간에 사용자 혹은 장치에 의해 수행되는 Activity를 추론하는 과정이나 결과로 해석될 수 있다. 그림 3은 상황 모델을 나타낸 것으로, 시간은 독립적인 정보이므로 상

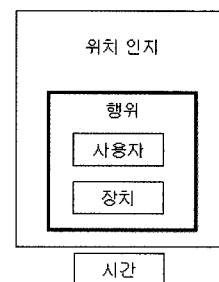


그림 3 상황모델

표 2 행위로 인한 장치의 특징적인 변화감지

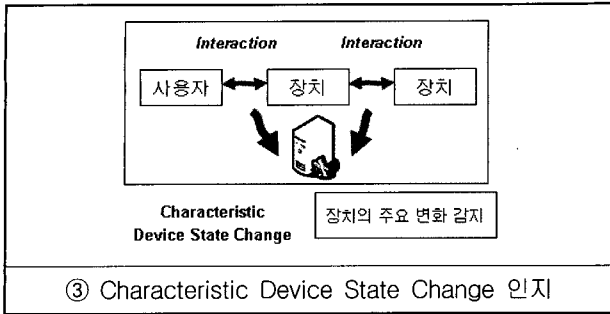


표 3 상위수준의 상황인지

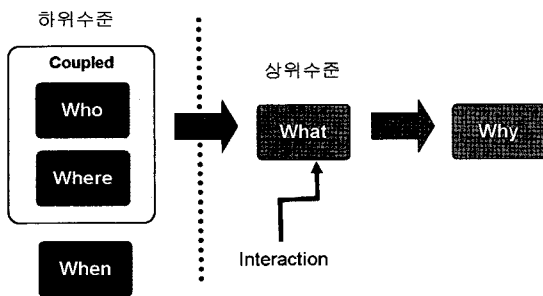
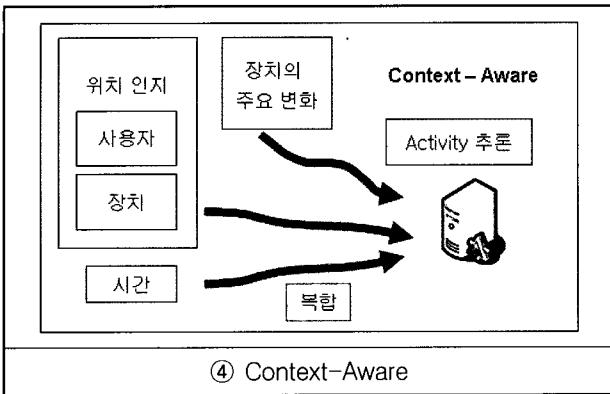


그림 4 하위 수준과 상위수준의 상황인지

자의 바깥에 표시를 하였으나, Activity를 추론하는데 중요한 정보로 활용된다.

목적(Goal)이 같은 일련의 작업을 묶어서, Activity로 정의하였는데, 이 Activity는 사용자의 행위나 장치의 동작으로 인해, 대상이 되는 사물의 변화가 수반되어야 한다. 사물의 변화에서 어떠한 Activity가 진행되었는지를 추론하기 위해서는, 특징적인 센서 정보들 간의 규칙(Rule)을 설정을 해 두게 되면, 작업들 간의 연관관계로서 해당 Activity를 추론하는 규칙을 시스템이 인지할 수 있다. 표 2는 사용자나 장치의 동작으로 인해 개별 작업이 규칙에 입력이 되어 있는 정보가 들어오게 되면 감지를 하는 것을 나타내고 있다. 표 3에서 장치의 주요 변화는 그림 3에 있는 상황모델에서 행위의 결과로 인식되는 것으로서, Activity를 추론하는 과정 및 결과인 상위 수준의 상황인지 (좁은 의미의 상황인지)로 설

명하고 있다. 이러한 상황인지의 과정은 잘 알려진 5W (Who, Where, When, What, Why)로 재해석할 수 있는데, 결국 Activity를 추론하는 것은 무슨 일이 일어나고 있는가(What is going on?)에 해당하므로, What을 추론한다고 말할 수 있다. 여기에서 Why는 의도를 추론하는 것인데, 본 논문에서는 고려하지 않는다(그림 4 참조)[10].

3. 오토노믹 컴퓨팅(Autonomic Computing)

이 장에서는 오토노믹 컴퓨팅의 속성과 IBM의 오토노믹 컴퓨팅 시스템 구조를 살펴본다. 오토노믹 컴퓨팅에서 '오토노믹(Autonomic)'은 생물학의 자율 신경계에서 사용되고 있는 '자율적'이라는 개념을 차용한 용어로서, 2001년 3월 IBM의 Paul Horn 부사장이 Harvard 대학에 있는 National Academy of Engineers에서 처음 그 개념을 소개하였다[11].

3.1 오토노믹 컴퓨팅의 속성

오토노믹 컴퓨팅은 시스템에 대한 복잡성이 증가하는 오늘날의 정보기술의 문제, 예상치 못한 외부 요인으로 인한 시스템 오류의 문제, 안정성을 보장하기 위한 시스템의 안정성 문제 등으로 시스템 관리자의 관여를 줄이며, 시스템 스스로 안정성과 보호를 기하기 위한 현실적인 필요성으로 연구가 시작되었다. 오토노믹 컴퓨팅을 위한 네 가지 속성으로 자가구성(Self-configuring), 자가치유(Self-healing), 자기최적화(Self-Optimization)와 자기보호(Self-protecting)를 정하여, 이를 기반으로 한 각 구성 컴포넌트들에 대한 기본 개념을 설정하였다.

3.2 오토노믹 컴퓨팅의 아키텍처

오토노믹 컴퓨팅의 근간이 되는 컴포넌트로는 크게 (1) Managed resource와 (2) Autonomic manager가 있다[12]. 그림 4는 제어를 받는 시스템 컴포넌트로서 Managed resource를 나타내고 있다. Managed resource는 센서와 Effector에 의해 제어되는 resource이다. 센서와 Effector의 조합은 Autonomic manager가 Managed Resource Touchpoint를 통해 조정이 된다. 주요 아이디어는 센서와 Effector가 각각 get operation과 set operation을 수행하며, 현재의 상태 정보를 관리

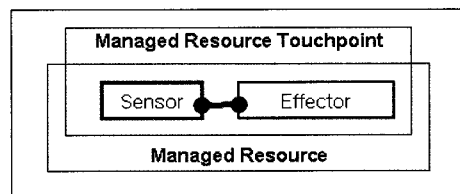


그림 5 센서와 Effector

하며, configuration의 변화에 대처하는 구조로 되어 있다. 여기에서 센서-Effector 기능을 웹 서비스로 구현을 하면 Managed resource touchpoint를 WSDL (Web Services Description Language)로 정의할 수 있다(그림 5 참조).

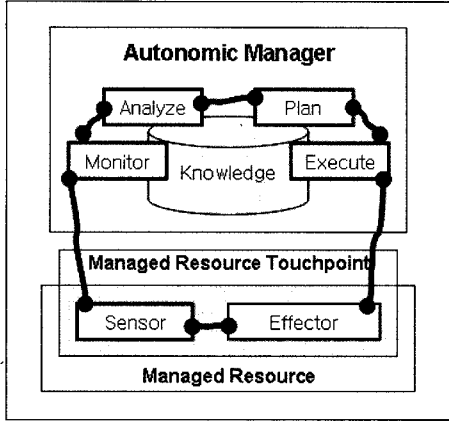


그림 6 Control loop

그림 6은 오토노믹 컴퓨팅의 Control loop를 나타낸 것으로서, 제어 흐름을 나타내는 것은 아니며, 시스템 구조를 나타내고 있다. 여기에서 Autonomic manager는 지식을 공유하는 네 개의 부분으로 나뉘어져 있는데, 그림에 나타나 있듯이 monitor, analyze, plan, execute 부분으로 되어 있다. 이 네 개의 부분을 연결하는 선은 공통의 메시징 버스(Messaging Bus)를 나타내는 것으로서, 비동기 통신을 통해 상호 Collaboration을 한다. 각 부분에 대해서 살펴 보면, ① Monitor 부분은 Managed resource로부터 수집한 정보를 필터링하고 관리한다 ② Analyze 부분은 복잡한 Situation에 대한 연관 관계와 모델을 제공한다. 이 부분은 Auto- nomic manager가 주변 환경을 학습하고, 미래의 Situation을 예측하는데 도움이 되는 메카니즘 구현을 가능하게 한다. ③ Plan 부분은 목표(Goal)에 해당하는 구체적인 Action 부분을 구체화한다. 이를 위해 정해진 정책(Policy)을 사용한다 ④ Execute 부분은 시스템의 갱신(Update)을 위한 Plan 부분의 실행을 제어한다.

4. 상황인지와 오토노믹 컴퓨팅과의 연계성

이 장에서는 오토노믹 컴퓨팅의 개념과 상황인지를 기반으로 한 지능형 서비스 시스템에의 적용에 관해 살펴본다. 상황인지가 센서 정보를 복합하여 하나의 구체적인 서비스 그룹으로 연결하는 메카니즘을 형성하기 위해서라고 하면, 오토노믹 컴퓨팅은 센싱 정보를 정해진 정책 기반으로 진단하고, 시스템의 현재 상태를 분석하며, 플래닝을 해서, 실행을 자율적으로 하는 시스템 구조를 갖고 있다. 이는 곧, 오토노믹 컴퓨팅은 센싱 정보

를 복합하여, 지식(Knowledge) 기반으로 현재의 시스템 상황을 인지하는 부분에서는 상황인지와 같은 개념이지만, 실행에 해당하는 부분은 서비스를 호출하는 부분으로서, 상황인지와는 구분되는 개념이다. 상황인지는 실제 공간에서의 다양한 센싱 정보를 기반으로 복합하는 센서 정보의 통합(Integration)이므로, 사용자나 장치의 위치 정보를 기반으로 하는 반면, 오토노믹 컴퓨팅은 가상 공간에서 현재의 시스템 Situation을 진단하고 분석하는 것이다. 그리고, 상황인지가 사용자나 장치의 인터랙션을 중요한 정보로 활용하는 반면, 오토노믹 컴퓨팅은 사용자의 관여를 줄이고, 시스템의 하부구조를 자율적으로 재구성하며 시스템에 치명적인 영향을 미치는 부분에 대해 시스템이 스스로 안정성과 보호를 지향하는 구조이다.

5. 상황인지 시스템의 구현 방법론

이 장에서는 유비쿼터스 컴퓨팅에서 상황인지 기반의 지능형 서비스 시스템을 구현하기 위한 구현 방법론을 제안한다. 상황인지를 기반으로 서비스를 제공하는 시스템이 구현되어 있다고 할 때, 시스템은 센서 정보를 복합하여 상황을 인지하고 나서야, 서비스를 제공하게 된다. 그러나 사용자의 인터랙션은 시스템의 서비스를 사용하는 것이기도 하지만, 시스템 입장에서는 새로운 센서 정보로 상황을 인지하여야 하므로, 상황인지와 서비스는 Feedback loop를 형성하게 된다. 이러한 상황을 인지하기 위해서는, 해당하는 서비스를 먼저 살펴보고 상황정보를 미리 정의하는 Reference 모델을 기반으로 해당 상황을 추론한다.

5.1 유비쿼터스 인터랙션의 디자인

유비쿼터스 컴퓨팅의 주요 특성으로는 보이지 않는 컴퓨팅 기술(Invisible Technology)이다. 이는 조용한 컴퓨팅 기술(Calm Technology)과 사라지는 컴퓨팅 기술(Disappering Technology)과 같은 개념선 상에 있는데, 여기에서 비가시성(Invisibility)은 인터랙션을 복합하여 추론하는 컴퓨팅 시스템을 인지하지 못하는 것으로서, 인터랙션의 대상은 사용자가 인지할 수 있어야 한다. 인터랙션의 대상과 사용 가능한 서비스를 인지하고 있을 때에, 유비쿼터스 시스템 환경에서 사용자는 신뢰하며 지능형 서비스를 제공받을 수 있다. 특히 개인의 ID(IDentification)와 같이 보호되어야 하는 정보는 사용자가 인지할 수 있는 시스템과의 특정한 인증 절차가 필요하다. 인터랙션 디자인은 시나리오를 설정하는 작업이 우선이 되며, 서비스를 제공할 수 있는 목록을 기반으로 Activity를 설정한다. 그림 7 ❶의 Meeting

room을 예제로 살펴보면, 서비스를 조명의 조도 조절로 정한다면, 이 때에 일어날 수 있는 Activity는 방안에 사람이 들어오거나, 나가는 경우가 있다. 그러나 회의가 진행되면, 조명의 조도를 조절을 하는 서비스를 제공하기 위해서는 방 안에서 회의가 진행 중인지 아닌지와 같은 상황을 인지하여야 한다. 결국, Activity를 Meeting, Empty, Enter와 같이 3가지로 설정을 할 수 있다.

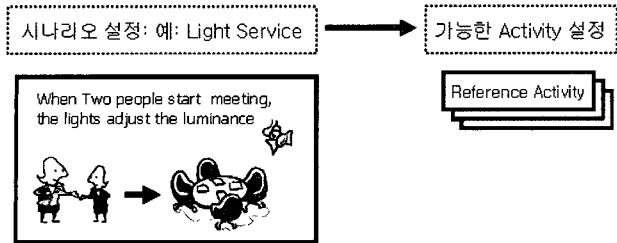


그림 7 ① 시나리오 설정

5.2 추론 엔진 설계

인지하려는 Activity 목록이 정해지면, 현재의 센서 정보들 간의 상관관계를 설정하여, 분류를 하는 작업이 필요하다. 센서 정보를 받아서 파일에 데이터를 쓰고 읽는 작업을 한다고 하면, 메타 데이터인 온톨로지를 미리 설정하는 것이 추론 시스템을 구현할 때에 유용하다[13]. 가령, protege 2000[14, 15]과 같은 온톨로지 에디터로 센서 정보를 분류하고 추론을 위한 센서 정보들 간의 관계를 미리 설정할 수 있다. 온톨로지가 정의되고 나면, 실제 센서 데이터를 복합하여 Reference Activity 목록 가운데에서 어느 Activity가 해당되는지를 시스템이 추론을 할 수 있도록 규칙을 설정해야 한다. 이러한 규칙을 설정하는데 있어서 잘 알려진 프레임워크로는 에이전트 기반의 JESS 엔진과 같은 소프트웨어를 활용할 수 있다[13, 16, 17](그림 7 ② 참조).

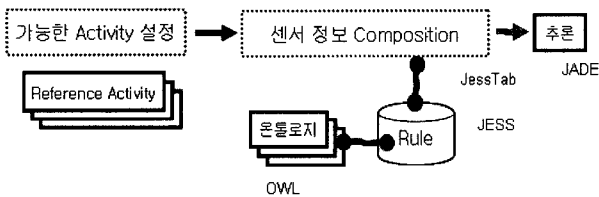


그림 7 ② 추론 엔진 설계

5.3 센서 정보의 복합

센서 데이터에 단순 반응을 하는 시스템은 엄밀한 의미에서 상황인지 기반의 시스템이라고 할 수 없다. 상황 인지는 일정한 시간에 특정한 장소에서 사용자의 행위나 장치의 동작과 같은 일련의 작업들을 묶은 Activity를 인지하는 것에 해당하므로, 먼저 여러 가지 센서 정보를 일정한 분류 기준으로 복합하여야 한다. 이렇게 센서 데

이터를 복합하는 것은 기존에 많이 연구가 되고 있는데, A.K. Dey는 GUI (Graphic User Interface) 기반의 Widget을 도입하였다[4]. 그러나 홈 네트워크 미들웨어인 UPnP와 같은 오픈 아키텍처를 활용하여 센서 장치를 개발하면, 센서 데이터를 추상화할 수 있는 투명한 구조로 구현을 할 수 있다. 그리고 UPnP는 가전 기기에 대한 제어(그림 8 참조)를 할 수 있는 서비스 목록을 Advertise 하며, 해당 서비스를 호출할 수 있기 때문에 센서 장치 뿐만 아니라, 서비스를 제공하는 장치에 대한 제어를 하나의 Control Point에서 모두 제어할 수 있는 장점이 있다.

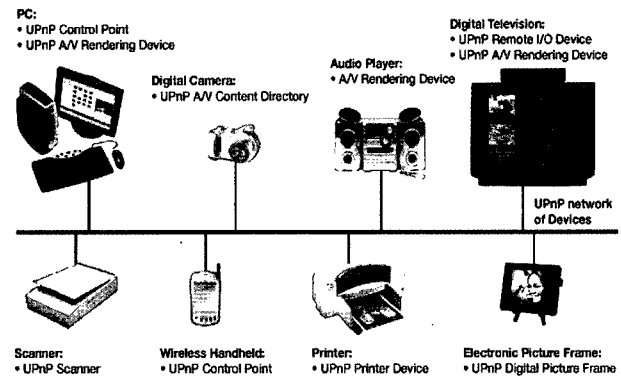


그림 8 UPnP 기반의 디지털 가전기기

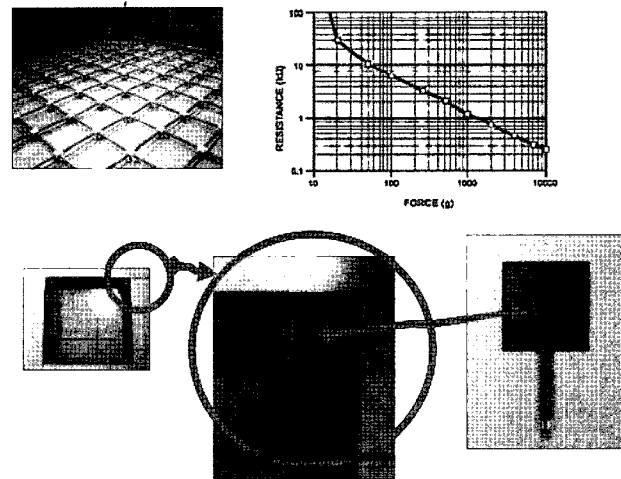


그림 9 바닥에 설치한 FSR Array

UPnP Control Point를 통한 센서 데이터를 복합하는 시스템의 실제 구현에 대해서 간단히 살펴보자. 하위 수준의 상황정보로서, 사용자의 Identity는 RFID와 같은 전자식별 장치로 인식을 하였거나, 비전을 통한 얼굴 인식으로 패턴인식을 하였다고 가정한다. 그리고, 장치가 UPnP 규격으로 구현이 되어 있다면 Control Point는 Device UUID를 검색함으로써, 해당 장치에 대한 Identity를 인식할 수 있다. 이 절에서는, 사용자가 있는 장소와 무게값을 인식하는 센서 장치를 UPnP로 구

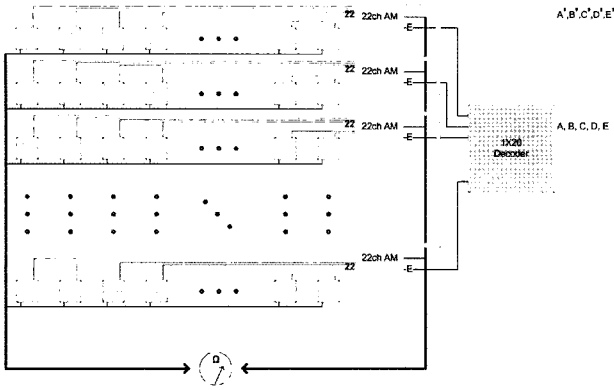


그림 10 FSR에 전류를 흘리기 위한 회로 구성

현하여, 그 데이터를 복합하는 시스템 구현에 대해 살펴 보자. 그림 9는 한국과학기술연구원의 미팅 룸의 바닥 면에 실제 압력을 센싱할 수 있는 센서 어레이를 가로 20줄과 세로 22줄을 설치한 상태를 나타낸다. 센서는 FSR (Force Sensor Resistor)로서, 힘이 가해지지 않은 상태에서는 저항이 무한대였다가, 센서에 힘이 가해지면 저항 값이 바뀌는 일종의 가변 저항이다. 이러한 가변 저항 440(20 x 22)개를 바닥 면에 설치하게 되면, 무게가 인식되는 위치와 그 값을 측정할 수 있다. 그림 10은 440개의 센서 하나하나에 전류를 순차적으로 흘리는 회로를 구성한 것으로 유연한 제어를 위해, PXA255 CPU와 같이 고속의 연산이 가능한 임베디드 장치를 활용하였다. 그림 10에서 A,B,C,D,E는 5비트의 제어 신호로서, 행(Row)으로 연결된 센서를 선택하는 신호이다. 그리고 A',B',C',D',E'은 5비트의 또 다른 제어 신호로서, 선택된 행에서 개별 센서를 선택하는 신호이다. 이 두 가지 신호를 이용하여 임베디드 장치는 제어를 하는데, 센서가 선택이 되어 전류가 흐르면 해당 전압을 측정하여 AD 컨버터를 거쳐, 컨버팅이 끝난 신호를 임베디드 장치에서 인터럽트 루틴을 통해 측정을 하는 구조로 되어 있다(그림 11 참조).

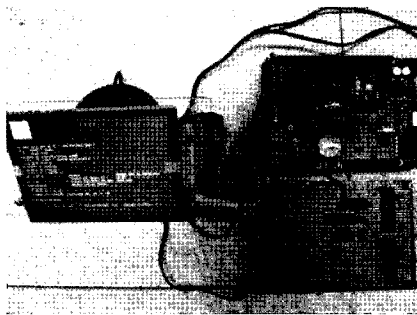


그림 11 임베디드 장치와 FSR과의 인터페이스

그림 12와 13은 각각, 사용자가 가로 2개와 세로 2개의 FSR Array가 놓여진 바닥면에 올라갔을 때 무게가 인식되는 위치와 그 무게값을 UPnP 서비스 형태로

제공하는 것을 보여준다. 그리고 센서 정보를 복합하는 역할을 UPnP Control Point에 구현할 수 있다(그림 14 참조).

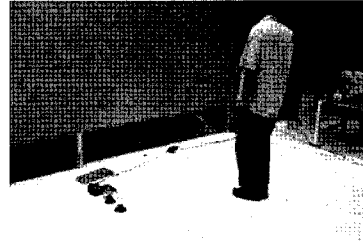


그림 12 가로2개와 세로2개의 FSR의 값 측정

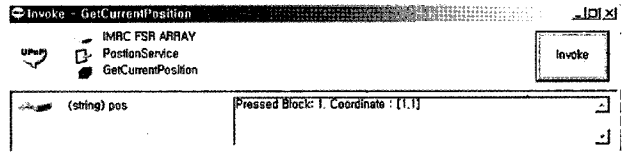


그림 13 무게를 인식한 위치와 측정값 서비스 (UPnP)

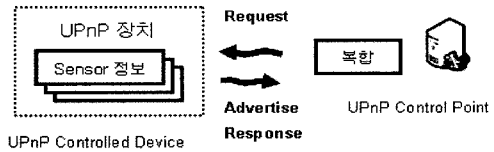


그림 14 UPnP 기반의 센서 정보의 복합

5.4 상황인지 기반의 오토노믹 서비스

시스템이 상황을 자율적으로 인지하여, 상황에 해당하는 미리 설정된 서비스 그룹을 자율적으로 호출하는 시스템을 살펴보자. 여기에서 자율은 오토노믹 컴퓨팅에서 말하는 '오토노믹'과 같은 개념을 지칭하는 것으로 사용하였다. 정책(Policy)을 설정한 것은 앞 장에서 살펴본 바와같이 Reference Context를 설정하는 것에 해당하며, 오토노믹 컴퓨팅에서의 지식은 상황정보를 복합한 결과를 일정한 자료구조나 데이터베이스에 저장하는 것에 해당하며, 본 논문에서는 온톨로지에 저장되는 센서 정보의 인스턴스 (Instance)들 간의 상관관계로 생각할 수 있음을 알아보았다.

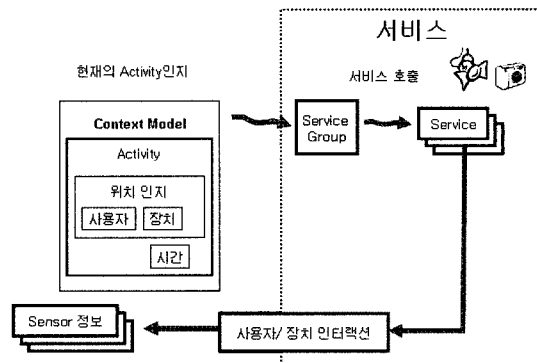


그림 15 자율적인 서비스 호출 시스템

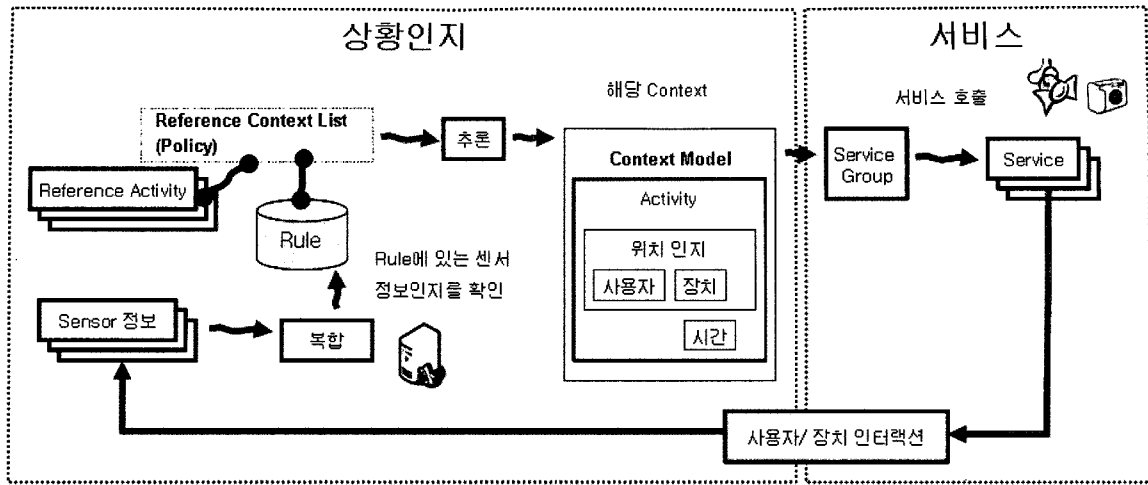


그림 16 상황인지와 지능형 서비스 간의 Feedback 시스템 구조

상황을 인지하면, 서비스 그룹으로 일대일 대응이 설정되어 있으면, 서비스 그룹 안에서 개별적인 장치의 서비스를 호출한다. 여기에서 사용자나 장치가 서비스를 호출하지 않고도, 자율적으로 시스템이 서비스를 호출하는 시스템을 구성할 수 있다. 이를 UPnP 아키텍처로 구현을 할 때에는 자율적인 Control Point를 구현하는 것을 의미한다. 사용자나 장치들 간의 인터랙션은 다시 센서에서 인지되어 Feedback Loop을 형성한다(그림 15 참조).

6. 결 론

본 논문에서는 상황인지의 개념을 하위 수준(사용자나 장치의 Identity, 사용자나 장치가 있는 위치, 시간)의 상황인지와 상위 수준의 상황인지로 나누어, 그 개념을 제안하였다. 하위 수준과 상위 수준을 나누는 기준으로는 Activity의 추론 여부였으며, Activity는 같은 Goal을 가지는 작업의 그룹으로 정의하였다. 상황인지와 서비스 간에는 Feedback loop을 형성한다는 것을 살펴보았는데, 전체적인 시스템 구조는 그림 16에 정리를 하였다. 그리고 오토노믹 컴퓨팅과 상황인지 간의 유사점과 차이점을 살펴보았다.

참고문헌

[1] 한국전산원, 정보화 정책이슈 04-정책-18 “u-Korea 구현을 위한 IT839 전략분석”, 2004.12.
 [2] P. Dourish, “What we talk about when we talk about context,” Personal and Ubiquitous Computing, Vol.8, pp.19-30, 2004.
 [3] B. Schilit and M. Themier, “Dissemination active map information to mobile

hosts,” IEEE Network Vol.8 Issue 5, pp. 22-32, 1994.

[4] A.K. Dey, “Providing Architectural Support for Building Context-Aware Application,” Ph.D Thesis, Georgia Institute of Technology, 2000.
 [5] M. Khedr and A. Karmouch, “Negotiating Context Information in Context-Aware Systems,” IEEE Intelligent Systems, Vol. 19, Issue 6, pp.21-29, 2004.
 [6] A. Ranganathan and R.H. Campbell, “An infrastructure for context-awareness based on first order logic,” Personal and Ubiquitous Computing, Vol.7 No.6, pp.353-364, 2003.
 [7] UPnP, <http://www.upnp.org>.
 [8] Jini, <http://www.jini.org>.
 [9] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” IEEE Computer, Vol.34, Issue 8, pp.57-66, 2001.
 [10] D.H. Kang, S.C. Ahn, H.D. Ko and W.D. Cho, “Context-Awareness for Ubiquitous Computing System,” 지능정보 시스템학회 추계 학술대회, COEX, 2004.11.
 [11] J.O. Kephart, D.M. Chess, “The Vision of Autonomic Computing,” IEEE Computer, Vol.36, Issue 1, pp.41-50, 2003.
 [12] B. Jacob, S. Basu, A. Tuli, and P. Witten, “First Look at Solution Installation for Autonomic Computing,” Chapter1. Autonomic Computing, 2004.

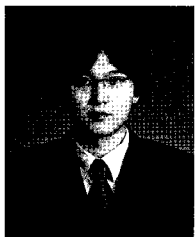
- [13] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakaraborty, "Intelligent Agents Meet the Semantic Web in Smart Spaces," IEEE Internet Computing, Vol.8, Issue 6, pp.69-79, 2004.
- [14] N.F. Noy, M. Sintek, S. Decker, M. Cruzbezy, R.W. Ferguson, and M.A. Musen, "Creating Semantic Web Contents with Protege-2000," IEEE Intelligent Systems, Vol.16, Issue 2, pp.60-71, 2001.
- [15] www.protege.stanford.edu.
- [16] JADE, www.jade.cselt.it.
- [17] H. Eriksson, "Using JessTab to Integrate Protege and Jess," IEEE Intelligent Systems, Vol.18, Issue 2, pp.43-50, 2003.

강 동 훈



1997. 2 포항공과대학교 기계공학과(학사)
 1999. 2 포항공과대학교 기계공학과(석사)
 1999. 2~2000. 6 삼성SDI 주임연구원
 2000. 6~현재 KIST 영상미디어 연구원
 관심분야: Robotics, 임베디드 디바이스,
 지능형 홈 네트워크 미들웨어
 E-mail : choco@imrc.kist.re.kr

안 상 철



2004. 2 경희대학교 컴퓨터공학과(학사)
 2004. 3~현재 연세대학교 컴퓨터공학과
 (석사과정)
 2004. 3~현재 한국과학기술연구원 학생
 연구원
 관심분야: HCI, 인공지능
 E-mail : prime@imrc.kist.re.kr

고 희 동



1989. 5 일리노이대학교 전산학과(박사)
 1988. 9~1990. 5 조지메이슨대학교
 전산학과 객원 조교수
 1990. 8~현재 한국과학기술연구원 책임
 연구원
 2001. 7~현재 영상미디어연구센터장
 관심분야: 가상현실, 인공지능, HCI
 E-mail : ko@imrc.kist.re.kr

권 용 무



1980. 2 한양대학교 전자공학(학사)
 1983. 2 한양대학교 전자공학(석사)
 1992. 8 한양대학교 전자공학(박사)
 1983. 1~현재 한국과학기술연구원 책임
 연구원
 관심분야: 영상기반 모델링, 메타데이터
 기반 콘텐츠저작, 디지털가상문화재,
 HCI
 E-mail : ymk@imrc.kist.re.kr

김 형 곤



1974. 2 한국항공대학교 전자공학(학사)
 1982. 8 Univ. of Kent 전자공학(석사)
 1985. 8 Univ. of Kent 전자공학(박사)
 1977. 1~현재 한국과학기술연구원 책임
 연구원
 관심분야: 인터랙티브 미디어, 몰입형 혼
 합 환경, 컴퓨터 비전 기반 콘텐츠
 생성, 멀티모달 인터랙션, 지능형
 공간, 디지털 라이프
 E-mail : hgk@imrc.kist.re.kr

조 위 덕



1977. 3~1981. 2 서강대학교 전자공학과
 (학사)
 1981. 3~1983. 2 한국과학기술원 전기
 및전자공학과(석사)
 1983. 3~1987. 2 한국과학기술원 전기
 및전자공학과(박사)
 1983. 3~1990. 3 금성전기(현 LG전자)
 기술연구소 DSP 연구실장
 1990. 3~1991. 10 한국생산기술연구원
 전자정보시스템연구부 팀장/조교수
 1991. 11~2003. 9 전자부품연구원 시스템연구본부 본부장
 2003. 1~현재 유비쿼터스컴퓨팅사업단 단장, 아주대학교 전자
 공학부 교수
 관심분야: 유비쿼터스 컴퓨팅/네트워크, 센서 네트워크, Post-
 PC(차세대 Smart PDA), Interactive DTV 방송
 기술, 고품질 홈서버/게이트웨이기술, 디지털방송/이
 동통신 연계 융합플랫폼기술, 무선인터넷응용기술
 E-mail : chowd@ajou.ac.kr