

유비쿼터스 컴퓨팅 환경에서 발생하는 에이전트간 충돌 해결 모델

이건수[♠] 김민구[♠]

[♠] 아주대학교 정보통신 전문 대학원
수원시 팔달구 원천동 산5번지

Tel: +82-31-219-2442, Fax: +82-31-219-1614, E-mail: minkoo@ajou.ac.kr

[♠] 아주대학교 정보통신대학 정보 및 컴퓨터 공학부
수원시 팔달구 원천동 산5번지

Tel: +82-31-219-2437, Fax: +82-31-219-1614, E-mail: minkoo@ajou.ac.kr

Abstract

오늘날 활발하게 이루어지고 있는 유비쿼터스 컴퓨팅 관련 기술 연구는 사용자가 시간과 장소에 구애받지 않고 네트워크에 접근해 다양한 컴퓨터 관련 서비스를 제공 받을 수 있는 방법에 초점을 맞추고 있다. 이처럼 시간과 공간의 한계를 뛰어넘은 네트워크로의 자유로운 접근은 일상 생활의 패러다임을 바꾸어 놓게 될 것이다. 유비쿼터스 컴퓨팅 기술을 통해 가장 큰 변화가 일어나는 분야는 일반 가정환경에서 일어나는 인텔리전트 홈 네트워크(Intelligent Home Network)라고 할 수 있다. 집에 돌아오면, 자동으로 문을 열어주고, 불을 켜주며, 놓쳤던 TV 프로그램을 자동으로 녹화해 놓았다가 원하는 시간에 보여주고, 적당한 시간에 목욕물을 미리 받아준다. 또한 집밖으로 나가기 전, 일기예보에 따라 우산을 챙겨주고, 일정을 확인시켜주며 입고 나갈 옷을 골라줄 수도 있다. 이 모든 일들이 유비쿼터스 컴퓨팅 기술이 가져올 인텔리전트 홈 네트워크의 모습이다. 그러나, 모든 사용자에게 효과적인 서비스를 제공하기 위해서는 홈 네트워크 상의 자원 관리에서 일어날 수 있는 에이전트들간의 자원 접근 권한 충돌을 효율적으로 방지할 수 있는 기술이 필요하다. 유비쿼터스 컴퓨팅 환경에서 자원관리 특성은 점유의 연속성, 자원 사이의 연관성, 그리고 자원과 사용자 사이의 연계성의 3가지 특성을 지니고 있다. 본 논문에서는 유비쿼터스 컴퓨팅 환경에서 일어날 수 있는 자원 충돌 상황을 효율적으로 처리하기 위한 자원 협상 방법을 제안한다. 본 방법은 자원 관리 특성을 바탕으로 시간논리에 기반을 둔 자원 선점과 분배 규칙으로 구성된다.

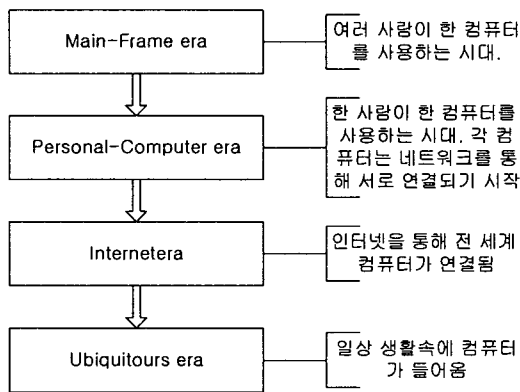
Keywords:

유비쿼터스 컴퓨팅, 에이전트, 협상 모델.

Introduction

1946년 미국 펜실베니아 대학에서 J.모클리와 J. P. 에커드가 에니악(ENIAC)이라는 최초의 컴퓨터를 발명된 이후, 컴퓨터는 오늘날 일상생활에 있어 떼어놓을 수 없는 삶의 일부가 되어 왔다. 처음 컴퓨터가 사람들에게 다가왔을 때, 하나의 컴퓨터를 여러 사람이 같이 사용했었다. 이때, 컴퓨터는 오늘날과 같이 일상생활의 일부로 사용되기 보다는 특정 연구분야 및 사업 환경에서 보다 좋은 성과를 올리기 위해 사용되었다. 이렇게 컴퓨터가 사용되던 시기를 가리켜 메인 프레임 시대라고 한다. 그 이후, 컴퓨터의 생산 능력의 발달로 컴퓨터의 가격은 보다 저렴해 졌고, 개개인의 사용자들은 자신의 컴퓨터를 소유할 수 있게 되었다. 컴퓨터는 보다 많은 대중에게 보급되었고, 점차 일상 생활에 컴퓨터가 차지하는 비중이 높아지게 되었다. 동시에, 각 컴퓨터는 내부 네트워크를 통해 서로 연결되기 시작하였고, 이로서 사용자는 자신의 컴퓨터를 사용하면서 네트워크로 연결된 다른 컴퓨터의 자원을 자유로이 사용할 수 있었다. 이 시기가 바로 퍼스널 컴퓨터의 시대로 일상으로 컴퓨터가 들어오기 시작한 시기였다. 그 이후, 인터넷의 보급을 통해, 컴퓨터는 명실상부한 삶의 동반자의 위치를 차지하게 되었다. 인터넷을 통한 온라인 채팅 사이트 등의 커뮤니티 활성화, 인터넷을 통한 온라인 게임의 붐과 그로 인한 온라인 수익성의 보장, B2C (Business to Customer) 혹은 B2B (Business to Business)모델의 성공에 따른 온라인 거래의 활성화 등은 컴퓨터를 일상생활을 영위함에 있어 가장 필수적인 요소로 만들었다. 그 결과 무인도에 가게 될 경우 가져가고 싶은 것 1순위가 인터넷이 되는 노트북이라고 할 정도이다. 이 시기가 바로 인터넷 시대이다. 그러나 이때까지의 컴퓨터와 인간의

관계는 사람이 컴퓨터를 찾아가는 구조였다. 이러한 사람과 컴퓨터의 관계 속에서 사용자는 자신이 필요로 하는 욕구를 충족시키기 위해 네트워크에 접근 가능한 컴퓨터를 찾아 접근해야만 했다. 단말기를 통해 자신이 원하는 것을 알리기 전까지, 사용자는 컴퓨터 기술이 제공하는 서비스를 받을 수 없었다. 이러한 기존 구조의 단점을 보완하기 위해 유비쿼터스 컴퓨팅 기술이 연구되었다. ‘도처에 있다’ 혹은 ‘언제 어디서나 존재한다’라는 의미의 유비쿼터스와 컴퓨터 기반의 서비스 제공을 의미하는 컴퓨팅의 합성어인 유비쿼터스 컴퓨팅 기술은 사용자가 컴퓨터나 네트워크를 의식하지 않은 상태에서 장소나 시간에 구애받지 않고 자유롭게 네트워크에 접근할 수 있는 환경 제공을 그 목표로 하고 있다. 이 같은 컴퓨터 관련 기술의 진보에 따라 인류의 삶이 변화된 단계는 <그림 1>에 나타나 있다.



<그림 1> 컴퓨터 혁명 과정

유비쿼터스 컴퓨팅 환경에서는 기존의 컴퓨터 기반 서비스 제공 방법과는 다른 방법으로 사용자에게 필요한 서비스가 선택되어 제공된다. 기존의 방법이 사람이 직접 특정 요구를 프로그램에 입력하면, 그 입력 값에 따른 처리 결과가 출력되었다. 이런 방법은 입력 값이 단순하고, 처리 방법이 명쾌한 경우, 빠른 결과를 얻어낼 수 있지만, 복잡한 환경이나, 사용자가 일일이 입력을 관리 할 수 없는 경우에는 적절한 기능 수행을 보장할 수 없었다. 이에, 유비쿼터스 컴퓨팅 환경에서 사용하는 서비스 제공 방법의 개괄은 다음과 같다. 우선 센서들을 사용하여 시스템이 관장하고 있는 환경 정보를 받아온다. 센서들을 통해 받아온 정보는 소위 컨텍스트 (context)라는 상황 정보로 바뀐다. 컨텍스트 정보는 센서들을 통해 받아온 단편적인 정보들의 의미를 부여해서 만드는 현재 시스템이 맞닥뜨린 환경의 상황을 의미한다. 이렇게 만들어진 환경 정보를 바탕으로 컴퓨터는 현재 상태를 이해하고, 이 상태에서 사용자의 요구를 예측하여 사용자가 먼저 무엇인가를 요구하기 전에, 그

서비스를 가지고 사용자에게 접근하게 된다. 이처럼, 사용자의 의도를 예측하여 원하는 서비스를 갖고 먼저 접근하는 방법이 유비쿼터스 컴퓨팅의 서비스 제공 방법이다. 그러나, 이 같은 사용자 요구 처리 방법에는 커다란 문제점이 하나 존재하는데, 자원 접근 충돌 가능성이 그것이다.

유비쿼터스 컴퓨팅의 시작은 주어진 환경을 지각(sensing)하는 것에서 시작한다. 그리고, 그렇게 읽어 들인 환경의 현재 상황을 파악하여 적절한 작업을 수행하게 된다. 이렇게 작업이 수행되고 나면, 그 작업만큼의 환경의 변화가 일어난다. 만일 특정 환경에 단 하나의 사용자가 존재하는 경우, 그 환경은 그 사용자의 행위에 의해서만 변명이 일어나고, 그 환경은 그 사용자에게만 서비스를 제공하면 되는 것이기 때문에 어려움이 없지만, 만일 둘 이상의 사용자가 존재하는 경우, 상대의 행동에 의해, 원치 않는 상황 변화가 일어날 수 있게 된다. 더욱이, 나의 요구와 다른 사람의 요구가 충돌이 일어나는 경우가 발생할 수도 있고, 요구에 대한 충돌이 일어나지 않는 경우에도, 내가 사용하려는 자원을 다른 사람이 미리 선점하고 있는 경우, 나에게 대한 서비스는 지연될 수 밖에 없다. 이러한 문제점은 동일 환경을 여러 사용자가 같이 사용하는 경우 발생하는 문제이다. 가령, 멀티 프로세싱 작업을 수행하는 운영체제에서도 이와 같은 문제에 당면하고 있다. “동시에 여러 작업에 대한 요청이 들어오는 경우, 어떻게 자원 분배를 하여 들어온 요청을 처리해 줄 것인가?”라는 문제는 운영체제의 오랜 고민이었고, 여러 해결 방법들이 고안되어 왔다. 그러나, 운영체제가 멀티 프로세싱 작업을 수행하기 위해 사용하는 자원 관리 방법은 유비쿼터스 컴퓨팅 환경에 적용될 수 없는데, 자원의 특성이 다르기 때문이다. 텔레비전이 하나 있는 방에 서로 다른 채널을 보고 싶어하는 두 사람이 있다고 가정하자. 이 때, 0.5초 간격으로 두 채널을 번갈아가며 보여준다고 이 두 사람의 욕망이 충족된다고 말 할 수는 없을 것이다.

본 논문에서는 유비쿼터스 컴퓨팅 환경에서 자원 충돌이 발생할 수 있는 경우를 살펴보고, 이러한 자원 충돌을 해결하기 위한 방법을 제안하고자 한다. 2장에서는 본 논문에서 제안하는 문제 해결 방법에 연관된 관련 연구들에 대하여 살펴보고, 3장에서는 제안된 방법의 구체적인 모습을 설명한다. 이렇게 제안된 방법은 4장의 실험을 통해 그 유효성을 검증하고, 그 결과를 확인해 본다. 그리고 5장에서 본 논문의 결론을 내리도록 한다.

Related research

본 논문은 유비쿼터스 컴퓨팅 기반의 인텔리전트 홈 네트워크 환경에서 발생하는 자원 충돌 문제를 해결하는 방법을 제안하고 있다. 이에 대한 관련

연구로 인텔리전트 홈 네트워크의 기반 기술이 되는 유비쿼터스 컴퓨팅 과 인텔리전트 홈을 소개하고, 본 연구에서 제안하는 인텔리전트 홈 네트워크 구축 및, 그 환경에서 발생하는 자원 충돌 방지를 위한 기술로, 멀티에이전트 시스템, 에이전트들 사이에 발생하는 협상 해결 방법, 각 에이전트가 주어진 작업을 수행하기 위한 추론 방법의 하나인 규칙기반 처리방법과, 규칙을 생성하기 위한 논리로서 시간 논리를 설명한다.

Ubiquitous Computing

‘도처에 있다’ 혹은 ‘언제 어디서나 존재한다’ 라는 의미를 지니고 있는 유비쿼터스는 88년 미국 제록스 팰로앨토연구소의 마크 와이저 소장에 의해 처음 사용되었다. 이렇게 사용된 유비쿼터스라는 단어는 사용자가 컴퓨터나 네트워크를 의식하지 않는 상태에서 장소에 상관 없이 자유롭게 네트워크에 접속할 수 있는 환경을 의미한다. 이러한 환경을 구축하기 위해 전자기기를 비롯한 모든 사물에 컴퓨팅 기능과 통신 기능을 부가한다. 유비쿼터스 컴퓨팅 환경 안에서 사용자에게 적절한 서비스를 제공하기 위해서는 다음의 방법을 따른다.

1. 사용자의 육체 및 감정 상태에 대한 정보 수집
2. 수집된 정보를 분석
3. 분석에 기반한 적절한 서비스 제공
4. 1번의 반복

유비쿼터스 컴퓨팅은 이 같이 반복되는 작업을 통해, 사용자의 상태 정보를 감지해 무엇이 필요한지를 추측하여 그것에 맞는 최적의 환경을 조성해줄 수 있다. 이 때 사용되는 상태 정보는 사용자가 무엇에 집중하고 있는지, 어떤 감정을 느끼고 있는지, 무엇을 하고 싶은지, 과거에 어떤 행동을 했는지에 대한 정보의 집합으로 여러 센서들의 감지해 낸 정보를 기반으로 만들어진다. 어떤 정보를 감지하고 감지한 정보를 통해 현재 상태를 파악해 내고, 파악한 현재 상태에서 사용자가 필요로 하는 서비스를 찾아내, 그 서비스를 어떻게 제공할 것인가가 유비쿼터스 컴퓨팅의 최종 목적이다 [2,3].

Intelligent Home

사용자의 의도를 파악해, 서비스 받기를 원하는 사용자에게 컴퓨터가 먼저 다가가는 유비쿼터스 컴퓨팅은 실생활의 여러 분야에서 응용되어 사용될 수 있다. 그 중, 인텔리전트 홈은 유비쿼터스 컴퓨팅을 통한 자동화된 가정 환경을 제공해 준다. 인텔리전트 홈은 가족 구성원을 일상의 가사에서 해방시켜 준다. 자동으로 집안 온도를 맞춰주고, 환기를 시켜주며 빨래가 차면 세탁해주고, 원하는 텔레비전 프로그램을 녹화해주고, 목욕물을 받아주며, 설거지를 해준다. 과거의 가정이 주부에게 전달된 가사를 통해 삶의 휴식처가 될 수 있었다면, 인텔리전트 홈은 모든 가족 구성원들에게 휴식처가

되어 줄 수 있다. 가사로부터의 해방은 오늘날 유비쿼터스 컴퓨팅 기술에 의해 현실화 되어가고 있다 [4,5].

Multi-Agent System

멀티 에이전트 시스템(MAS)은 하나의 에이전트로 해결하기 어려운 복잡한 문제를 처리하기 위해 여러 에이전트들의 협력을 구하는 방법이다. 이 같은 MAS은 단순한 에이전트의 협력을 통해 문제를 해결하려 하기 때문에 각각의 에이전트는 혼자서 주어진 작업을 처리할 수 없고, 시스템에 대한 일부분의 정보만을 갖는다는 특성을 갖고 있다. 이는 곧, MAS에서 주어진 작업을 처리하기 위해서는 일에 대한 적절한 분배, 지식의 공유, 역할 충돌 방지, 작업 결과의 통합과정을 필요로 한다. MAS는 유비쿼터스 컴퓨팅이 요구하는 분산된 여러 센서들의 정보를 수집 관리하고 그것을 바탕으로 상태 정보를 생성해 적합한 서비스를 선택 제공해주는 일을 효과적으로 수행할 수 있는 체제를 갖고 있다. MAS를 위한 시스템 구조로는 Object Manager Group (OMG)' s model, Foundation for Physical Agents (FIPA)' s model, Knowledgeable Agent oriented System (KaoS)' s model 과 General Magic' s model 등이 이미 제안되어 있다. 이들 각각의 멀티 에이전트 시스템 모델은 작업 에이전트들의 협력을 통해 주어진 과제를 수행할 수 있는 기본 플랫폼을 제공해 주고 있다 [6].

Resource Negotiation

한정된 자원을 여러 시스템이 공유하기 위해서는 누가 언제 어떤 자원을 먼저 사용할 것인가에 대한 협상 방법이 필요하다. 자원 협상 방법은 그 자원이 어떤 특성을 갖고 있는가에 따라 달라진다. 가령 멀티 프로세싱 작업의 경우 시분할 처리 방법을 사용해 여러 사용자가 CPU의 시간 자원을 나눠 사용할 수 있다. 웹 서버의 경우, 접속자 수가 증가함에 따라 미리 사이트를 운영하는 것처럼, 자원의 양을 늘리는 방법을 택할 수도 있다. 자원 협상을 위해서 어떠한 방법을 사용할 것인가 하는 것은 그 자원이 어떠한 특성을 갖고 있는가에 따라 달라진다 [7].

Rule-based System

규칙 기반 시스템은 주어진 도메인에서 지식을 표현하는 방법의 하나로, 기존의 사실 현상들의 집합으로 지식을 표현하는 방법과는 달리, 각기 다른 환경에서 특정 행동의 수행을 지정하거나, 특정 행동의 실행을 금지하는 규칙들로 지식을 표현한다. 이렇게 지식을 표현하기 위한 규칙들은 IF-THEN 형식을 갖고 있는데, 규칙이 적용가능한 한 상황을 판단하기 위한 조건 절과, 그 조건이 충족되었을 경우, 실행해야 할 행동을 표현하는 행위 절로 이루어져 있다.

생성된 규칙들을 바탕으로 현재 상황에서 어떤 행위를 수행해야 하는가를 결정 짓는 것은 지금 상황이 갖고 있는 규칙들 중에 어느 규칙의 조건절과 일치하는지를 판단하는 것이 중요하다. 규칙 기반 시스템의 경우, 정해진 규칙에 의해서만 추론이 진행된다는 점에서 시스템이 능동적인 변화에 쉽게 적응하기 어렵다는 단점이 있다. 그러나, 모델을 단순화 시킴으로써 추론의 과정을 쉽게 볼 수 있고, 구현 및 수정이 용이하다 [8].

Temporal Logic

시간 논리는 1960년대 발표된 시제논리(Tense Logic)의 확장으로 논리체계 안에서 시간 개념을 갖고 있는 정보를 표현하기 위한 방법으로 "항상 (Always)", "때때로 (Sometimes)" 그리고 "절대로 (Never)"의 3가지 개념을 기본 연산자로 사용하여 단일 정보를 표현한다. 이러한 방법으로 시간의 요소를 획득한 단일 명제들은 서로 다른 정보들과 연관 관계를 갖게 된다. 정보들 사이의 관계는 "~이전 (before)", "~와 동시 (equal)", "~의 바로 뒤 (meet)", "~와 중복 (overlaps)", "~의 동안 (during)", "~와 동시에 시작 (start)", "~와 동시에 종료 (finish)"의 7개로 표현된다. 이상의 연산자들을 통해 각 현상들은 $t \leq t' \in (T, \leq)$ 의 집합으로 표현될 수 있고 이는 곧 $t \in T$ 인 사건이 일어난 뒤 $t' \in T$ 인 사건이 일어났음을 의미한다. 이처럼 시간 요소를 논리에 적용함으로써, 특정 현상이 참 또는 거짓이 되기 위해 다른 현상과의 연속적인 발생과정을 명시할 수 있고 이는 곧 각 현상들이 서로 어떤 연결 관계를 갖고 있는가를 표현할 수 있다. [9].

Approach and Methods

본 논문에서 제안하는 인텔리전트 홈 서비스는 홈 아키텍처와 제안된 아키텍처 상에서 발생하는 에이전트간 자원 충돌을 방지하기 위한 방법으로 구성되어 있다.

Intelligent Home Architecture

인텔리전트 홈 네트워크를 구현하기 위한 구조로 본 논문에서 제안하는 방법은 <그림 2>와 같다. 홈 네트워크는 센서 네트워크, 퍼스널 에이전트 네트워크, 홈 서버 네트워크, 그리고 가전 기기 네트워크의 4가지 하위 네트워크로 구성되어 있다. 각각의 네트워크가 담당하는 작업은 다음과 같다.

1. Sensor Net

가정 환경 정보를 수집하여 데이터 베이스에 저장한다.

2. Home-Appliance Net

집안 내 존재하는 가전 기기들의 네트워크로, 각각의 가전 기기들은 내장 에이전트를 통해 관리되고, 가전 기기 네트워크를 담당하는 중개자의

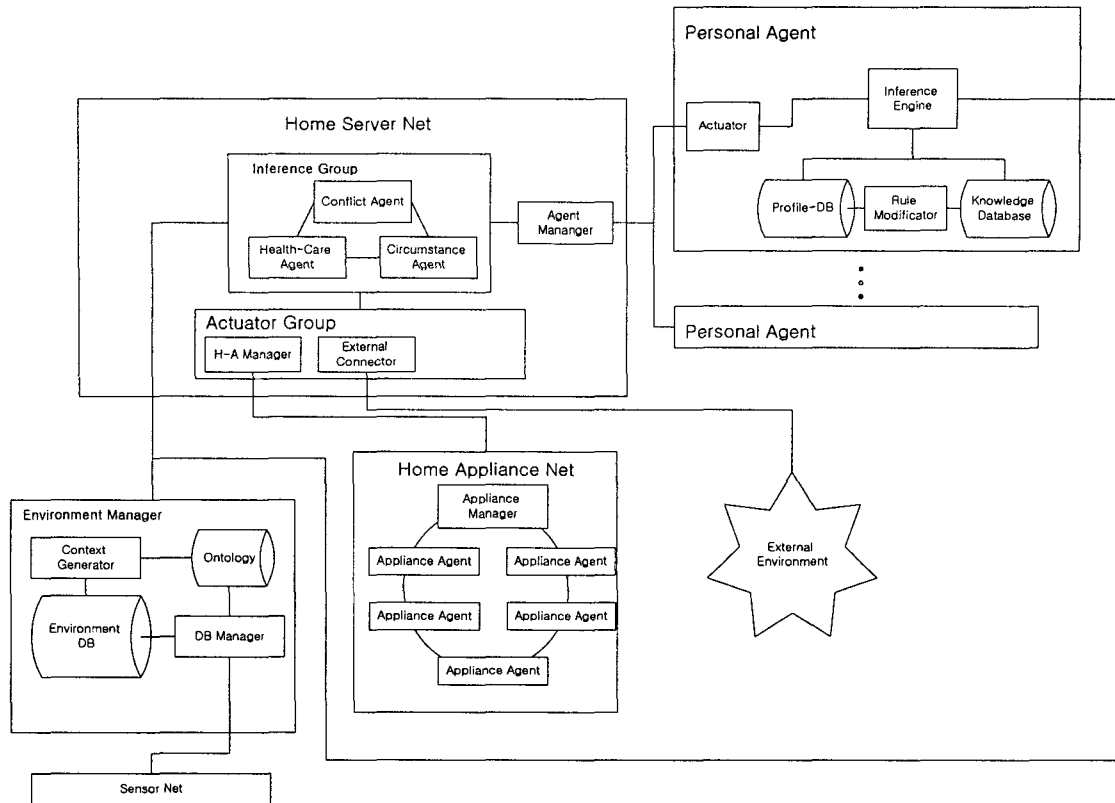
역할을 하는 '기기 관리자(Appliance Manger)'의 관리를 받는다.

3. Personal Agent Network

각 가정 구성원들에게 할당되어 개인화된 서비스를 제공해주는 에이전트들의 네트워크로, 개인 프로파일에 기반하여 각 사용자와 인텔리전트 홈 네트워크와의 연결 고리 역할을 한다.

4. Home Server Network

퍼스널 에이전트들이 개인화된 서비스를 제공하는 과정에서 개별 에이전트의 능력으로 처리할 수 없는 작업들을 담당하여 처리하는 에이전트들의 집합으로 보다 전문화된 서비스를 제공해 준다. 홈 서버 네트워크는 크게, 충돌 해결 에이전트(Conflict Agent) 건강 관리 에이전트(Health-Care Agent) 그리고 환경 에이전트(Circumstance Agent)의 에이전트들로 구성되어 있다. 충돌 해결 에이전트들은 퍼스널 에이전트들이 각기 담당하고 있는 사용자들의 개인화된 서비스를 제공하는 과정에 있어서 충돌이 발생하는 경우, 에이전트들 사이의 충돌을 중개해주는 역할을 한다. 건강 관리 에이전트는 퍼스널 에이전트가 제공하는 서비스 보다 전문화된 건강 관리를 위한 에이전트로, 가령 사용자의 건강 센서로부터 이상 증후가 들어온 경우, 퍼스널 에이전트가 그 이상 증후에 적합한 대응을 할 능력이 없을 경우, 보다 전문화된 서비스를 제공하기 위해 그 부분의 서비스 관리 제어 권을 받아와 처리하는 역할을 한다. 인텔리전트 홈 네트워크에서 처리해야 하는 서비스들이 보다 세분화되고 각 세분된 분야에 대한 전문성이 요구된다면, 건강 관리 에이전트처럼 특정 분야에 관한 전문화된 지식을 갖고 있는 에이전트의 종류가 늘어나게 된다. 다만 <그림 2>에 나타난 인텔리전트 홈 네트워크에서는 전문화된 지식이 필요한 도메인이 건강 관리로 한정되어 있기 때문에 건강 관리 에이전트만 존재하고 있다. 환경 에이전트 역시 가정 환경에 대한 최종 관리자 역할을 수행하고 있다. 이들 4개의 하위 네트워크들이 서로 밀접한 연관관계 속에서 동작하게 된다. 센서 네트워크는 가정 환경 정보를 지각하여 그 정보를 환경 관리자(Environment Manager)에게 전달한다. 환경 관리자는 자신이 갖고 있는 온톨로지를 바탕으로 센서들로부터 받아들인 환경 정보를 정리하여 데이터 베이스에 저장한다. 이렇게 저장된 환경 정보들은 다른 에이전트들의 요청이 들어오는 경우, 적절한 컨텍스트 정보로 변환되어 응답해준다. 환경 관리자에게 컨텍스트 정보를 요청할 수 있는 에이전트들은 퍼스널 에이전트들과 전문화된 서비스를 위한 홈 서버 네트워크의 구성원들이다. 각 퍼스널 에이전트들은 자신이 담당하고 있는 사용자의 특성화된 서비스를 제공하고, 각 사용자 역시 자신의 요구를 자신의 퍼스널 에이전트를 통하여 요청하게 된다.



<그림 2> 인텔리전트 홈 네트워크 구조도

사용자의 요청은 퍼스널 에이전트를 통하여 처리되고, 그 결과는 홈 서버 네트워크로 넘어간다. 홈 서버 네트워크 안의 에이전트 매니저(Agent Manager)는 각 퍼스널 에이전트로부터 넘어온 추론 결과가 전문화된 추가 서비스를 필요로 하는 경우, 네트워크 안의 전문화 에이전트들에게 전달하고, 그렇지 않은 경우 직접 발동기 그룹(Actuator Group)에 전달한다. 전문화 에이전트들에게 넘어간 서비스의 경우에도 추가 추론 과정을 거치게 되고 그 결과는 발동기 그룹에 전달된다. 발동기 그룹은 가전 기기 네트워크 관리자(H-A Manager)와 외부 환경 접근자(External Connector)로 이루어져 있다. 가전 기기 네트워크 관리자는 최종 결정된 결과를 실제 가전 기기에게 적용하여 일을 수행하는 역할을 하고, 만약 최종 서비스 요청 결과가 가정 환경 외부의 도움을 필요로 한다면, 외부 환경 접근자에게 보내져 외부 환경과 통신하게 된다.

Suggested Method

Resource Collision

인텔리전트 홈 아키텍처에서 살펴 보았듯이, 각각의 가족 구성원은 자신의 에이전트를 할당 받고, 그 에이전트를 통해 인텔리전트 홈 서비스를 제공 받는다. 외출하면서, 원하는 텔레비전 프로그램을 에이전트에 입력하면, 에이전트는 VCR과 연동하여, 그 프로그램을 녹화하고, 사용자가 집으로 돌아오면, 녹화된 프로그램을 시청할 것인가를 물어본 뒤,

적절한 표현장치(Displayer)를 찾아 보여준다. 이 같은 가정 환경 조정은 그 집에 거주하고 있는 사람이 혼자일 경우, 최상의 서비스를 제공 할 수 있지만, 거주자의 수가 2명 이상으로 늘어나게 되는 경우, 가정 환경에 대한 접근 충돌이 일어나게 된다. 가령, 아버지가 6시에 방영하는 TV 프로그램을 녹화하고자 한다고 하자. 아버지의 에이전트는 아버지가 설정한 욕구를 충족시키기 위해 6시에 VCR을 가동해, 설정된 프로그램을 녹화하려고 한다. 이때, 어머니가 퇴근하면서 빌려온 비디오 테이프를 보려고 할 때, VCR은 아버지의 에이전트가 할당한 녹화 작업을 수행해야 할 것인지, 아니면 어머니의 에이전트가 요구하는 특정 비디오 테이프의 재생을 수행할 것인지 결정 할 수 없게 된다. 물론, 컴퓨터에 TV 카드를 달아서, 컴퓨터를 사용해 방송 프로그램을 녹화하는 방법이 있을 수도 있다. 유비쿼터스 컴퓨팅 환경에서 특정 작업을 수행하기 위한 고정된 방법이 있는 것은 아니기 때문에 이처럼, 자원에 대한 접근 충돌이 일어났을 경우, 대용 자원을 찾아서 그 요구를 가장 충실하게 수행할 수 있는 다른 방법을 찾을 수도 있다. 그러나 역시 이런 경우에도 문제는 존재한다. 아버지가 거실에서 벽걸이 텔레비전으로 특정 방송을 시청하고 있다고 하자. 이때, 어머니가 특정 드라마를 보기 위해 급히 집에 귀가 했다. 어머니의 에이전트는 어머니가 요청한 드라마를 보여주기 위해 거실의 대형 텔레비전을 사용하려고 하지만, 그 텔레비전은 이미 아버지의 에이전트에

의해 사용되고 있다. 그래서, 어머니의 에이전트는 대응 표현 장치를 검색하여, 주방에 있는 조그만 모니터를 찾아내 그 모니터로 드라마를 보여준다고 하자. 과연 어머니는 밖에 나갔다 들어와서는 주방에 달려있는 조그만 모니터를 통해, 그것도 서서, 이어폰을 귀에 꼽고 보고 보면서 유비쿼터스 컴퓨팅 기술에 감사할 수 있을까? 사람의 욕구는 무척이나 다양하고 그 충족의 조건이 지극히 예민하기 때문에 단순히 사용자에게 의해 요청된 작업이 수행되었다는 것만으로 사용자를 만족시킬 수 없다. 더구나 운영체제가 멀티 프로세싱 작업을 수행하기 위해 컴퓨터 자원을 사용하는 방법을 사용하기에는 사람과 컴퓨터의 차이가 너무 크다. 가령, 아버지와 어머니가 서로 다른 채널을 동시에 보고 싶어한다고 해서, 0.5초 간격으로 채널을 바꿔가면서 두 명에게 동시에 서비스를 제공하고 있다고 말할 수 없는 것과 마찬가지로이다. 이처럼, 유비쿼터스 컴퓨팅 환경에서 자원관리는 <표1>에서 보여지는 3가지 특징을 갖고 있다.

<표 1> 유비쿼터스 컴퓨팅 환경의 자원 속성

특성	의미
점유의 연속성	한번 점유된 자원은 주어진 작업을 완료하기 전까지 독점적으로 사용된다.
자원 사이의 연관성	각각의 자원들은 서로 밀접한 연관을 갖고 있다.
자원과 사용자 사이의 연계성	하나의 자원 사용으로 인한 환경 변화는 그 환경을 공유하는 다른 사용자에게 영향을 미친다.

첫 번째 속성인 점유의 연속성은 일단 사용자의 요청에 의해 하나의 작업이 할당된 자원은 그 작업을 수행하기 전까지는 다른 사용자의 요구에 응답할 수 없다. 가령, 사용자에게 의해 특정 채널을 보여주고 있는 텔레비전은 그 방송이 끝나기 까지, 혹은 그 사용자가 더 이상 텔레비전을 시청하지 않을 때까지 다른 사람이 원하는 방송을 보여줄 수 없다. 사용자들 사이의 협상을 통해 다른 채널로 바꾸기로 결정 되었다면, 그것은 이미 기존 점유자가 자신의 권리를 포기 혹은 양도한 결과이다. 두 번째 속성인 자원 사이의 연관성은 주어진 작업을 수행함에 있어서, 자원들 사이의 밀접한 연관 관계가 존재함을 의미한다. 가령 동일 공간상에 존재하는 텔레비전과 오디오는 동시에 가동될 수 없다. 텔레비전을 시청하는 동안, 오디오가 녹음의 기능을 수행할 수는 있지만, 특정 사용자가 텔레비전을 시청할 때, 다른 사용자가 그 옆에서 라디오를 들을 수는 없다. 이는 각 자원들의 기능이 시간과 공간에 따라 일으킬 수 있는 충돌 가능성을 나타낸다. 세 번째 속성인 자원과 사용자 사이의 연계성은 유비쿼터스 컴퓨팅 환경의 근본인 자원 사용을 통한 환경 변화에 기반하고 있다. 가령, 한 사용자가 실내 온도를 낮추기 위해 에어컨을 사용하면, 그 공간의

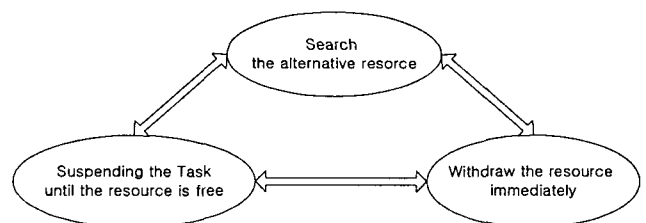
상태가 변화하게 된다. 같은 곳에서 다른 사용자는 원치 않는 환경 변화에 의해 추위를 느낄 수도 있게 된다. 이 같은 자원관리의 특성은 자원에 접근하여 주어진 작업을 수행하는 과정에서 여러 충돌 가능성을 내포하고 있다. <표2>는 이러한 특성에 기반한 충돌 형태를 보여주고 있다.

첫 번째 유형이 충돌이 발생하는 원인은 사용자의 특정 요구를 수행하기 위해 필요한 자원이 이미 다른 사용자에게 의해 점유되어 있는 상황이다. 이 경우 요청된 작업을 수행 할 수 없다. 두 번째 충돌은 사용자의 요청을 처리하기 위해 여러 자원의 연속적인 소비를 필요로 하는 경우, 중간 단계의 자원이 이미 사용 중일 때 발생한다. 이 경우는 작업 단계의 변화를 통해 충돌을 해결 할 수 있다. 세 번째 충돌은 서로 다른 사용자의 서로 다른 요청의 결과에서 발생하는 충돌로 특정 요청의 결과가 다른 사용자의 요청 결과에 반하는 경우이다. 마지막 충돌 유형은 서로 다른 사용자의 동일 요청에서 발생하는 충돌로 특정 사용자의 요청을 수행하는 작업이 다른 사용자의 요청과 일치하는 경우, 환경을 지각하는 센서들이 첫 번째 요청 처리 과정을 두 번째 요청 처리로 오해하는 과정에서 발생한다. 이러한 충돌을 해결하기 위한 방법은 다음 절에서 설명한다.

<표2> 자원 속성에 따른 자원 충돌 형태

자원 속성	충돌 형태
점유의 연속성	한 사용자의 요구를 충족시키기 위한 자원을 다른 사용자가 먼저 사용하고 있는 경우
자원 사이의 연관성	한 사용자의 요구를 충족시키기 위해 다른 사용자의 요구를 무시해야 하는 경우
자원과 사용자 사이의 연계성	한 사용자의 요구를 충족시키기 위한 환경의 변화가 다른 사람의 요구를 침해하는 경우 동일한 요구가 서로 다른 사용자에게서 발생한 경우

Anti-Collision Method



<그림 3> 충돌 해결 방법

충돌이 발생하는 경우, 발생한 충돌을 해결하는 방법은 <그림 3>에 나타난 것과 같다. 첫 번째 방법은 사용하려는 자원에 대한 점유가 풀릴 때까지 기다리는 것이다. 두 번째 방법은 다른 사용자에게

의해 점유된 자원을 대체할 수 있는 다른 자원을 찾아서 사용하는 것이다. 마지막 방법은 다른 사용자에게 의해 점유된 자원을 즉각 회수하여 처리하는 방법이다. 이 세 가지의 방법 중 어느 것을 언제 사용할 것인가는 충돌의 형태, 요청의 종류, 그리고 요청한 사용자의 속성에 의해 결정된다. 가령 화장실이 급한 사용자의 요청은, 이미 화장실에서 세면을 하고 있는 사용자의 자원 사용권을 즉각 받아들일 수 있지만, 텔레비전을 보고 싶은 사용자의 요청은 이미 다른 방송을 보고 있는 사용자에게서 리모콘을 뺏아올 수는 없다. 또한 충돌이 일어나는 경우 대안을 찾는 요청이어도 그 대안의 결과가 자원을 기다리는 것보다 만족도가 현저히 낮아진다면, 기다리는 편이 현명한 판단이다. 또 기다리는 것보다 다른 대체 자원을 사용하여 즉각적인 서비스를 받는 것이 만족도가 높다면, 기다리는 편보다 대체 자원을 사용하는 편이 현명한 판단이 된다. 이는 전적으로 사용자가 어떤 요청을 하는지, 요청한 사용자가 어떤 성격의 소유자인지에 따라 달라질 수 있다. 이를 위해 본 논문에서 제안한 방법은 다음의 3가지 정보를 이용한다.

<표3> 충돌 해결을 위해 필요한 정보의 종류

정보 종류	의미
사용자 프로파일	서비스에 대한 사용자의 만족도를 측정할 수 있는 선호도 및 사용자 정보
서비스 속성	현재 진행중인 서비스 별 프로파일 컨텍스트
충돌 해결을 위한 규칙	서비스 속성에 기반한 충돌 해결 방안 선택을 위한 시간논리 기반의 규칙

사용자 프로파일은 사용자가 서비스를 제공 받을 때, 어떤 점에서 만족감을 느끼는 지에 대한 서비스 타입 선호도와, 그 사용자의 특성 정보이다. 가령, 사용자가 다혈질에 피곤하다면, 그 사용자의 요청을 먼저 처리해주는 것이 다른 가족들로 하여금 스트레스를 덜 받게 해주는 방법이 될 것이다. 또한, 사용자의 일정이나, 사용자의 심리 상태도 그가 요청한 서비스의 중요도를 변화시킬 수 있는 요소가 될 수 있다. 서비스 속성은 일반적인 서비스 타입을 의미한다. 마치, 소변이 마려운 것은 이를 닦는 것보다 중요한 서비스일 것이고, 감기 걸린 사람의 실내 온도를 높이려는 요구는 담배를 피기 위해 창문을 열려는 욕망보다 중요하다. 물론 그 반대로 생각할 수도 있고, 어느 서비스 타입이 어떤 중요도를 갖는가 하는 것은 전적으로 그 인텔리전트 홈을 사용하는 사용자들의 합의를 통해 결정되어야 할 것이다. 충돌 해결을 위한 규칙은 시간 논리를 기반으로 하고 있다. 가령, 텔레비전을 점유하고 있는 사용자가 20분 후에 그 점유 상태를 푼다면, <그림 3>의 충돌 해결 방안 중, 대체 자원을

사용하는 것의 효과와 자원 접근 권한을 기다리는 것의 효과를 보다 섬세하게 비교 할 수 있게 된다. 또한 이는 서비스 속성정보의 중요성도 보다 세밀하게 적용될 수 있다. 그러나, 이 역시 사용자들 개개인의 성격에 따라 규칙이 변화할 수 있기에, 본 제안 모델에서는 일반적인 충돌 해결 규칙으로만 제안한다. 실제 규칙의 생성은 시뮬레이션 과정에서 살펴 보기로 한다. 이 모든 정보들은 각 에이전트들 사이의 정확한 정보 공유를 위해 온톨로지를 사용하여 표현되고, 센서 네트워크로부터 받은 정보를 컨텍스트 정보로 변환하는 과정 역시 온톨로지를 사용하여 이루어진다. 이상의 정보를 바탕으로 충돌이 발생하는 경우를 찾아내어 자원을 분배하는 방법은 다음의 과정을 따른다. 우선, 충돌 해결 에이전트(Conflict Agent)는 각 요청에 따라 충돌이 발생하는 과정을 찾아낸다. 충돌 발생을 찾아내는 방법은 다음의 과정을 따른다.

1. 서비스 요청 큐에 들어오는 서비스를 처리하기 위한 자원을 시간대별로 예약한다.
2. 필요한 자원이 필요한 시간에 다른 서비스 요청에 의해 예약되어 있지 않다면, 서비스를 실행한다. 단 각각의 서비스들은 서로 독립적으로 자원 접근을 요청하는 것으로 처리한다.
3. 다른 서비스 요청에 의해 예약되어 있는 자원을 접근하려 한다면 충돌 발생

위의 과정에서 충돌이 발생할 경우, 해결 방안은 <그림 3>의 방법을 따른다. 즉, 기다리거나, 다른 대체 자원을 찾거나 자원에 대한 즉각적인 회수하는 방법이다. 이 때, 어느 상황에서 어떤 방법을 택할 것인가를 선택하기 위해 사용하는 규칙은 다음과 같다.

1. 현재 요청이 위급한 것이라면, 즉각적으로 자원을 회수한다.
2. 위급한 것이 아니라면 대체 자원을 찾는다.
3. 대체 자원의 사용이 사용자의 프로파일에 기반하여 특정 만족도 이상을 보장할 수 없다면 원래 필요로 한 자원의 접근이 허가될 때까지 기다린다.
4. 자원 접근 권한을 얻기까지의 대기 시간에 대한 사용자의 불 만족도가 프로파일에 기반하여 특정 불 만족도 이상이라면 대체 자원을 선택한다.

이 때, 어느 요청이 위급한 것인지, 어느 정도의 기다림이 혹은 대체 자원의 효과가 사용자를 만족시키는 지 혹은 불 만족 시키는지를 판단하기 위하여 미리 정의된 시간 논리에 의한 규칙을 사용한다. 이 때, 사용되는 만족도 계산 방법은 다음과 같다.

$$\text{만족도} = \frac{\text{사용자의 우호도} \times \text{서비스의 중요도}}{\text{서비스 응답 시간} \times \text{원 요청 서비스와의 차이}}$$

사용자의 우호도는 사용자의 프로필 정보에 따라, 사용자의 현재 기분 상태, 매너 좋음 등의 값이 되고, 서비스의 중요도는 서비스 속성의 값이며, 서비스 응답 시간은 얼마나 빨리 서비스를 받았는지, 그리고 원 요청 서비스와의 차이는 사용자가 기대했던 서비스와의 차이를 의미한다.

Simulation

본 논문에서 제안된 시스템을 평가하기 위하여 다음의 테스트 환경을 가정했다. 자원 충돌 시나리오는 <표2>에서와 같이 4개의 시나리오로 구성되어 있다. 우선 각 시나리오를 테스트 하기 위해 시나리오를 표현하는데 사용된 어휘들의 온톨로지를 작성하였다. 작성된 온톨로지를 기반으로 센서들이 수집한 환경 정보를 바탕으로 컨텍스트 정보가 생성되었다. 사용자는 시나리오를 따라 행동하고, 이에 따른 에이전트는 사용자의 의도에 따른 서비스를 선택하고 홈 서버 네트워크에 서비스를 요청한다. 이렇게 요청된 서비스를 제공하기 위해 홈 서버 네트워크에서는 미리 정의된 규칙에 의해 자원 충돌 상황을 제어하고 원하는 서비스를 각 사용자에게 제공할 수 있는가를 테스트 하였다. 각각의 테스트 과정은 <그림 4>에서 보여지는 것과 같고, 각 과정에 대한 세부 설명은 다음과 같다.

Scenario

<표2>에 나타난 4가지 충돌 상황을 나타낼 수 있는 각각의 시나리오는 다음과 같다.

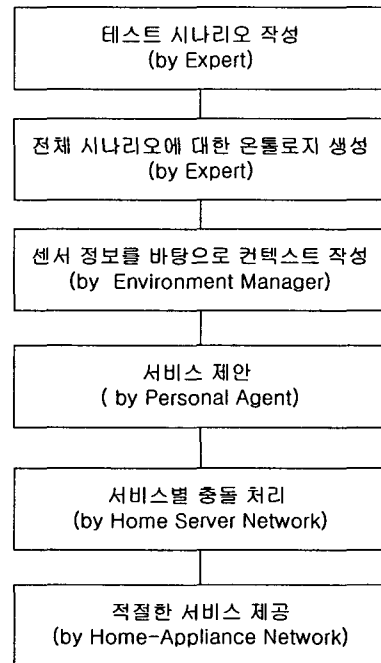
1. 한 명의 요구를 충족시키기 위한 자원을 다른 사용자가 먼저 사용하고 있는 경우

철수와 영희에게는 고등학교 3학년인 민수라는 아들이 있다. 영희는 드라마를 보기 위해 드라마 시간에 맞춰 집에 돌아왔다. 집에 왔더니, 민수가 거실에서 교육방송을 보며 공부를 하고 있었다. 녹화를 하기 위해 VCR을 보니, 이미 철수가 다른 채널에서 상영중인 영화를 녹화하고 있었다. 영희의 에이전트는 영희의 요구를 수행하기 위해 어떠한 결정을 내려야 할까?

2. 한 사용자의 요구를 충족시키기 위해 다른 사용자의 요구를 무시해야 하는 경우

철수네 집에는 화장실이 2개 있다. 하나는 거실에 있고, 다른 하나는 안방에 붙어 있는 것으로 안방을 통해서만 들어갈 수 있다. 어느날 저녁, 철수와 영희는 부부싸움을 했다. 부부싸움 끝에 영희는 울면서 안방으로 들어가 문을 잠가버렸다. 그때, 철수는 갑자기 배가 아프기 시작했다.

거실의 화장실은 이미 민수가 차지하고 있었다. 민수는 입시 준비로 받는 스트레스때문인지 변비로 고통받고 있었고, 한번 화장실에 들어가면, 쉽사리 나오지 않았다. 철수는 화장실에 가고 싶었지만, 안방의 화장실은 영희가 안방 문을 열어줘야만 들어갈 수 있고, 거실의 화장실은 이미 민수가 사용하고 있었다. 철수의 에이전트는 어떻게 철수가 안방의 화장실로 들어갈 수 있는 길을 찾아낼 수 있을까?



<그림 4> 테스트 시퀀스

3. 한 명의 요구를 충족시키기 위한 환경의 변화가 다른 사람의 요구를 침해하는 경우

철수와 영희는 부부 사이이다. 이 부부는 맞벌이를 하고 있는데, 내일은 철수의 중요한 미팅이 있는 날이다. 철수는 미팅 준비를 위해 회사에 8시까지 출근해야 한다. 영희는 최근 회사의 일이 밀려서 3일째 야근을 하다가 오늘에서야 비로서 일이 끝나 지친 몸을 이끌고 집에 돌아왔다. 내일은 휴가를 내서, 늦게까지 잘 수 있다. 철수는 잠들기 전, 자신의 에이전트에게 6시에 깨워줄 것을 요청했다. 영희는 자신의 에이전트에게 절대 깨우지 말 것을 요청했다. 다음날 6시 철수의 에이전트는 알람을 울려야 하고, 영희의 에이전트는 알람을 울리지 않게 하여야 한다. 이튿날 알람은 과연 울려야 할까?

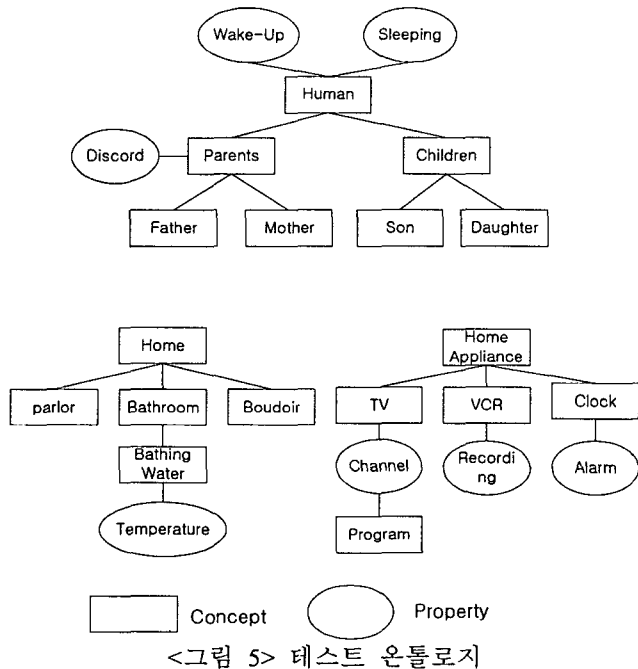
4. 동일한 요구가 서로 다른 사용자에게서 들어온 경우

영희는 집에 돌아오기 전 자신의 에이전트를 통해 목욕물을 미리 받아놓을 것을 부탁했다.

그러나, 영희가 목욕물을 미리 받을 것을 부탁하는 시간, 철수가 이미 목욕을 하고 있는 중이었다. 영희의 에이전트는 이미 목욕물이 받아져 있는 것을 확인하고 영희의 작업 요청을 완료시켰지만, 막상 영희가 집에 돌아왔을 때, 철수는 목욕을 끝내고, 욕조를 비운 뒤였다.

Ontology

본 시나리오를 테스트 하기 위한 온톨로지는 <그림 5>에서 보여지는 것과 같이 사용자, 집 구조, 가전 기기의 세 부분으로 이루어져 있다. 컨셉들은 보다 많은 속성들을 갖지만, 본 테스트 시나리오에서 사용되지 않는 속성들은 표현되어 있지 않다..



Context Generator

테스트 시나리오를 위한 컨텍스트 정보는 한정되어 있기 때문에 실제 센서 네트워크를 통한 환경 지각 정보를 컨텍스트 정보로 직접 변환시킬 필요가 없다. 그러나, 일반적으로 컨텍스트 정보를 만들기 위해서는 컨텍스트 정보를 생성하기 위해 필요한 정보의 종류를 온톨로지를 사용하여 규정짓고, 그에 맞는 센서 정보를 통합하는 방법을 따를 수 있다. 본 테스트에서 사용하는 컨텍스트 정보는 첫 번째 시나리오의 경우, "민수가 거실에서 교육 방송을 보고 있다", "철수는 VCR에 영화를 녹화시키고 있다", "영희는 드라마를 보고 싶어한다" 의 세가지로 표현된다..

Service Selection

각 퍼스널 에이전트는 센서들의 정보를 바탕으로 생성한 컨텍스트 정보와 사용자의 프로파일을

통합적으로 분석해 사용자의 요구를 예측한 뒤 그에 부합하는 서비스를 선택해 제공 할 수 있다. 또한 사용자의 직접적인 요청에 의해 특정 서비스가 선택될 수 있다. 본 테스트에서는 시나리오에 따른 서비스 요청이 정해져 있으므로, 퍼스널 에이전트의 서비스 선택 과정은 생략되었다. 그러나 서비스를 요청할 때, 추가되는 사용자의 프로파일 및 선호도 정보는 시나리오 대로의 서비스 요구와 함께 전달되었다.

Conflict solving Rules

자원 충돌을 해결하기 위해 필요한 정보의 종류와 규칙들은 충돌이 발생하는 상황에 따라 각기 다르다. 첫 번째 시나리오에서 발생하는 충돌을 회피하기 위한 정보와 규칙은 <표4>에 나타나 있다. 정보 내용은 각 정보에 따른 세부 속성에 따라 구별 짓지 않았다.

<표4> 첫 번째 시나리오의 충돌 해결 정보와 규칙

정보 종류	정보 내용	
사용자 프로파일	철수:	다혈질에 과파함
	영희	사려 깊고 다정 다감함
	민수	이기적이고 독선적임
서비스 속성	철수	VCR을 통한 영화 녹화 중요도 : 보통 VCR 자원 선점
	영희	TV를 통한 드라마 시청 중요도 : 보통
	민수	교육방송 시청 중요도 : 높음 TV 자원 선점
충돌 해결을 위한 규칙	1. 자원은 선점자에게 우선순위가 있음 2. 반사회적인 사람은 쉽게 불만에 빠짐 3. 중요한 서비스는 먼저 처리되어야 함 4. 방송 시청은 큰 화면을 선호함 5. 방송은 처음부터 끝까지 봐야 함	

Result

각 시나리오에 대한 결과는 다음과 같다. 첫 번째 시나리오의 경우, 영희는 결국 재방송을 기다려야 했다. 두 번째 시나리오의 경우, 민수는 안방의 화장실을 사용할 수 있었다. 세 번째 시나리오의 경우 알람이 울렸고, 마지막 시나리오의 경우 영희는 철수가 목욕 한 것과 상관 없이 목욕을 할 수 있게 되었다. 각각의 경우, 그러한 결과가 나온 까닭은 다음과 같다. 첫 번째 시나리오의 경우, 철수의 개인 프로파일의 속성을 다혈질에 성격이 급한 것으로 설정했기 때문이고, 두 번째 시나리오의 경우, 철수의 상황이 너무 급했기 때문에 즉각적인 자원 환수 과정이 발생했다. 세 번째 경우, 철수의 미팅이 더 중요했다고 판단했기 때문에 알람이 울렸고, 마지막 시나리오의 경우 철수의 요청과 영희의

요청이 각기 다른 자원 접근을 시도했기 때문이었다. 이 같은 결과는 자원을 분배하기 위한 판단 기준의 속성 값의 변화에 따라 다른 결과를 가져올 수 있다.

Conclusion

유비쿼터스 컴퓨팅 기반의 인텔리전트 홈 네트워크는 영화에서나 볼 수 있었던, 신 개념의 가정 환경을 제공해 줄 준다. 그러나, 이 유비쿼터스 컴퓨팅 환경이 제어하는 가정 환경은 기존의 운영체제가 로컬 컴퓨터 자원을 사용하여 각 프로세서에 할당하고, 사용자와 실시간으로 응답하는 기술 이상의 것을 요구하고 있다. 서로 다른 사용자들의 요청에 응답하기 위하여 한정된 가정 환경의 자원을 분해하여 제공하기 위해서는 효율적인 자원 접근 권한 제어 방법이 필요하다. 이에 본 논문에서는 센서들이 지각한 컨텍스트 정보를 바탕으로 시간 논리를 사용한 규칙 기반 시스템을 사용하여 자원 접근 권한을 조절하는 방법을 제안했다.

제안된 방법을 통해 모든 자원들은 각 사용자들의 서비스 요청에 따라 적절히 분배가 이루어졌지만, 그 분배의 결과는 각각의 상황에 따라 달라진다. 다만, 서비스 처리 결과는 모든 사용자의 만족도를 최대한 높이기 위한 방향으로 진행되었다는 점을 살펴볼 수 있었다. 인텔리전트 홈 네트워크의 태생적인 자원 분배의 한계는 모든 서비스 요청의 응답이 동시에 실현될 수 없음을 보여주고 있다. 다만, 본 논문에서 제안하는 방법을 통해, 그 상황에서 가장 적절한 자원 분배 방식을 통해 모든 사용자의 만족도를 높이는 것을 목표로 각 에이전트들은 자원 접근 권한을 놓고 협상을 벌였다. 이 방법은 각 사용자에게 최대의 만족을 제공할 수는 없지만, 모든 사용자들의 만족도의 합을 최대로 높이는 것을 목표로 하고 있다.

Acknowledgments

본 논문은 유비쿼터스 컴퓨팅 및 네트워크 원천기반 기술 개발사업내 uT-DB 기반의 지능성장 학습 엔진 사업의 일환으로 지원 받아 수행되었음

(과제번호 M103KT010007-04K2001-00713)

References

- [1] Mark Weiser *The Computer for the 23st Century*, Scientific American Ubicomp Paper 1991-09.
- [2] Abowd, G.D. *Software design issues for ubiquitous computing* VLSI '98. System Level Design. Proceedings. IEEE Computer Society Workshop on , 16-17 April 1998 Pages:104 - 109
- [3] Scholtz, J., Consolvo, S. *Toward a framework for evaluating ubiquitous computing applications* Pervasive Computing, IEEE , Volume: 3 , Issue: 2 , April-June

2004 Pages:82 - 88

- [4] Caldwell, D.G., Linge, N. *A new perspective on the intelligent home* Gray, J.; Impact of Multimedia Services on the Home Environment, IEE Colloquium on , 12 Jan 1996 Pages:7/1 - 7/4
- [5] Jae Chul Moon, Soon Ju Kang *A multi-agent architecture for intelligent home network service using tuple space model* Consumer Electronics, IEEE Transactions on , Volume: 46 , Issue: 3 , Aug. 2000 Pages:791 - 794
- [6] D.C.; Cameron, J.M., Arkin, R.C MacKenzie, *Specification and execution of multiagent missions*. Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on , Volume: 3 , 5-9 Aug. 1995 Pages:51 - 58 vol.3
- [7] Xin Wang, Schulzrinne, H. *An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications* Selected Areas in Communications, IEEE Journal on , Volume: 18 , Issue: 12 , Dec. 2000 Pages:2514 - 2529
- [8] Ligeza, A. *Towards logical analysis of tabular rule-based systems* Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on , 26-28 Aug. 1998 Pages:30 - 35
- [9] James F. Allen *Maintaining Knowledge about Temporal Intervals* Communications of the ACM 26(11), 1983, 832-843