

유비쿼터스 환경을 위한 모바일 멀티 에이전트에 관한 연구

김 만선
공주대학교
mansun@kongju.ac.kr

A Study on Intelligent Multi Agents for a Ubiquitous Environment

Mansun Kim
Kongju National University

요 약

최근에 PC기반의 온라인 게임과 유무선 연동이 가능한 유비쿼터스 게임기의 개발이 활발하게 이뤄지고 있다. 본 논문은 단순히 연동이 가능한 환경에서 더 나아가 지능적이고 효율적인 게임 서비스를 제공하기 위하여 지능형 멀티 에이전트 시스템과 효율적인 자원 관리 시스템을 제안한다. 유비쿼터스 컴퓨팅 환경에서 지능형 에이전트는 사용자의 주변 환경을 인식하고 사용자의 목적에 적합한 행위를 자율적으로 선택하여 제공할 수 있어야 한다. 또한 메모리 용량이 협소한 자원 관리 시스템을 보완하여 내부의 메모리 공간을 최적의 상태로 유지할 수 있어야 한다.

키워드 : 지능형 에이전트, 모바일 에이전트, 유비쿼터스 게임기

1. 서 론

유비쿼터스 컴퓨팅(Ubiquitous Computing)[1][2][3]이 지향하는 “Everyday Computing”은 사용자가 이동을 하여 장소가 변하거나, 휴대하거나 사용하는 장비가 바뀌거나, 시간이 지나더라도, 지속적으로 “연결되어” 원하는 작업을 수행할 수 있게 해주는 연속적인 컴퓨팅 환경의 구현을 목표로 한다. 최근 PC 기반의 온라인 게임과 유무선 연동이 가능한 유비쿼터스 게임기의 개발이 활발하게 이뤄지고 있다. 이런 환경에서 기능을 수행하려면 다음과 같은 특성을 가져야 할 것이다.

(a) 지능형 인터페이스 기술 : 사용자들이 휴대하는 단말기는 정보 처리 및 계산 능력을 가지면서, 통신 능력도 갖추어야 한다. 그리고 사용자들이 사용하는데 쉽고 편해야 하므로 사용자 인터페이스 소프트웨어에 적응성 혹은 적용할 수 있는 능력을 포함시킬 수 있어야 한다.

(b) 컨텍스트 인식(Context Awareness) 기능 : 현재 사용자가 처한 상황에 맞는 정보와 서비스를 제공하려면 사용자의 ID, 위치, 상태, 동작 등을 알아낼 수 있어야 한다. 이를 위해 GPS, Augmented Reality(증강 현실), RFID 등 다양한 기술이 사용될 수 있다.

(c) 자동화된 자료 수집 및 검색 기능 : 사용자의 모든 경험을 기록하고 저장한 다음, 쉽고 빠르게 그 자료들을 검색할 수 있도록 해주어야 한다. 대용량의 자료 처리, 직관적이고 신속한 정보 검색을 할 수 있어야 한다.

이런 지능적이고 효율적인 서비스 제공을 위해서 사용자 주변의 장치와 정보를 공유시킴으로써 서비스 지원 매체의 다양화 작업도 필요하다. 본 논문은 이런 과정을 통해 게임을 즐길 뿐만 아니라 사용자의 요구(needs)를 파악하고, 흥미를 더욱 증가시킬 수 있는 모바일 환경을 지원하는 지능형 에이전트를 제안한다.

2. 관련 연구

2.1 지능형 모바일 에이전트

에이전트(Agent)는 지능을 가진 인간처럼 의사 결정을 하고, 학습하고, 스스로 행동하며, 주어진 임무를 자동적으로 처리해주는 시스템이다. 에이전트는 여러 가지 관점에서 정의될 수 있으나, 일반적으로 자율성(autonomy), 목표지향성(goal directness), 반응성(reactivity), 사회성(social ability), 이동성(mobility), 이성(rationality), 적응성(adaptability) 등의 특성을 가지면서 사용자 또는 다른 에이전트와 상호 작용하는 소프트웨어라고 정의할 수 있다.

에이전트는 지시된 작업을 수행하고 조정하기 위한 기능을 제공하는 제어지식 에이전트와 수행할 역할의 기능을 규정하고 있는 영역지식 에이전트, 그리고 프로시저 콜과 메시지 교환을 처리하는 통신 모듈 에이전트로 구성된다.

2.2 컨텍스트 인식 기능

컨텍스트 인식은 사용자의 환경으로부터 상황 정보를 인식하고 통합하는 기술을 말한다. 그리고 컨텍스트 인식을 위한 정보 구조는 응용 분야마다 다양하며 사용자ID, 위치, 상태, 동작, 온도, 심리적 요소 등을 주로 이용하고 있다[4][5].

사용자와 주변 환경에 대한 정보를 제공하는 모든 것을 센서로써 이용한다. 사용자 및 주변 환경의 정보를 제공하는 센서가 하드웨어로 또는 소프트웨어로 구현되는지에 따라 물리적(physical) 센서와 논리적(logical) 센서로 구분된다. 또한 환경 변화를 감지하여 발생하는 신호의 처리 방법에 따라 능동적(active) 센서와 수동적(passive) 센서로 구분된다[6].

2.3 자원 관리 시스템(RMS)

모바일 디바이스 내부의 저장 공간에서 외부의 서버까지 메모리 영역을 확장하여 데이터 활용을 가능하게 하고 내부의 자원관리 기능을 수행할 수 있는 에이전트이다.

3. 제안하는 시스템

3.1 자원 관리 시스템(RMS : Resource Management System)

RMS는 데이터 저장 공간을 확대하여 자원 관리를 효율적으로 운영하기 위한 시스템이다. 실행 파일이 필요한 리소스 파일을 호출할 경우 RMS는 서버에 요청하여 리소스 파일을 다운로드 받아 동적으로 로딩한다. 기존의 다운로드 시스템은 콘텐츠 일을 다운로드 받아 수행하는 반면 RMS는 확장된 프로세스를 수행하는 것에 차이가 있다.

RMS를 적용한 디바이스는 모바일 공간 확장으로 확대된 프로세스 적용이 가능하며 내부 파일을 효과적으로 관리하여 최적의 메모리 공간을 유지하는 효과가 있다.

표 3.1은 RMS API와 기능을 보여준다. 그림 3.1은 자원을 공유하는 RMS 서버와 RMS Agent간의 수행구조를 보여준다.

표 3.1 RMS API set

RMS API set	기능
RMS_LoadResource()	실행파일이 수행하기 위한 리소스 파일 호출함수
RMS_SearchLocation()	리소스의 위치를 검색하여 위치 정보 제공
RMS_Link_App()	스토리지에 있는 리소스 파일이 기존의 어플리케이션과 링크됨
RMS_Get_Resource()	RMS_SearchLocation의 위치 정보에 따른 리소스 핸들 반환
RMS_Process_Type()	리소스의 반환 및 관련 어플리케이션 정보 제공
RMS_VM_Interface()	리소스 호출 후 디바이스에 내장되어 있는 VM과의 호환성 유지

3.2 지능형 멀티 에이전트 (IMA : Intelligent Multi Agent)

사용자는 센서를 통해서 5WIH(Who, Why, What, Where, When,

How)에 해당되는 사용자 정보를 사용자인터페이스 에이전트에 전달하게 된다. 사용자 인터페이스에 대한 에이전트는 크게 다음과 같은 3가지 역할을 수행한다고 볼 수 있다. 첫째, 사용하기 쉬운 인터페이스를 제공하는데 사용자 개개인의 패턴(pattern)을 반영하고 매크로(macro) 등을 사용하여 반복적인 작업에 대한 편리성을 제공한다. 둘째, 학습(learning)을 통해 사용자의 습성을 파악하여 적용시킨다. 셋째, 멀티미디어 인터페이스를 제공하는 역할을 수행한다. 그림 3.2는 IMA의 전반적인 구조를 보여준다.

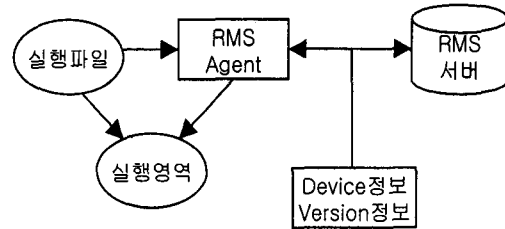


그림3.1 RMS 서버와 RMS Agent

DB는 자원을 공유하는 기술을 이용하게 되는데 그리드 컴퓨팅, Peer to Peer(P2P)로 가능하다. 그리드 컴퓨팅 기술은 고성능 대형 컴퓨터를 중심으로 발전하고 있으며, P2P는 이동성이 요구되는 모바일 기반 중심으로 발전하고 있다. P2P 프로토콜은 에이전트의 상태를 기억하고 일단 접속이 끊어지더라도 계속해서 이전의 질의에 대한 유용한 정보나 상태의 변화를 제공할 수 있어야 한다.

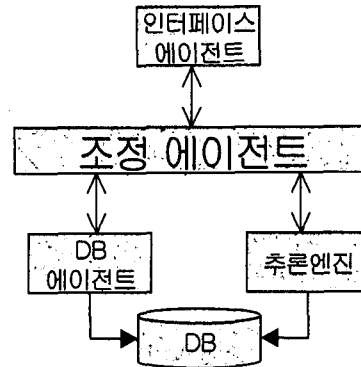


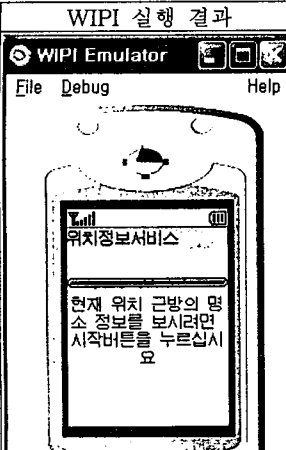
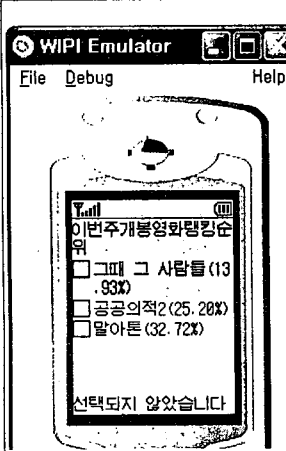
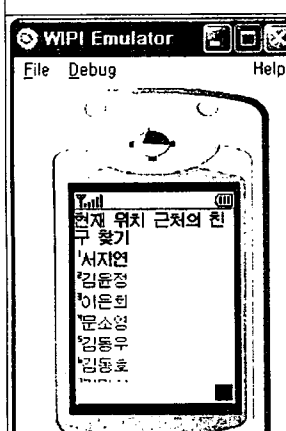
그림 3.2 IMA시스템 구조

조정 Agent는 예측하지 못한 환경 변화에 반응하여 조정할 수 있는 역할을 한다.

추론 엔진(inference engine)은 지식 베이스를 이용하여 문제를 해결하기 위해 논리적으로 지식을 제어하며, 새로운 지식을 추론하기 위해 규칙들을 어떻게 적용해야 할 것인가를 결정하는 규칙해석기와 규칙들이 적용되는 순서를 결정하는 스케줄러로 구성된다.

4. 실험 및 결과

표 4.1 WIPI 실행 결과와 소스

WIPI 실행 결과	자바 소스
	<pre>Private void load(){ sh = new ShellComponent(); sh.setTitle("위치정보서비 스"); FormComponent frm = new FormComponent(); if(bINTERACTIVE) label1 = new LabelComponent(" 현재 위치 근방의 명소 정보를 보 시려면 시작버튼을 누르십시오");</pre>
	<pre>ShellComponent sh = new ShellComponent(); sh.setTitle("이번주개봉영 화랭킹순위"); Image img_enter = null try{ img_enter = Image.createImage("girl.png"); } catch(Exception e){ m_cbc1 = new CheckboxComponent("말아톤 (32.72%)", img_enter); m_cbc2 = new CheckboxComponent("공공의적 2(25.20%)", img_enter); m_cbc3 = new CheckboxComponent("그때 그 사 람들(13.93%)", img_enter);</pre>
	<pre>listcmp.setPacked(true); listcmp.append("서지연 ", img_enter); listcmp.append("김윤정 ", img_girl); listcmp.append("이은희 ", img_sball); listcmp.append("문소영 ", img_sgirl); listcmp.append("김동우 ", img_enter); listcmp.append("김동호 ", img_girl); listcmp.append("김만선 ", img_sball); listcmp.append("김진희 ", img_sgirl);</pre>

4.1 실험 환경

모바일 서버를 위해서 C++, 모바일 에뮬레이터인 WIPI(Wireless Internet Platform for Interoperability)를 사용하였으며 자바를 이용해 공유 네트워크를 구현하였다. 모바

일 서버의 사양은 셀러론 2.8Ghz CPU, 메모리 248 MB, 운영체제는 윈도우 XP이다.

4.2 서비스 실행 결과

'위치 정보 서비스' 는 게임을 즐기는 사용자가 근처의 유명한 식당이나, 명소 등을 찾을 수 있도록 서비스를 제공하기 위한 초기 화면이다. 시작 버튼을 누르게 되면 유비 센서에 의해서 위치 정보를 입력받고 멀티 에이전트에 의해 근방의 명소 정보를 보여주게 된다.

'이번주 개봉영화 랭킹 순위 서비스' 는 영화관에서 제공된 개봉 영화의 목록과 랭킹 순위를 제공하는 서비스로서 티켓 구매 현황(단위:%)등이 포함된 정보를 제공하게 된다. 표 4.1은 WIPI 실행 결과와 자바 소스의 일부를 보여준다.

'현재 위치 근처의 친구 찾기 서비스' 는 <http://www.cyworld.com>에서 제공되는 '일촌 ON' 기능과 유사한 개념으로서 P2P 공유를 이용해서 근처의 위치에서 활동하는 미리 등록된 친구의 컨택스트 정보를 받아서 친구의 목록을 보여주게 된다.

5. 결론 및 향후 연구과제

최근 유비쿼터스 게임기의 개발이 활발하게 이뤄지고 있는 시점에서 본 논문은 게임을 즐기는 사용자의 흥미를 더욱 향상시킬 수 있는 모바일 환경을 제안하고자 지능형 멀티 에이전트 시스템과 자원관리시스템을 제안한다.

유비쿼터스 컴퓨팅 환경에 맞도록 적용된 지능형 멀티 에이전트 시스템은 사용자의 컨택스트 정보를 인식하고 지능적이고, 다양한 서비스를 제공할 수 있다. 기존의 모바일 환경은 이동의 편리성을 가지고 있는 대신에 메모리 용량이 협소하여 효과적인 자원 관리를 수행하지 못하는 단점이 있었다. 본 논문에서 효율적인 자원 관리 시스템(RMS)을 적용하여 내부의 메모리 공간을 최적의 상태로 유지할 수 있다.

향후 연구 계획은 지능형 모바일 에이전트의 기능을 보강하여 서버에서 호출시 로드 밸런싱을 효율적으로 관리할 수 있는 방법에 관하여 연구할 계획이다.

참고 문헌

[1] Weiser, M., <http://www.ubiq.com/weiser>.
 [2] Weiser, M., "The computer for the 21st Century", Scientific American, Vol. 265, No.3, pp.94-104, September, 1991.
 [3] Weiser, M., "Some computer science issues in ubiquitous computing", Communications of ACM, Vol.36, No.7, pp.75-84, July, 1993.
 [4] J. R. Quinlan, C4.5 Programs for Machine Learning, San Mateo, CA, Morgan, Kaufaman.
 [5] Foundation for intelligent physical agent. <http://www.fipa.org>
 [6] B. V. Dasarathy, "Sensor fusion potential exploitation: Innovative architectures and illustrative approaches", In Proceeding of IEEE, Vol. 85, pp.24-38, Jan. 1997.