

# 멀티 에이전트 기반 헬스케어 상황정보 서비스 플랫폼의 설계<sup>☆</sup>

## Design of platform supporting for healthcare context information service based on multi-agent

박 무 현\*                      정 창 원\*\*                      주 수 중\*\*\*  
Moohyun, Park              Chang-Won Jeong              Su-Chong Joo

### 요 약

본 논문에서는 홈 환경에서 멀티 에이전트 기반의 헬스케어 상황정보 서비스를 제공하기 위한 소프트웨어 구조의 설계에 대해 기술한다. 본 플랫폼에서의 분산객체그룹 프레임워크(Distributed Object Group Framework, DOGF)는 수행객체들 및 헬스케어 지원 센서 또는 기기들의 논리적인 서비스별 그룹화를 지원하고, 멀티 에이전트 프레임워크인 JADE(Java Agent DEvelopment framework)는 사용자의 이동성과 이질적인 환경에서의 서비스를 지원한다. 플랫폼 상의 멀티 에이전트는 건강 관리 및 유지와 관련된 여러 종류의 헬스케어 상황정보 서비스를 제공하기 위하여 각각의 환경에 대한 정보를 수집하는 에이전트와 모바일 기기의 특징에 따라 서로 다른 서비스를 하는 에이전트 그리고 이들을 관리하는 에이전트로 분류할 수 있다. 이러한 JADE의 에이전트와 분산객체그룹 프레임워크의 동작은 모바일 프락시에서 인터페이스를 하며 정보 교환을 제공하거나 이동 패턴을 지원한다. 본 논문에서는 이러한 동작을 통하여 분산객체그룹 프레임워크와 JADE가 헬스케어 상황정보 서비스를 제공하기 위해 멀티 에이전트에 기반을 둔 플랫폼과 서비스별 에이전트의 설계에 대해 기술하였고, 끝으로 헬스케어 상황 정보 서비스를 위한 물리적인 시스템 환경과 플랫폼 기반의 프로토타입을 보였다.

### Abstract

In this paper, we describes the design of software architecture supporting for healthcare context information service platform based on multi agent in home environment. In this platform, the DOGF supports the execute object and healthcare sensors and device's logical services grouping. JADE framework can support mobility in heterogeneous environment. The multi agents on platform order to support a healthcare context information service it will be able to divide. An agent collects an environment information from distributed devices. Another an agent follows mobile-device specific and it does a different service. And an agent where it manages like this. The mobile-proxy&agent is an interface part between DOGF and JADE, support data interchange or mobility pattern. For DOGF and JADE to provide healthcare context information service, we describes the design of multi agent software platform and multi agent classification by services. Finally we showed the system environments which is physical environments and prototype based on platform for healthcare context information services.

☞ keyword : Multi agent, Healthcare context information service, Distributed object group framework, JADE, 멀티 에이전트, 헬스케어 상황정보 서비스, 분산객체그룹 프레임워크

## 1. 서 론

\* 준 회 원 : 원광대학교 컴퓨터공학과 석사과정  
mhyuning@wonkwang.ac.kr

\*\* 정 회 원 : 원광대학교 전기전자 및 정보공학부 박사 후  
연구원 mediblu@wonkwang.ac.kr

\*\*\* 종신회원 : 원광대학교 전기전자 및 정보공학부 교수  
scjoo@wonkwang.ac.kr(교신저자)

[2007/08/29 투고 - 2007/09/18 심사 - 2008/01/22 심사완료]

☆ “이 논문은 2007년 정부(교육인적자원부)의 재원으로 한

최근 유비쿼터스 컴퓨팅을 기반으로 헬스케어에 대한 연구가 홈 네트워크 환경에서 개인 건강 유지와 관리를 위한 실용적인 헬스케어 서비스 연구로 활발하게 진행 중에 있다. 또한, 병원과 같은 기관에서도 원활한 비즈니스 업무를 위한

국학술진흥재단의 지원을 받아 수행된 연구임” (지방연구중심대학육성사업/헬스케어기술개발사업단)

통합 솔루션 개발 연구로 진행되고 있다. 본 논문에서는 가정 내에서 사용자들의 개인 건강 유지 및 관리를 위해 상황에 따르는 정보 서비스를 제공하는데 목적을 둔다. 이를 위해서 가정 내에 물리적인 컴퓨팅 환경 구성과 다양한 자원과 응용들이 제공하는 서비스에 필요한 데이터 처리와 데이터 특성에 따르는 입출력 장치의 매칭 그리고 데이터의 변환과 같은 기술이 요구된다. 특히, 이러한 환경은 사용자의 상황에 따라 다양한 분산 플랫폼의 유무선 네트워크 인프라를 기반으로 정보를 수집하거나 교환하며 정보 서비스를 제공한다. 그러나 기존의 클라이언트/서버 패러다임을 이용한 서비스 제공 방법은 구성요소간의 빈번한 상호작용과 실시간 처리시간의 지연과 위치에 따라 변화되는 상황을 지원하기 위한 방법으로 미흡하다[1].

따라서 이러한 문제점을 해결하기 위해 본 논문에서는 멀티 에이전트 기반 헬스케어 상황정보 서비스[12] 플랫폼에 중점을 두어 기술한다. 본 논문에서 제안하는 플랫폼은 본 연구실에서 개발한 DOGF(Distributed Object Group Framework)[2]와 Tilab에서 개발한 JADE(Java Agent Development framework)[3]를 기반으로 한다. DOGF는 개별적인 물리 장치 또는 각각의 응용들을 구성하는 분산 객체를 기능별 또는 영역별로 그룹화 하여 각각의 객체로부터 생성되는 정보들을 사용자의 목적에 맞게 서비스 할 수 있도록 지원하는 프레임워크이다. 또한, JADE는 특정 응용에 독립적인 환경을 제공하여 분산 응용 에이전트 개발을 지원하는 소프트웨어 플랫폼이다. 제안한 서비스 플랫폼 상의 멀티 에이전트는 건강의 관리 및 유지와 관련된 헬스케어 상황정보 서비스를 제공하기 위해 각각의 환경에 대한 정보를 수집하는 수집 에이전트와 모바일 기기의 특징에 따라 서로 다른 서비스를 하는 서비스 에이전트 그리고 이들에 대한 관리를 지원하는 마스터 에이전트 등으로 분류하였다. 이러한 에이전트들의 동작은 DOGF의 구성요소인 모바일 프락시 에이전트에

서 제공하는 인터페이스를 통해 정보 교환을 제공하거나 이동 패턴을 지원한다. 또한 상황정보는 4W1H 상황정보 모델[4]에 따라 헬스케어 상황정보를 제공한다. 4W1H 상황정보 모델로부터 분석된 상황정보는 <위치>, <시간>, <위치, 시간>에 따르는 이벤트 규칙에 따라 데이터의 형태 및 종류에 따라 사용자에게 가용한 출력 장치를 통해 서비스를 제공한다. 또한, 본 논문에서 제안한 플랫폼의 물리적인 시스템 환경과 각각의 모바일 장치들의 시스템을 체크하는 프로토타입의 시뮬레이터를 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 상황인식과 멀티 에이전트 플랫폼에 대해 기술하고, 연구 방법에 대해 설명한다. 3장에서는 본 논문에서 제안하는 헬스케어 상황정보 서비스의 플랫폼에 대해 기술한다. 4장에서는 멀티 에이전트지원 헬스케어 상황정보 서비스 과정 및 멀티 에이전트 구조에 대해서, 마지막 5장에서는 플랫폼 구축 현황과 향후 연구내용에 관하여 기술한다.

## 2. 관련연구

본 장에서는 상황인식과 멀티 에이전트 플랫폼에 관련된 연구에 대해서 살펴보고 이를 토대로 연구 범위와 내용에 대해서 기술한다.

### 2.1 상황인식

상황인식 시스템은 각종 센서 및 장치들로부터 많은 정보를 수집하고 특정 알고리즘을 통하여 주변의 상황을 인식하고 사용자에게 가장 필요한 정보를 판단하여 사용자에게 유용한 정보를 제공한다. 상황정보는 사용자가 이용할 수 있는 객체와 사람의 위치, 활동, 상태와 같은 정보들을 말한다. 이러한 상황정보의 수집 및 교환을 통하여, 특정 알고리즘을 통한 해석 및 추론의 처리 과정을 거쳐 인식된 상황을 사용자에게 적절한 서비

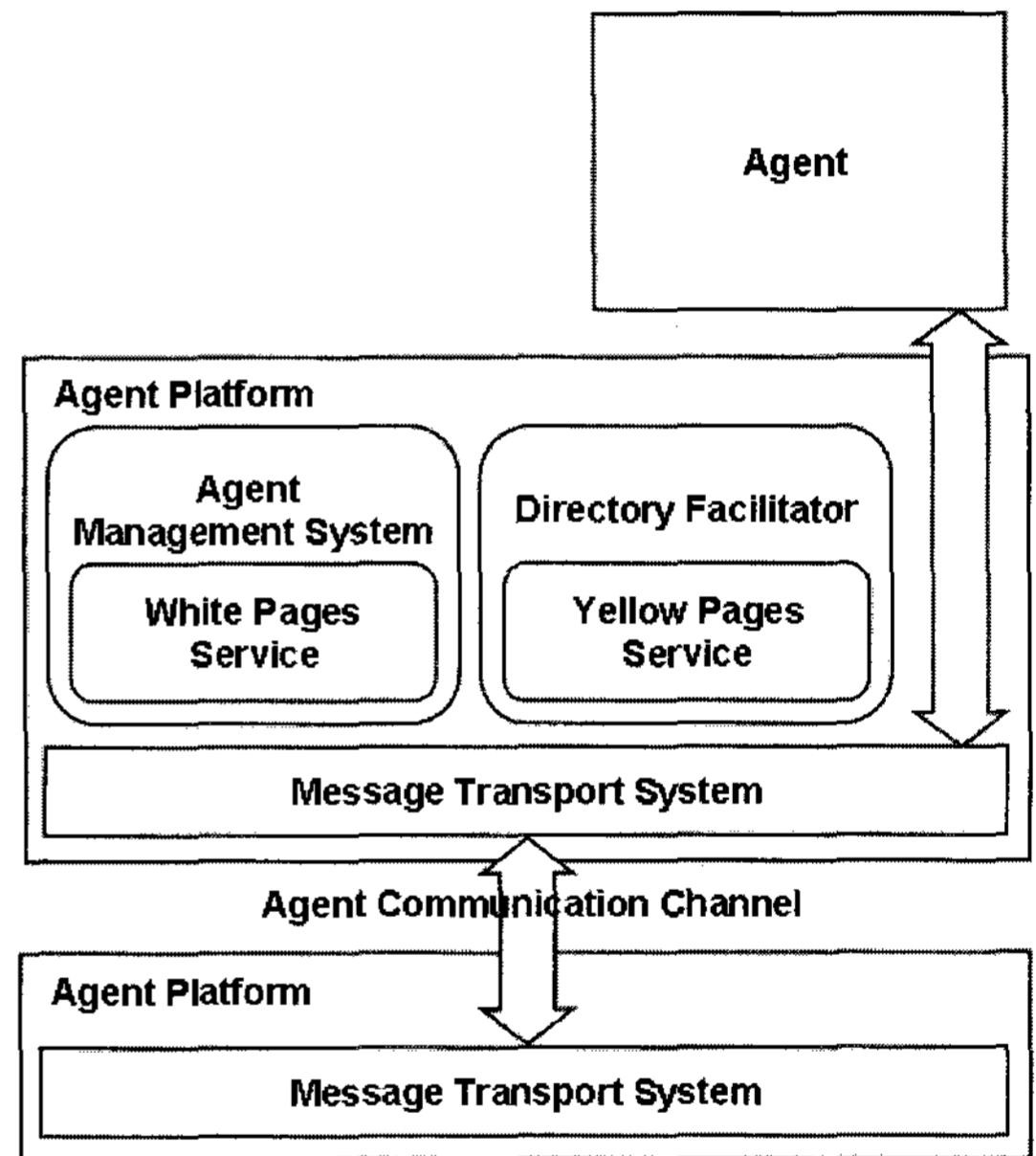
스의 형태로 제공하는 것을 상황인식 서비스라 한다. 이와 관련되어 상황인식과 상황인식에 관한 서비스로 대표적인 연구는 다음과 같다.

AT&T사에서 진행했던 연구[5]로, 홈 환경에서 사용자의 위치를 인식하여 사용자 주변의 환경에 적합한 상황인식 서비스를 제공한다. 홈 환경내의 사용자 위치를 인식하기 위하여 특정한 신호를 내보내는 배지를 각 사용자들에게 착용하게 하였다. 그리고 컴퓨터는 착용하고 있는 사용자의 배지 신호를 감지하여 위치를 인식하였다. 배지의 신호를 수신한 시스템은 신호의 세기를 계산하여 사용자의 신원 그리고 위치정보를 파악하여 상황에 따른 정보를 제공할 수 있게 하였다. MIT의 ComMotion 프로젝트[6]에서는 위치와 시간을 통해 사용자의 상황정보를 인식 하였다. 사용자에게 전달되는 리마인더(reminder) 정보는 위치정보에 따라 생성되고 사전에 의도된 사용자가 그 특정한 위치에 있을 경우, 그 메시지는 사용자가 가지고 있는 장치에 음성으로 듣거나 스크린 화면을 통해 메시지를 볼 수 있다. 미국 Georgia Tech 대학의 CyberMinder 연구[7]는 더욱 복잡한 상황정보를 인식하는 기술을 개발하였다. 실내의 정보뿐만 아니라 실외의 정보까지 포함하여 상황정보를 생성 했으며, 이러한 상황정보를 수집하여 요구된 기반정보를 바탕으로 상황인식과정을 거쳐 사용자에게 서비스를 제공할 수 있다. 본 논문에서는 이와 같이 상황인식 서비스를 제공하기 위하여 4WIH 상황정보 모델을 통한 상황인식 서비스를 지원한다. 즉, 누가(who), 어디서(when), 언제(when), 무엇을(what), 어떻게(how)라는 정보를 가지고 <위치>, <시간>, <위치, 시간>에 기반을 둔 리스트를 통하여 사용자의 상황에 가장 적합한 상황정보를 제공한다.

## 2.2 멀티 에이전트

멀티 에이전트는 고유한 동작을 하는 에이전트들이 상호작용을 하며 어떤 목적에 맞는 동작을

하는 에이전트 메커니즘의 하나이다. 본 논문에서는 플랫폼의 성격에 부합하기 위하여 이동성을 지원하는 에이전트들의 상호작용 과정을 통하여 서비스를 제공하게 된다. 본 플랫폼의 특정 에이전트들은 서로 다른 장치들을 이동하여 필요한 정보를 제공하거나 서비스 하며, 자신의 임무를 마친 에이전트는 다시 서버로 돌아와 새로 받은 데이터를 처리하거나 자신이 가진 데이터를 필요로 하는 다른 환경으로 이동하여 서비스를 제공한다[13].



(그림 1) FIPA Agent Platform

멀티 에이전트 플랫폼은 에이전트 정규화 기구인 FIPA(The Foundation for Intelligent Physical Agents)의 규정에 따라 기본적으로 필요한 생명주기 관리 서비스, 화이트 페이지 서비스, 옐로우 페이지 서비스, 메시지 전송 서비스를 제공한다. 다음의 그림 1은 FIPA에서 정규화 한 에이전트 플랫폼의 구조이다[8]. AMS(The Agent Management System)는 플랫폼에서 에이전트 사용을 위한 제어 역할을 한다. 네이밍 서비스와 에이전트에 대한 생명주기를 관리한다. 또한 화이트 페이지 서비스

를 제공하기도 한다. DF (Directory Facilitator)는 에이전트 플랫폼에 옐로우 페이지 서비스를 제공하는 역할을 한다. MTS(Message Transport System)는 메시지를 송·수신함에 있어 연속성을 유지시키며 ACL(Agent Communication Language) 메시지를 연결해 주는 역할을 한다. ACL 메시지는 비동기식으로 다른 에이전트와 통신하는데 사용된다. 또한, ACC(Agent Communication Channel)는 플랫폼의 내부와 외부의 기본적인 접속 경로를 제공한다. 이는 신뢰성 있는 정확한 메시지를 전달하는데 가장 기본적인 통신 방법이다. 또한 ACC는 서로 다른 에이전트 플랫폼 사이에서 IIOP나 HTTP를 사용하여 통신한다.

이러한 에이전트 플랫폼의 종류를 살펴보면, Odyssey와 Aglet, 그리고 JADE가 있다.

Odyssey 프로젝트[9]는 이전 General Magic사의 텔레스크립트(TeleScript) 구조를 채택하였다. 텔레스크립트는 C++언어로 구현된 에이전트 개발용 생성 환경이다. 이 제품은 자바가 나오기 전까지는 가장 많이 사용되어 왔던 에이전트 시스템으로 Agent, Worker, place의 세 가지 클래스가 있다. 하지만 참고할 수 있는 문서의 양이 적고, 시스템 구조의 폐쇄성 때문에 제한적 사용만 가능하였다. Worker는 Agent의 서브클래스로서 에이전트가 어떻게 동작하는지에 대한 정보를 제공한다. Place 클래스는 에이전트가 어디에 있으며 동작을 하는지에 대한 정보를 가지고 있다. Odyssey 에이전트는 간단한 메소드 호출을 이용하여 통신하고, 상위 레벨에서는 통신기능을 지원하지 않는다. Odyssey의 특이한 점은 다양한 다중 전송 메커니즘을 사용한다. 최근 RMI와 DCOM, IIOP를 지원하지만 안정적인 면에서 부족함을 보이고 있다.

IBM Japan에서 개발한 Aglet[10]은 자바의 애플릿 모델을 따르는 초창기 자바 기반 에이전트 개발도구이다. 이 모델은 미리 정의된 메소드를 가지고 있는 프레임워크를 제공하기 때문에, 프로그래머는 편의에 맞게 오버라이드하여 원하는 기능을 구현할 수 있다. 개념적으로 보면, Aglet은

Aglet이 동작하게 맞추어진 호스트 사이클 오가는 자바 객체이다. 즉, Aglet은 컴퓨터 네트워크상에서 스스로 다른 호스트로 이동하며 실행할 수 있는 모바일 자바 객체로 정의할 수 있다. 메시지로 응답을 받을 수 있고, Aglet 객체 모델은 context, proxy, message, itinerary, identifier와 같은 개념을 가지고 있다. 이러한 개념은 클래스를 가지고 이동할 수 있는 환경을 제공한다. Aglet은 에이전트끼리 간단만 Proxy객체를 통하여 메시지를 주고 받을 수 있다. 하지만, 그룹 객체간의 대화는 불가능하고 메시지의 전달방식에 프락시를 사용하는데 이는 상황에 따라 전송하는데 높은 대역폭이 요구됨에 따라 시스템 부하를 초래할 수도 있다.

ObjectSpace사의 Voyager[11]는 사실상 에이전트 분야에서 Java언어 프로그래밍을 확실하게 인식시킨 분산 애플리케이션 플랫폼이다. 사실상 Voyager가 등장하면서부터 Aglet와 Odyssey와 같은 멀티 에이전트 플랫폼 또한, Java언어로 만들어지게 되었다. Voyager 에이전트는 확장된 Java 메커니즘을 사용하는 Voyager ORB를 통해 설계되었다. ORB로서의 Voyager는 생산성, 호환성, 높은 수행 성능을 가지고 있다. 특히, Voyager에서 제공되는 데이터 통신 기능은 상당한 융통성을 가지고 있다. 메소드 호출에도 비동기식, 동기식, 후속 호출 식을 지원하고 있으며, 메신저라 불리는 경량 에이전트가 이 메소드 호출의 전달을 맡고 있다. 메소드 호출은 가상의 객체 프록시를 통해 원격 객체에 대한 호출과 로컬 메소드 호출이 동일한 문법을 사용한다. 그러나 Voyager는 다른 플랫폼과 상호작용하는데 동기화 문제점을 갖고 있다.

JADE는 FIPA의 멀티 에이전트 플랫폼의 규정에 따르는 에이전트 개발 소프트웨어 프레임워크이다. JADE는 다른 연구보다 비교적 늦게 시작했지만, 객체가 스스로 동작하고 이동성을 지원하는데 강력한 에이전트 분산응용 지원 플랫폼이다. 에이전트가 동작하는데 있어 다른 에이전트와의 상호작용이 가능하고 동작의 조절도 가능하다. 또

한, 플랫폼 자체가 개방적인 Open source로서 개발에 부담이 없다. 또한 모바일 환경을 지원하는 LEAP 플랫폼이 있어 다양한 환경을 지원할 수 있다. 다음 표 1은 앞서 언급한 에이전트 지원 프레임워크를 비교 분석한 표이다.

(표 1) 에이전트 지원 프레임워크의 특징

	Voyager	Odyssey	Aglet	JADE
Remote creation of agent	Yes	No	No	Yes
Messaging modes	Transparent	None	None	Transparent
Move to object	Yes	No	No	Yes
Move to program	Yes	Yes	Yes	Yes
Mobility	Java object serialization	Java RMI, CORBA IIOP	Aglet Transfer Protocol	CORBA IIOP, HTTP
Itineraries	Yes, no special API	Yes, special API	Yes, special API	Yes, special API
Connectivity	Full	Restricted	Restricted	Full
Security	Limited, sandbox model	Limited, sandbox model	Limited, sandbox model	Security manager
Pricing	Free commercial	Free noncommercial	Free noncommercial	Free noncommercial

### 2.3 연구 내용 및 방법

본 연구실에서는 분산객체그룹 프레임워크를 헬스케어에 적용하여 u-헬스케어 지원 분산 프레임워크를 개발하여 다양한 헬스케어 홈서비스에 적용하여 왔다. 특히, 홈 환경에서 사용자에게 스케줄 및 알림 서비스를 제공하는 상황 정보 시스템을 개발하였다. 기존 헬스케어 상황정보 시스템은 사용자가 홈 환경 내의 거실, 침실, 부엌과 같은 생활영역들 사이를 이동할 경우 끊임없이 연속적인 서비스를 제공하는데 목적을 두었다. 이를 위해 모바일 프락시를 이용하여 상황정보를 제공하였다. 모바일 프락시는 서버와 클라이언트 호스트를 연결하여 서버에서 제공하는 데이터를 클라이언트 호스트에게 전달하는 역할을 한다. 그러나 이동성[14] 지원에 있어 클라이언트/서버 메커니즘의 형태로 요청에 따라 서비스가 제공되며, 클라이언트 단말의 특성을 고려하지 않고 동일한 컴퓨팅 환경을 이용하였다. 또한, 데이터를 송·수

신 하는데 많은 대역폭을 사용하는 단점을 갖고 있다. 이러한 헬스케어 상황정보 시스템을 보완하기 위해서 개선되어야 할 몇 가지 요구 사항은 다음과 같다.

- 센서 및 각종 기기로부터 계속적으로 발생하는 데이터에 대한 처리과정이 부족하여 메모리 상에 불필요한 데이터를 남기게 된다. 이러한 데이터가 쌓이게 되면 사용자 컴퓨터 성능이 저하된다. 필요한 데이터만을 가지고 처리하고 불필요한 데이터는 스스로 삭제하여 데이터의 관리가 용이하게 해야 한다.

- 서버에서는 클라이언트의 상황을 고려하지 않고 방대한 양의 데이터를 한 번에 전송한다. 네트워크상에서는 모든 데이터를 동시에 수용해야 하기 때문에 대역폭을 많이 사용한다. 이러한 동작은 다른 프로세서에 영향을 미쳐 시스템에 급격한 과부하 현상을 가져올 수 있다. 이를 해결하기 위해서는 캡슐화 된 실행 객체와 데이터가 움직여 클라이언트 호스트에서 동작하게 한다. 캡슐화 된 객체는 적은 대역폭을 사용하기 때문에 다른 프로세서에 영향을 덜 미친다.

- 사용자의 현재 환경을 고려하여 서비스하기 위해서 다양한 모바일 기기와 고정기기에서 동작되는 시스템이 구성되어야 한다. 특히, PDA와 같은 모바일 장치들에서는 동작이 불가능하다. 다양한 시스템에서 다양한 환경에서의 통합화 된 플랫폼이 요구되며 또 다른 장치를 추가하거나 제거하여도 현재 존재하는 장치들만을 가지고 동일한 서비스가 가능해야 한다.

- 현재 데이터베이스에 저장되어 있는 정보만을 가지고 상황을 추론한다. 즉, 정형화된 규칙에 의해서만 서비스의 제공이 가능하다. 지능화된 시스템을 구축하기 위해서는 사용자의 행동에 대한 패턴과 현재까지의 정보를 적절하게 조합하여 상황을 추론해야 한다. 사용자의 응답형태와 움직이는 행동에 따라 패턴을 갱신하여 서비스가 가능해야 한다.



이와 같은 요구사항을 만족 시키고 개선된 상황정보를 서비스하기 위하여, 각각의 서비스 에이전트는 정보저장소객체에 저장된 기술 정보들 중 필요한 데이터만을 가지고 다른 호스트로 이동하여 동작함으로써 불필요한 데이터의 발생을 최대한 억제한다. 또한, 특정 시간대 로컬에 존재하는 에이전트들만이 상호작용을 통해 동작하여 최소한의 메모리만을 가지고 동작을 수행한다. 이를 이용하여 다른 프로세스, 또는 시스템에 부담을 주지 않을 수 있다. 그리고 클라이언트 호스트에서 실행 될 객체들은 각각의 시스템 성능에 맞추어 생성되어있는 에이전트가 동작을 하여 다양한 모바일 장치의 지원이 가능하다. 실행객체는 서비스를 제공한 후 사용자의 응답을 가지고 서버 시스템으로 이동하거나, 시스템이 동작하면서부터 호스트들을 이동하며 데이터를 수집하는 객체를 통하여 사용자의 식사습관, 취침습관 등의 사용자 행동 패턴을 수집하고 서버 시스템에 전달한다. 이러한 정보들을 가지고 사용자의 다음 상황을 예측할 수 있으며 조금 더 지능적인 시스템으로의 지원이 가능하다.

또한, 헬스케어 상황정보 서비스 지원을 위한 정보수집 에이전트는 네트워크를 통해 연결되어 있는 모바일 기기들로 이동하며 가정 내에 설치되어 있는 센서 및 헬스케어 기기들로부터 위치 정보, 건강정보, 쾌적환경정보, 이벤트정보를 수집한다. 정보수집 에이전트가 수집한 정보를 <위치>, <시간>, <위치, 시간> 규칙에 맞는 정보로 생성하면 또 다른 에이전트는 사용자에게 가장 알맞은 맞춤형 헬스케어 상황정보를 유추하고 서비스 한다. 이러한 서비스를 하기 위하여, 사용자가 논리적인 영역을 이동하면 에이전트가 사용자를 따라다니며 이동성을 지원하고, 홈 환경에 존재하는 시스템의 가용성을 높이기 위하여, 다양한 모바일 장치 환경을 지원하기 위하여 멀티 에이전트 플랫폼이 요구된다. 표 1에서 Odyssey와 Aglet은 원격의 객체 제어의 지원하지 못하고, 보안성과 객체의 이동에서 제한적이라는 단점이 있

다. 반면 Voyager와 JADE는 강력한 기능을 가지고 있고, 비슷한 특징이 있는데, FIPA의 규정에 따른 플랫폼과, 확장성이 높고, 모바일 기기 지원 플랫폼을 가지고 있는 JADE를 사용하여 헬스케어 상황정보 서비스 지원을 위한 멀티 에이전트 플랫폼을 구축 한다.

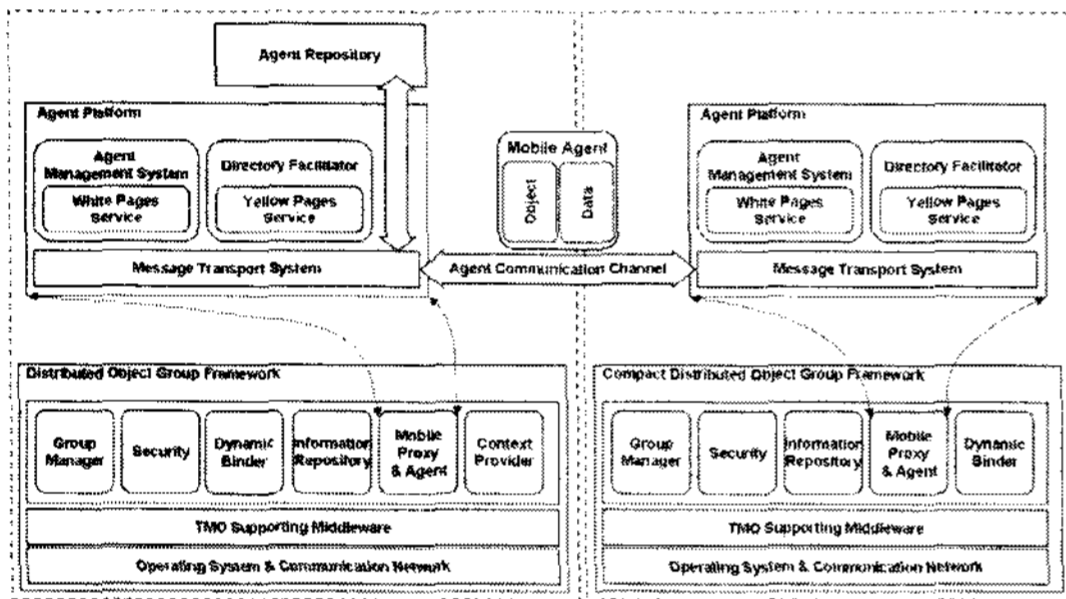
### 3. 헬스케어 상황정보 서비스 플랫폼

본 장에서는 헬스케어 상황정보 서비스 제공을 위한 멀티 에이전트기반의 소프트웨어 플랫폼구조와 구성요소간의 상호작용 그리고 에이전트의 이동성을 지원하기 위한 필요 요소들에 대해 기술한다.

#### 3.1 헬스케어 상황정보 서비스 플랫폼 구조

본 논문에서 제안하는 멀티 에이전트기반 헬스케어 상황정보 서비스 플랫폼은 u-헬스케어 지원 분산 프레임워크를 기반으로 하며, 멀티 에이전트 플랫폼인 JADE를 포함한 소프트웨어 환경이다. 특히, u-헬스케어 지원 분산프레임워크의 구성요소인 모바일 프락시&에이전트는 DOGF와 JADE의 인터페이스 역할을 한다. 이로 인하여, 제안한 플랫폼은 분산 객체의 객체그룹 관리 서비스와 이동 서비스를 제공한다. 상황정보 서비스 제공을 위해 DOGF는 상황정보를 생성하고, JADE는 상황에 적합한 에이전트를 생성한다. 이때, 에이전트가 이동하게 될 장치가 모바일 장치이면 모바일 장치의 CDOGF와 상호작용 한다. CDOGF는 DOGF에서 필요한 요소들만을 선택하여 메모리가 작은 PDA와 같은 모바일 장치에서 분산객체를 관리할 수 있도록 재구성한 프레임워크이다. 각각의 특정 동작을 하는 에이전트들은 평상시 Agent Repository에 저장되어 있다가 상황에 따른 호출이 있을 경우 Agent Communication Channel을 통하여 사용자가 위치하고 있는 곳의 모바일 장치로 이동하여 멀티 에이전트로써의 상호작용 과정

을 거쳐 데이터를 수집, 분석 및 서비스 한다. 모든 태스크를 마친 에이전트들은 다시 서버로 돌아와 수집한 정보를 관리 역할의 에이전트에게 전달하거나 가지고 있는 잔여 태스크를 처리한 후 소멸된다. 그림 2에서 볼 수 있듯이 에이전트는 실행 객체와 데이터를 가지고 이동을 하기 때문에 외부 간섭으로 생기는 오류를 최소화 할 수 있다. 또한 적은 대역폭을 사용하기 때문에 다른 프로세스와의 충돌 가능성도 줄일 수 있다. 다음 그림 2는 본 논문에서 제안하는 헬스케어 상황정보 서비스 제공을 위한 멀티 에이전트 소프트웨어의 플랫폼이다.



(그림 2) 헬스케어 상황정보 서비스 플랫폼

### 3.2 헬스케어 상황정보 서비스를 위한 모바일 에이전트의 동작과정

헬스케어 상황정보 서비스 플랫폼은 이질적인 환경에서 사용자를 따라다니며 사용자가 어떤 기기를 사용해도 같은 서비스를 연속적으로 제공할 수 있도록 지원해야 한다. 이를 위해서 에이전트들은 각각의 특성에 맞게 고정적이거나 이동성을 가져야 한다. 또한, 서비스를 수행하거나 데이터를 수집하는 에이전트들에게는 반드시 이동성의 지원이 필요하다. 이러한 모바일 에이전트는 상황에 맞는 장치들을 찾아 움직이게 되는데, 모바일 에이전트가 움직이게 되면 그 장치에 있는 JADE에서 실행이 되고, DOGF에서 에이전트가 동작할 동안 발생하는 정보를 받아 서비스 객체에게

전달하여 처리한다. 이러한 모바일 에이전트는 다양한 모습으로 홈 환경 내의 클라이언트 호스트들을 이동하게 되는데, 에이전트가 이동하는 방법은 헬스케어 상황정보 서비스를 제공하기 위해 실행객체의 동작 목표와 이동 대상 장치의 특성 그리고 역할에 따라 링형 순환 방식, 그물형 순환 방식, 분기식 순환 방식으로 구분 할 수 있다.

링형 순환 방식은 자신의 양쪽에 있는 호스트와 일대 일 회선 연결을 하여 동일한 네트워크 내에 있는 호스트들을 순차적으로 돌아다니는 방식이다. 모바일 에이전트는 프로그램을 가지고 있는 Agent Repository에서 출발하여 A호스트로 이동한다. A호스트에서의 태스크를 모두 마친 에이전트는 B호스트 이동하여 같은 동작을 반복한다. 모든 호스트에서의 일을 마친 에이전트는 다시 Agent Repository로 돌아와 수집한 데이터 처리 에이전트에게 전달한다. 이 방식은 적절한 환경을 유지하기 위한 목적으로 각 방의 온도를 확인 하는 등의 환경정보를 수집하는 동작을 할 때 사용한다.

그물형 순환 방식에서 하나의 호스트는 다른 모든 호스트들과 일대 다의 연결성을 가지고 모바일 에이전트가 모든 호스트를 돌아다니는 방식이다. 모바일 에이전트는 이러한 순환 방식을 통하여 규칙적이지 않은 사용자의 이동성을 지원한다. 특정 호스트의 영역에서 임의의 다른 호스트의 영역으로 사용자가 이동하였을 경우 모바일 에이전트는 서버를 거치지 않고 바로 이동하여 연속적인 서비스를 지원한다. 헬스케어 상황정보 서비스를 지원하기 위한 에이전트에 순환 방식을 사용하는데, 서버에서 제공되는 상황정보를 사용자에게 알리는 동작을 한다. 이 순환 방식의 또 다른 장점은 호스트 중 하나의 장치에 오류가 있을 경우 다른 장치로 대체할 수 있다는 장점을 가지고 있다.

마지막으로 분기식 순환 방식이 있는데 이 방식은 하나의 에이전트가 같은 시간에 동일한 네트워크에 있는 모든 호스트에서 같은 태스크를

처리 할 수 있는 방식이다. 즉, 에이전트는 자신과 동일한 복제 에이전트를 생성하여 각각의 호스트들에 보낸다. 각각의 에이전트들은 담당한 호스트로 이동하여 태스크를 처리한 후 다시 Agent Repository로 돌아와 자신의 태스크를 마친 후 스스로 소멸한다. 분기식 순환 방식은 갑작스런 변화가 있는 환경정보가 제공될 경우 홈 환경 내의 정보를 수집하거나 사용자가 위급한 상황에 있을 경우 등록되어 있는 보호자들에게 알림서비스를 제공할 수 있다.

#### 4. 헬스케어 상황정보 서비스를 위한 멀티 에이전트

본 장에서는 멀티 에이전트 기반 헬스케어 상황정보 서비스 플랫폼 상에서 헬스케어 상황정보 서비스와 수행 과정을 보인다. 특히, DOGF 구성 요소와 JADE 플랫폼의 상호작용을 통하여 서로 다른 에이전트들이 헬스케어 상황정보 서비스를 제공하기 위한 상호작용 및 방법과 각각의 장치들에서 에이전트가 동작하는 수행 과정을 기술한다.

##### 4.1 헬스케어 상황정보 서비스

헬스케어 상황정보 서비스는 주변 환경으로부터 수집되는 모든 정보들로부터 사용자에게 가장 적합한 응용을 통하여 제공하는 서비스를 말한다. 즉, 센서 및 기기들로부터 수집된 정보를 위치정보, 건강정보, 환경정보, 이벤트정보로 분류하고 사용자의 상황을 인식하여 사용자를 위한 위치인식, 스케줄 리마인더, 건강 처방, 주의 알림, 음악 치료 서비스를 제공한다.

이러한 서비스를 제공하기 위해서는 사용자 및 각종 제공 장치의 위치를 정확하게 인식하는 위치인식 서비스가 필수적이다. 본 헬스케어 상황정보 서비스에서는 각각의 위치를 파악하는 방법으로 ON/OFF 스위치와 Cricket 센서를 이용하여 센서 및 장치들의 위치와 사용자의 행위에 따라 발

생되는 이벤트 정보를 활용하였다. 이는 위치센서의 고유 ID와 홈 환경과의 매칭 과정을 통하여 위치 파악이 가능하게 하였으며, 주변 환경이 변경될 경우 지원영역을 재구성하여 위치에 기반을 둔 서비스를 지원한다. 또한 홈 영역을 지역단위로 구분하고, 지정한 영역 범위 안에 위치한 장치에서 제공되는 서비스 기능도 함께 연관하여 그룹으로 지정한다. 이렇게 위치인식 서비스에 따른 그룹화 된 영역의 클라이언트 호스트에 상황정보 데이터를 송신함으로써 이동경로에 따라 적시에 가장 적합한 장치를 통한 서비스 제공이 가능하다.

헬스케어 상황정보 서비스는 위치인식 서비스를 기반으로 스케줄 리마인더, 건강 처방, 주의 알림, 음악 치료 서비스의 제공이 가능한데 서비스를 위한 데이터 타입은 음성과 텍스트의 형태로 사용자에게 제공한다. 먼저, 사용자의 위치에 기반을 두어 클라이언트 호스트가 결정되면 상태정보 에이전트는 선택된 클라이언트 호스트의 성능을 확인하고 서비스가 가능한 실행객체를 음성 또는 텍스트 데이터와 함께 이동하여 동작한다. 또한 이동성을 가진 에이전트들은 사용자를 실제로 따라다니기 때문에 연속적인 서비스의 제공이 가능하다.

이 때, 제공되는 헬스케어 상황정보 서비스는 홈 환경 내의 사용자의 위치 변화에 따라 주변 상황정보를 기반으로 가장 적합한 헬스케어 서비스를 각각의 영역에서 제공한다. 상황정보 서비스는 처방·주의 및 스케줄과 같은 알림서비스를 제공하기 위해서 적용한 상황정보 생성방법으로 4W1H 상황정보 모델과 이를 기반으로 하는 이벤트 규칙은 <식 1>, <식 2>와 같다.

*Context*( $r1<who>$ ,  $r2<where>$ ,  $r3<when>$ ,  $r4<what>$ ,  $r5<how>$ )

<식 1>

*EventRule*( $r1 \wedge r2 \wedge r3$ )

<식 2>

위 모델에서는 *who*, *where*, *when*, *what*, *how*와 같이 5가지의 상황정보가 생성되는데, 이 상황정보는 이벤트 규칙에 맞추어 <위치>, <시간>, <위



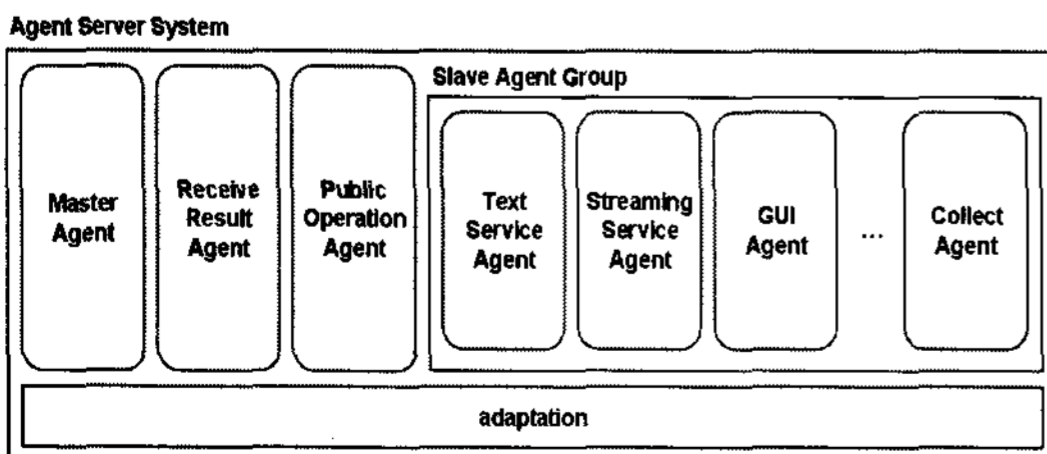
치, 시간>에 활성화 된다. 즉, 사용자가 지정한 이벤트 규칙 <식 2>에 따라 상황정보가 활성화 된다. 다음 표 2는 위에서 정의한 <식 1>과 <식 2>가 참조하는 헬스케어 상황정보 서비스 수행을 위한 상황정보 리스트를 나타내고 있다.

(표 2) <위치>, <시간>, <위치, 시간> 기반 상황정보 리스트

	Person (who)	Location (where)	Time (when)	Message (what)	Type (how)
Location	Any	Heather	Any	<거실 온도 18℃, 습도 38%입니다.>	Text
		Bathroom	Any	<변재 수은 20℃, 격정 온도 30℃입니다.>	Text
		Refrigerator	Any	<내일 내시경 검사입니다. 금식하세요.>	Voice
Time	Any	...	...	...	...
		Any	07:30	<혈압을 측정하세요.>	Voice
		Any	13:00	<식사 후 혈당을 체크하세요.>	Text
		Any	15:00	<오후 4시에 병변진료가 예약되어 있습니다.>	Voice
Location + Time	Any	Kitchen	08:00	<결장식단은 상곡과 원장국입니다.>	Voice
		Kitchen	21:00	<가스 밸브를 점검하세요.>	Voice
		Porch	22:00	<밤에 온도가 너무 낮습니다. 외출을 삼가세요.>	Voice

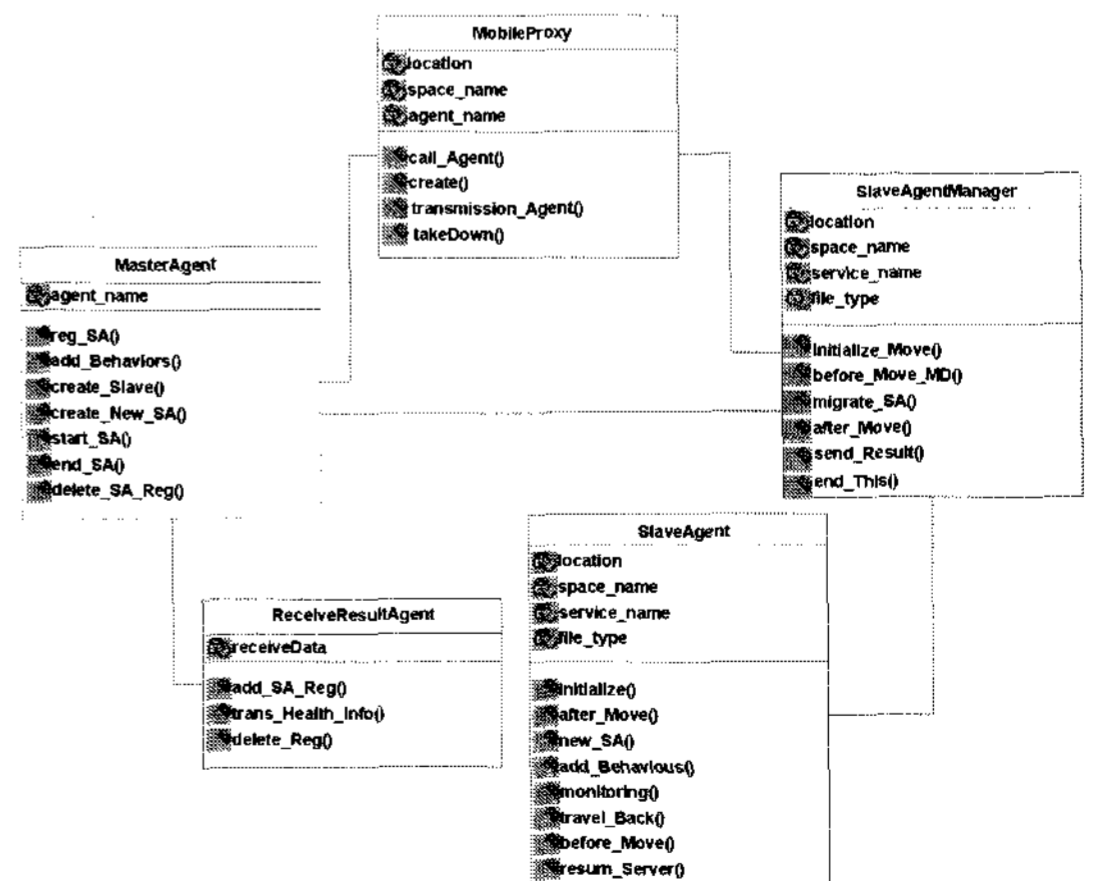
### 4.2 멀티 에이전트 구조

헬스케어 상황정보 서비스를 지원하는 에이전트들은 그룹관리자에 의하여 클라이언트 호스트로 이동하여 텍스트 서비스를 지원하거나 스트리밍 서비스를 지원한다. 이동을 한 에이전트들은 그룹관리자객체가 관리하는 센서 및 기기들의 정보를 수집하는 등, 태스크를 수행한다. 또한, 필요에 따라 다른 특성을 가진 에이전트를 추가로 생성할 수 있다. 이와 같은 동작을 하는 에이전트 시스템을 다음 그림 3과 같이 구분하였다.



(그림 3) 멀티 에이전트 서버 시스템의 에이전트

에이전트들은 DOGF에 있는 그룹관리자객체의 지시에 따라 클라이언트 호스트로 이동하여 동작한다. 이때, 클라이언트 호스트의 시스템 사양에 따라 서로 다른 서비스가 가능한 에이전트들이 이동한다. 그림 4의 Agent Server System에서는 그룹관리자객체로부터 사용자의 위치를 확인받아 사용자의 가까이에서 인터페이스를 제공할 수 있는 모바일 또는 고정기기들을 검색하여, 가장 적합한 모바일 기기를 선택한다. 이렇게 선별된 기기의 CPU, Memory의 시스템 사양을 파악한 후 선택되어진 모바일 기기에 대하여 적합한 동작을 하는 에이전트들을 선별한 후 서비스를 시작한다. 노트북이나 데스크톱과 같이 성능이 높은 호스트에서는 상황정보제공자객체에서 제공하는 음성 정보를 서비스 할 수 있고, PDA와 같이 다소 낮은 사양의 호스트에서는 Text를 화면에 표시함으로써 상황에 맞는 정보를 제공할 수 있다. 이 때 Public Operation Agent가 사용자 정보 확인 명령이 있을 때마다 링형 순환 방식으로 각각의 시스템들을 이동하며 서비스가 가능한 기기를 파악하고 적합한 서비스를 제공하기 위해 다른 에이전트를 선별한다. 이러한 구조를 보이는 멀티 에이전트 시스템의 구조를 다음 그림 4와 같은 클래스 다이어그램으로 나타내었다.



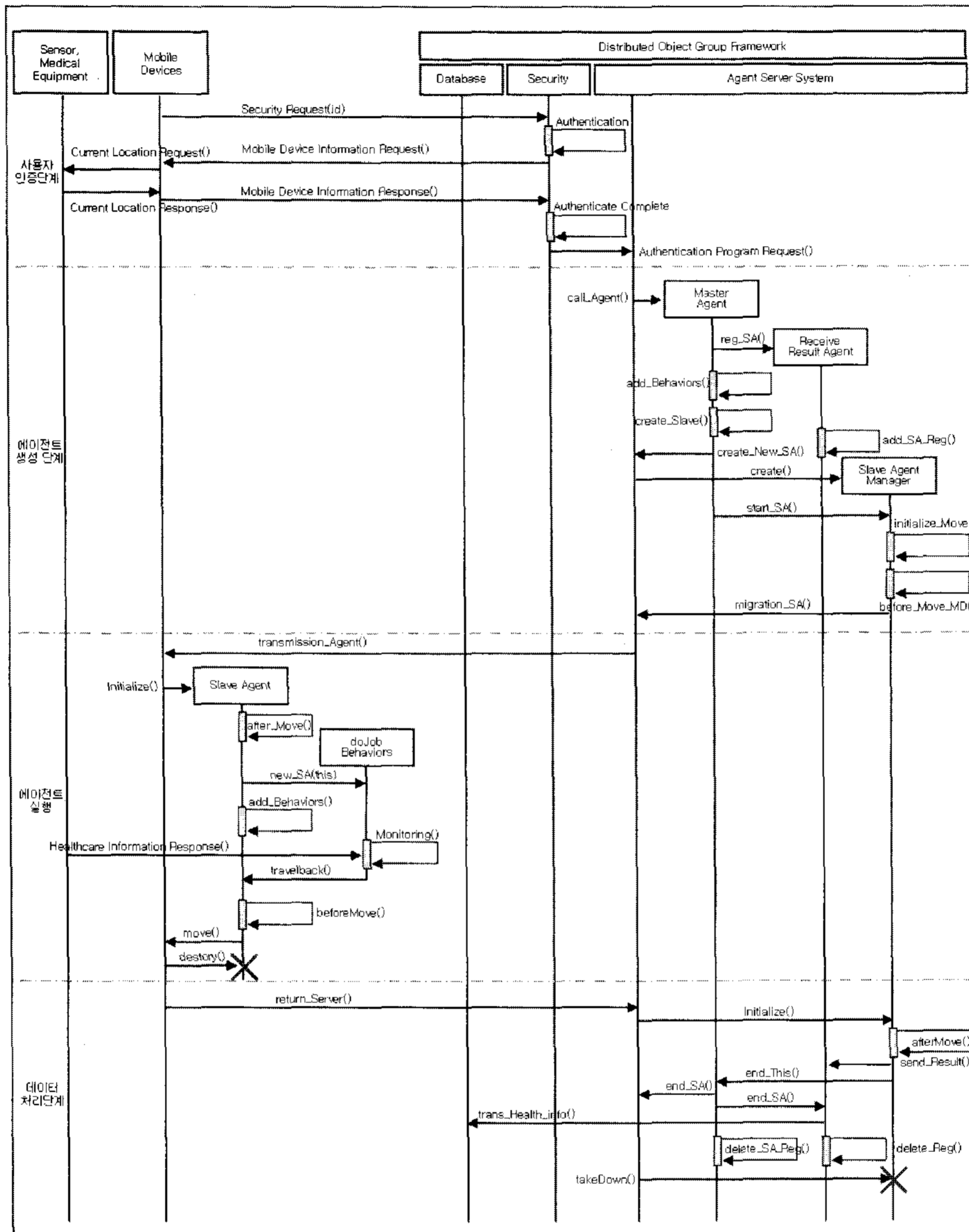
(그림 4) 헬스케어 상황정보 서비스 클래스 다이어그램

플랫폼이 준비되면 Master Agent, Receive Result Agent, Public Operation Agent, Slave Agent Group 등 모든 에이전트들은 에이전트 서버 시스템에 실행객체로 준비된다. 본 논문에서의 Slave Agent Group은 Text Service Agent, Streaming Service Agent, GUI Agent, Collect Agent로 구성된다. 이러한 Slave Agent Group의 Slave Agent들은 사용자의 상황에 맞게 개발하여 추가될 수 있으며, 다른 Management 또는 Slave Agent들과 함께 동작하게 된다. 우선, 에이전트 서버 시스템에서는 DOGF의 상황정보 제공자로부터 상황정보를 제공 받아 Slave Agent의 이동 및 호출, 그리고 태스크가 끝난 에이전트의 진행을 종료의 동작을 제어 한다. Master Agent는 Slave Agent가 동작할 기기들의 특성을 파악하고 Agent Server System에 필요한 에이전트 객체를 요구한다. 생성되는 에이전트를 Receive Result Agent에 등록, 삭제하는 역할과 에이전트의 이동 명령을 내리기도 한다. Receive Result Agent는 현재 동작하고 있는 에이전트의 정보를 가지고 있으면서 태스크를 마친 에이전트들이 가지고 온 정보를 수집하여 정보저장소객체에 데이터들을 제공한다. Slave Agent Group은 클라이언트 호스트에서 동작할 에이전트들을 말하며 각각의 에이전트들은 나름대로의 특성을 가지고 동작한다. 예를 들어, 스트리밍 서비스 에이전트 같은 경우는 클라이언트 호스트의 시스템 환경이 스트림 데이터처리를 지원 할 경우 클라이언트 호스트로 이동하여 서버에서부터 가져온 스트림 데이터를 사용자에게 서비스한다. GUI Agent 또한 클라이언트 호스트로 이동하여 서비스에 따른 사용자 GUI환경을 제공한다. Slave Agent들은 각각의 에이전트가 이동성을 지원하며 그 기능이 다르기 때문에 서로 상호 보완관계를 가지고 상호동작을 한다. 즉, PDA에서 현재 사용자 주변의 환경정보를 알고 싶을 때는, Text Service Agent와 GUI Agent, Collect Agent가 이동하여 Collect Agent는 환경정보를 수집하고 Text Service Agent를 통하여 GUI Agent에 표시해 주는

멀티 에이전트 형식의 기능을 한다.

#### 4.3 헬스케어 상황정보 서비스 수행 과정

멀티 에이전트를 기반으로 하는 헬스케어 상황정보 서비스의 플랫폼에서 상황정보 서비스를 제공하기 위해서는 사용자 인증, 에이전트 생성, 에이전트 실행, 데이터 처리의 네 단계의 수행과정을 거친다. 사용자 미리 시작되어 있던 플랫폼을 통해서 인증 단계에 들어가게 되는데, 인증 단계에서는 서비스의 제공, 또는 정보 수집의 요청이 발생할 경우 클라이언트 호스트의 현재 위치를 요구한다. 요구를 받은 클라이언트 호스트는 센서로부터 위치정보를 수집하여 DOGF의 보안객체로 전송하고 보안객체에서는 주기적으로 Public Operation Agent가 수집해온 사용자 위치정보와 비교하여 Slave Agent의 동작을 허가한다. 보안객체에서는 사용자 인증단계에서 생성된 인증확인 메시지를 에이전트 서버 시스템으로 전송하며, 이때 상황 제공자객체에서는 헬스케어 상황정보 생성 모델인 4W1H와 이를 기반으로 하는 이벤트 규칙을 적용하여 헬스케어 상황정보를 생성한다. 생성되어진 상황정보는 DOGF의 상황정보 제공자를 통하여 에이전트의 동작을 결정하게 되는데, 동작되어질 에이전트의 정보와 서버 시스템에 서비스에 필요한 데이터도 함께 제공한다. 사용자의 인증에 대한 단계가 마치면 바로 에이전트의 생성단계로 넘어가게 되는데, 이 때 미리 생성되어 있는 슬레이브 에이전트들 중 적합한 에이전트를 선별하고 등록하는 과정을 가진다. Master Agent는 에이전트 서버 시스템으로부터 데이터와 에이전트들의 호출 명령을 받게 되면 자신과 Receive Result Agent의 Registry에 동작하게 될 Slave Agent들의 이름을 등록한다. 그리고 어떤 에이전트를 동작시킬 것인지 수집되어 있는 위치, 시스템 성능 정보와 서비스 정보를 에이전트 서버 시스템에 전달한다. 서비스 되어질 에이전트들이 결정되면, 에이전트 서버 시스템은 Slave Agent Manager



(그림 5) 멀티 에이전트 시스템에서 모바일 에이전트의 동작 과정

에게 Slave Agent들을 준비 시키라는 메시지를 전송하고 Master Agent에서는 Slave Agent들의 생성 초기화와 동작에 필요한 준비 단계를 지시한다. Slave Agent들은 클라이언트 호스트로 이동하기 전에 초기화 되고 서비스 하게 될 데이터를 정리 하는 준비 과정을 거치게 된다. 모든 준비가 완료 되면 에이전트 서버 시스템은 클라이언트 호스트

로 Slave Agent들을 이동시킨다. 다음 그림 5에서 이러한 동작을 하는 에이전트들에 대한 순차적 동작과정을 ETD로 나타내었다.

사용자인증 단계와 에이전트 생성 단계를 거쳐 클라이언트 호스트로 이동한 Slave Agent는 에이전트 실행단계로 접어들게 된다. 에이전트 실행 단계에서는 Slave Agent들이 각각의 태스크를 처

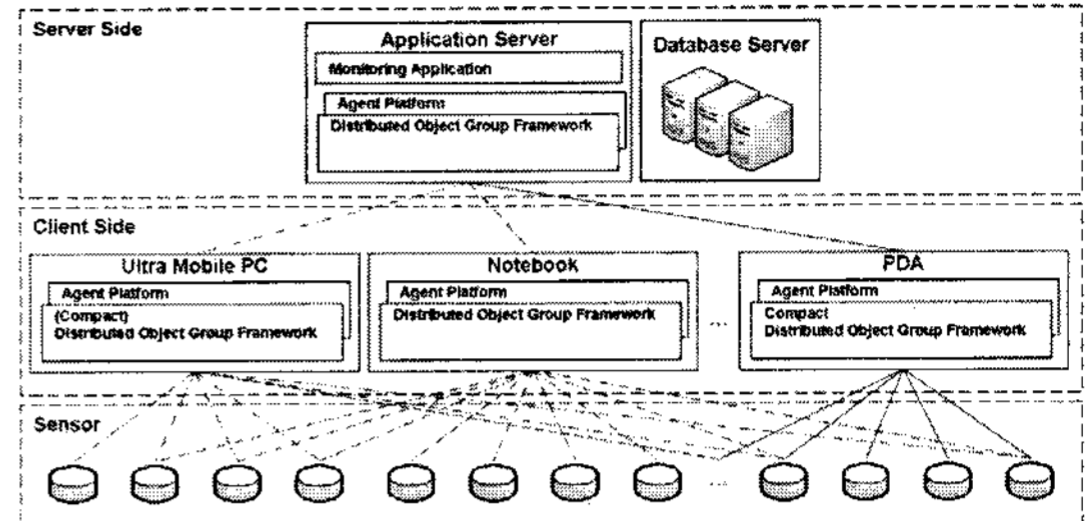
리하기 시작하는데, GUI Agent와 같이 서비스 지원 에이전트가 가장 먼저 동작하고, Text Service Agent와 Streaming Service Agent와 같은 서비스에이전트를 통하여 상황 제공자 객체에서 생성된 헬스케어 상황정보 데이터를 사용자에게 Text 또는 음성의 형태로 서비스한다. 서비스를 마친 Slave Agent들은 스스로 제거되어 사라지지만, Collect Agent와 같이 정보를 수집하는 Slave Agent는 사용자의 응답에 따라 그 응답정보를 가지고 Agent Server System으로 돌아온다. 마지막으로 데이터 처리단계에서 사용자 응답정보를 가지고 돌아온 Slave Agent들은 응답정보에 대한 데이터를 처리 한 후 Receive Result Agent로 결과 값을 전송한다. Receive Result Agent는 수신한 정보들을 정보저장소

객체에 저장하여 다음 상황정보를 위한 데이터로 활용한다. Slave Agent Manager는 수행을 모두 마친 Slave Agent들의 삭제 메시지를 Master Agent에 보낸다. Master Agent는 자신과 Agent Server System, 그리고 Receive Result Agent의 Registry에 있는 Slave Agent의 데이터를 삭제 지시한다. Receive Result Agent도 Registry를 비롯한 잔여 데이터를 삭제하고, Agent Server System은 Slave Agent를 삭제하라는 명령을 Slave Agent Manager에게 송신한다. Slave Agent Manager의 Slave Agent들도 이 메시지를 받게 되면 모든 라이프 사이클을 마치고 스스로 종료한다.

#### 4.4 헬스케어 상황정보 서비스 제공을 위한 멀티 에이전트 시스템 환경

다양한 센서 및 모바일 장치 그리고 기기들로 구성된 분산 플랫폼과 이들 간의 통신을 위한 무선 통신 인프라를 포함한 시스템 환경에 대한 내용을 기술하고, 홈 환경 내에서 헬스케어 상황정보를 서비스를 위한 멀티 에이전트 시스템에 관한 환경을 구축 하였다. 다음 그림 6에서는 이러한 동작을 하는 시스템 구성과 소프트웨어 구성

요소 배치 상황과 이들간의 상호관계를 나타내고 있다.

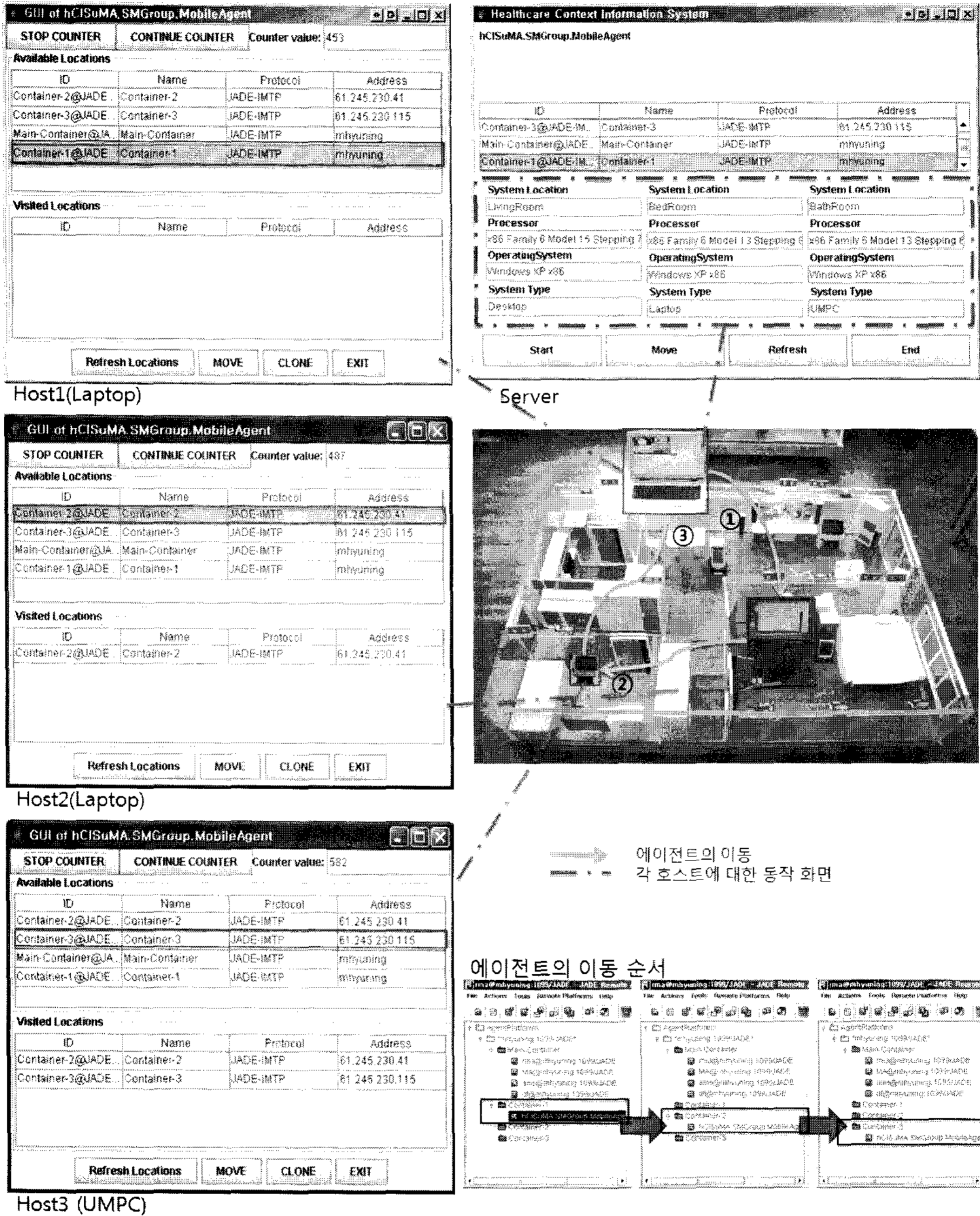


(그림 6) 헬스케어 상황정보 서비스 플랫폼의 시스템 환경

서버 측에는 분산되어 있는 모바일 장치와 같은 객체들을 관리하기 위한 DOGF와 DOGF로부터 제공받는 상황정보를 사용자에게 서비스하기 위한 에이전트 플랫폼인 JADE가 있다. 또한, 응용 서버에는 관리자와의 인터페이스를 위한 모니터링 어플리케이션이 동작한다. 그리고 각종 정보 및 에이전트의 보관을 위한 데이터베이스 서버가 존재한다. 이와 같은 서버의 소프트웨어 환경을 통하여 각종 에이전트가 클라이언트 측의 클라이언트 호스트로 이동하게 되는데 PDA에는 CDOGF와 JADE가 있고 UMPC와 노트북에는 DOGF 또는 CDOGF와 JADE가 있어 서버로부터 받은 에이전트들의 동작을 관리한다. 이 때, 이동한 에이전트는 사용자에게 서비스를 제공하거나 물리적 장치들로부터 각종 정보를 수집하는 등의 동작을 한다. 모바일 장치의 DOGF 또는 CDOGF는 환경정보, 건강정보, 위치정보를 제공하는 센서 및 물리적인 장치들을 관리한다.

헬스케어 상황정보 서비스를 위해 홈 환경 내에 있는 센서 및 기기들로부터 환경정보, 건강정보, 위치정보를 수집한다. 환경정보는 Nano-Q<sup>+</sup>에서 온도, 습도, 조도 등의 정보를, u-Star에서는 동체감지와 진동정보를 수집할 수 있다. 건강정보는 Clever에서 혈당, 혈압, 맥박의 정보를 수집할 수 있으며, Webdoc을 통하여 체지방 정보를 수집할





(그림 7) 헬스케어 상황정보 서비스 시스템 정보수집 에이전트의 동작

수 있다. 위치정보는 Cricket과 On/Off 스위치가 있는데 각각 초음파와 사용자 이벤트를 이용하여 위치정보를 수집한다. 또한, Hmote2420은 실내 위

치추적을 위한 초음파센서와, 온도, 조도, 습도, 이산화탄소량, 미세 먼지량 측정과 같은 환경정보를 수집하며, 맥박센서와 비접촉 체온 측정센서로



서 건강정보를 수집할 수 있다. 모바일 장치로는 PDA와 UMPC, 노트북 등을 사용하며 이를 통해 다양한 환경에서의 에이전트의 이동성 지원이 가능함을 보인다.

서버에는 현재 가용한 슬레이브 에이전트들의 목록과 서버에 연결 되어 있는 클라이언트 호스트 리스트, 각 각의 호스트들에 대한 시스템 정보를 표시하는 영역으로 나뉘어 있다. 이 프로토타입에서 제공하는 슬레이브 에이전트는 홈 환경내의 장치들을 돌아다니며 시스템 정보를 수집하는 에이전트로서 이동 가능한 호스트와 지나온 호스트를 표시하는 영역으로 구분할 수 있다. 시스템 정보 수집 에이전트는 두 개의 에이전트로 구성되어 서로 상호작용 과정을 가지며 동작한다. 시스템 정보 수집 에이전트는 시스템의 정보를 수집하고 현재의 에이전트의 위치 확인, 마스터 에이전트와의 연결성을 지원하는 에이전트와 GUI 에이전트로 구분할 수 있다. 먼저 서버에서 플랫폼이 실행되면 정보 수집 에이전트가 생성되는데 이 에이전트는 Laptop과 UMPC를 사례로 이동하며 각각의 시스템의 정보를 수집한다. 모든 클라이언트 호스트로의 이동 및 정보 수집을 마친 에이전트는 다시 서버로 돌아와 수집한 데이터를 Receive Result Agent에게 전달하고 Receive Result Agent는 받은 데이터를 분석하여 저장하고 서버의 GUI에 표시하게 된다. 위 그림 7은 이러한 동작을 하는 멀티 에이전트 기반 헬스케어 상황정보 서비스 플랫폼의 프로토타입을 나타낸 그림이다.

## 5. 결론 및 향후 연구

유비쿼터스 컴퓨팅 환경에서의 헬스케어 상황정보 서비스는 장소와 상황에 제약받지 않고 사용자의 편의를 위한 서비스가 가능해야 한다. 개인의 건강을 유지하고 관리해야 한다는 점에서 사용자에게 확실적인 서비스를 제공해야 함은 물론이며, 건강과 관련되어 있는 분야이니 만큼 서

비스를 이용하는 사용자가 불편함이 없어야 한다. 특히, 사용자의 상황에 따라 다양한 분산 플랫폼의 유무선 네트워크 인프라를 기반으로 정보를 수집하거나 교환하며 실시간 정보 서비스를 제공해야 한다.

본 논문에서는 분산 응용 서비스를 수행하는 객체들의 그룹 관리 및 단일 시스템 환경의 논리적인 체계를 만들기 위하여 물리적으로 분산된 객체들 간의 복잡한 인터페이스에 대한 투명성을 제공하는 분산객체그룹 프레임워크와 이질적인 환경에서 에이전트의 생산 및 생명 주기의 관리를 지원하는 프레임워크, JADE를 활용하여 멀티 에이전트기반의 헬스케어 상황정보 서비스 제공을 위한 플랫폼을 구축하였다. 헬스케어 상황정보는 분산객체그룹 프레임워크의 상황 제공자객체에서 4WIH 상황정보 모델에 의해서 생성된다. 상황 제공자객체로부터 생성된 헬스케어 상황정보는 각종 슬레이브 에이전트를 통하여 사용자 맞춤형 서비스를 지원한다. 헬스케어 상황정보 서비스를 제공하는 것 외에도 상황정보를 생성하기 위한 정보를 수집하는 수집에이전트도 연속적인 동작을 하며 네트워크 환경을 이동한다. 이 때, 분산객체그룹 프레임워크의 모바일 프락시&에이전트 객체는 분산객체그룹 프레임워크와 JADE의 인터페이스 역할로 적절한 클라이언트 호스트를 선택하고 사용자에게 필요한 슬레이브 에이전트들을 선택하여 서비스를 지원한다. 멀티 에이전트 패러다임은 기존 클라이언트/서버 패러다임의 빈번한 상호작용에서 비롯한 실시간 처리시간의 지연문제, 그리고 장소에 따라 변하는 상황을 지원하지 못하는 문제점을 해결할 수 있다. 따라서 본 헬스케어 상황정보 서비스 플랫폼은 사용자에게 헬스케어 상황정보 서비스를 지원하기 위한 빠른 데이터의 처리율 향상과 이질적인 환경에서도 연속적으로 사용자의 상황에 적합한 맞춤형 서비스의 지원이 가능하다.

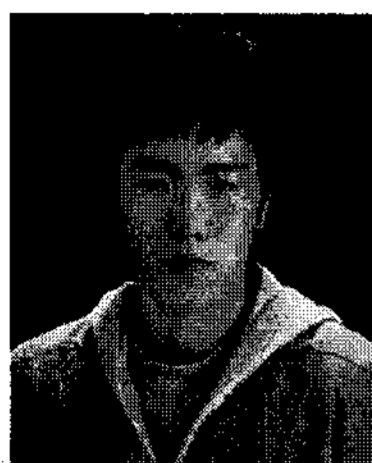
향후 연구로는 본 논문에서 설계한 멀티 에이전트 기반 헬스케어 상황정보 서비스 플랫폼과

기존 상황정보 시스템과의 비교 분석을 통하여 본 플랫폼의 우수성을 보이고 멀티 에이전트 플랫폼의 구현을 통해 다양한 클라이언트 호스트의 단말 상에서 멀티미디어 상황정보 서비스를 개발하고자 한다.

## 참고 문헌

- [1] Kum Yang, Alex Galis, "Policy-driven Mobile Agents for Context-aware Service in Next Generation Networks", Mobile Agents for Telecommunications Applications, MATA, LNCS 2881, pp.111-120, 2003.
- [2] Chang-Sun Shin, Chung-Sub Lee, and Su-Chong Joo, "Healthcare Home service System Based on Distributed Object Group Framework", Lecture Notes in Computer Science, Vol. 3983, pp.786-807, 2006.
- [3] Emerson Ferreira de Araujo Lima, Patrica Duarte de Lima Machado, JorgeCesar Abrantes de Figueiredo, Flavio Ronison Sampaio, "Implementing Mobile Agent Design Patterns in the JADE framework.", <http://jade.tilab.com/papers/EXP/Ferreira.pdf>.
- [4] K.N. Truong, G. D. Abowd, and J. A. Brotherton, "Who, What, When, Where, How: Design issues of capture & access applications", in Proceedings of the International Conference on Ubiquitous Computing, pp.209-224, 2001.
- [5] Bill Schilit, Norman Adams and Roy Want, "Context-aware computing applications", In proceedings of IEEE Workshop on Mobile Computing Systems and Applications, 1994.
- [6] Natalia Marasse, Chris Schmandt, "Location-aware information delivery with ComMotion", In Proceedings of Second International Symposium on Handled and Ubiquitous Computing, HUC 2000, Springer Verlag. Bristol, UK, pp.157-171, 2000.
- [7] Anind K. Dey, Gregory D. Abowd, "CyberMinder: A Context-aware system for supporting reminders", In Proceedings of second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, Springer Verlag, pp.172-186, 2000.
- [8] Foundation for Intelligent Physical Agents. Specifications. Available from <http://www.fipa.org>, 2007.
- [9] J. E. White, "Telescript technology : The foundation for the electronic marketplace.", General Magic White Paper, 1994.
- [10] Prithviraj Dasgupta, Nithy Narasimhan, Louise E. Moser, P. M. Melliar-Smith, "agNet: Mobile Agents for Networked Electronic Trading", IEEE Transaction on Knowledge and Data Engineering, Vol. 11, No. 4, pp.509-525, 1999.
- [11] Sergio Viademonte, Frada Burstein, "An empirical study of distribution based on Voyager : A performance analysis", Proceedings of the 33rd Hawaii International Conference on System Sciences, pp.2015-2023, 2000.
- [12] 박무현, 서종성, 신창선, 정창원, 주수종, "홈 환경에서 모바일 프락시를 이용한 상황정보 시스템 구축", 한국인터넷정보학회 학술지, 제 8권 1호, pp.189-194, 2007.
- [13] 이승율, 이근상, 전병국, 최영근, "에이전트의 모듈 및 서비스 객체 이동 설계", 한국정보과학회 논문집, Vol. 26, No. 2, pp.104-106, 1999.
- [14] 최성진, 강인상, 백맹순, 손진곤, 황종선, "다중 지역 이동 에이전트 컴퓨팅 환경에서의 동 에이전트 위치 관리 및 메시지 전달 기법", 한국정보과학회 논문집, Vol. 30, No, 1, pp.190-192, 2003.

● 저 자 소 개 ●



**박 무 현(Moohyun, Park)**

2007년 원광대학교 전기전자 및 정보공학부 졸업(학사)

2007년~현재 원광대학교 컴퓨터공학과 석사과정

관심분야 : 분산 컴퓨팅, 유비쿼터스, 에이전트, 상황인식

E-mail : mhyuning@wonkwang.ac.kr



**정 창 원(Chang-Won Jeong)**

1993년 원광대학교 컴퓨터공학과 졸업(학사)

1998년 원광대학교 컴퓨터공학과 졸업(석사)

2003년 원광대학교 컴퓨터공학과 졸업(공학박사)

2004년~2006년 전북대학교 차세대 LBS 응용 연구센터 연구교수

2006년~현재 원광대학교 전기전자 및 정보공학부 박사 후 연구원

관심분야 : 분산객체 컴퓨팅, 멀티미디어 데이터베이스, LBS, 텔레메틱스

E-mail : mediblue@wonkwang.ac.kr



**주 수 종(Su-Chong Joo)**

1986년 원광대학교 전자계산공학과 졸업(학사)

1988년 중앙대학교 컴퓨터공학과 졸업(석사)

1992년 중앙대학교 컴퓨터공학과 졸업(공학박사)

1993년 미국 University of Massachusetts at Amherst, Post-Doc.

2003년 미국 University of California at Irvine, Visiting Professor.

1990년~현재 원광대학교 전기전자 및 정보공학부 교수

2007년~현재 원광대학교 정보전산원 원장

관심분야 : 분산 실시간 컴퓨팅, 분산객체모델, 시스템 최적화, 멀티미디어 데이터베이스

E-mail : scjoo@wonkwang.ac.kr