정보가전 기반의 건강관리 멀티 에이전트 시스템

박수진* : 이상민** · 김일곤***

Health Care Multi Agent System Based on Information Appliances

Su-Jin Park : Sang-Min Lee · Il-Kon Kim

요 약

정보가전에 대한 연구가 현재 활발히 진행 중에 있고 정보가전이 실현 되었을 때 다양한 서비스와 연계가 가능하고, 사람들의 생활환경에 많은 변화가 일어날 것으로 전망된다. 본 논문에서는 정보가전 기반의 건강관리 멀티 에이전트 시스템을 설계와 구현을 기술하였다. 홈 제어 네트워크로는 전력선 기반의 LonWorks를 이용하고, OSGI(Open Services Gateway Initiative)를 이용하여 LonWorks기반의 디바이스들을 제어한다. 멀티에이전트 시스템으로는 FIPA(The Foundation for Intelligent Physical Agents) Specification을 따르고, 이 플랫폼 위에서 건강관리 에이전트, 가전제품관리 에이전트, 건강관리 Library, 요리 Library, 운동 Library를 구현하였다.

1. 서론

인터넷의 발달로 인하여 모든 가전기기들이 서로 정보를 공유하고 통신하여 외부에서도 접근이 가능한 정보가전이 현재 연구 중에 있다. 정보가전이 생활화되면 외부에서의 접근뿐만 아니라 가전기기들이 가지는 정보를 사용하여 다양한 서비스가 가능하다. 서비스들은 프로그램의 형태로 홈 서버에 존재하는데 각각의 프로그램은 운영체제가 달라 서로 통신을 할 수 없고, 호환이 되지 않는 문제가 발생한다. 멀티 에이전트 시스템은 자치성을 가진 에이전트들이 모여 서로 지식을 교환 하며 자신의 목표를 추구하는 시스템으로 이러한 문제점을 해결하는데 적합하다.(Huhns&Singh, 1998)

건강관리 서비스의 중심이 되는 건강관리 에이전트는 인터페이스를 통해 사용자와 1:1 통신이 가능하며, 홈 서버에 있는 데이터베이스를 이용하여 건강관리에 필요한 정보들을 직접 관리한다. 건강에 대한 기본적 지식을 바탕으로 홈 서버에 저장된 자료를 분석하여 다른 Library와의 정보교환으로 건강관리 서비스를 수행한다.

건강관리 에이전트는 가전기기를 제어하는 가전제품 관리 에이전트를 통하여 정보가전과 연동한다. 가전기기들은 전력선 기반의 LonWorks를 가능하게 하는 Neuron Chip을 탑재하고, 가전제품 관리 에이전트는 OSGI를 이용하여 LonWorks 기반의 디바이스들을 제어한다. OSGI는 외부 네트워크와 내부 네트워크에 사용되는 다양한 프로토콜간의 bridge 역할을 하며 서비스의 전달 기능을 담당하는 역할을 한다.(OSGI)

외부 Contents인 건강관리 Library, 요리 Library, 운동 Library는 각 분야의 전문적인 지식을 바탕으로 건강관리 에이전트가 요청하는 정보를 제공해 준다. 본 논문에서는 다이어트에 최적화 된 Library를 구성하였다.

본 논문의 구성은 다음과 같다. 2절에서는 정보가전과 멀티에이전트 시스템 플랫폼인 FIPA OS에 대한 내용을 기술하고, 3절에서는 전체 시스템의 구성과 시나리오에 관해 기술하고 4절에서는 각 모듈별 기능과 동작에 관해 기술한다. 5절에서는 건강관리 멀티 에이전트 시스템의 개별화에 대해 기술하고 마지막으로, 6절에서는 건강관리 서비스의 향후 과제에 대해 기술하였다.

2. 정보가전과 멀티 에이전트 플랫폼

2.1 정보 가전

인터넷 정보 가전 기술이란 아파트 내의 PC와 프린터 등과 같은 PC 관련 기기는 물론 모든 가전기기들을 하나의 네트워크로 연결하여, 서로의 정보를 공유하고, 각각의 기기가 인터넷에 동시에 접속할 수 있으며, 이에 따라 내부 네트워크 혹은 인터넷을 통하여 외부에서도 제어를 가능하게 하는 첨단 통신 시스템이다. 이와 같은 기능을 수행하려면, 가정 내에는 각 기기들 간의 네트워킹이 형성

* 경북대학교 컴퓨터 과학과
** 경북대학교 컴퓨터 과학과
*** 경북대학교 컴퓨터 과학과

되어 상호 기기 간의 통신은 물론, 이를 통한 정보의 공유 및 엔터테인먼트 향유, 그리고 에너지 절약 기능과 홈오토메이션 기능 등을 제공할 수 있는 시스템과 소프트웨어가 지원되어야 한다.(전호인 and 송동일, 2001)

홈 네트워크는 여러 단체로부터 표준화가 진행되고 있으며, 그 형태도 크게 유선과 무선으로 분류된다. 유선형태의 대표적인 것으로 기존의 전화선을 이용한 HomePNA(Home Phoneline Networking Alliance), PC와 주변기기간 인터페이스인 IEEE 1394, USB(Universal Serial Bus), IEEE 1394b와 망 연동을 위한 인터넷 프로토콜을 기반으로 모든 댁내 장치들을 통합하기 위한 VESA Home Network 및 전력선(PLC) 등 다양한 형태가 존재하며, 무선형태로 HomeRF(Home Radio Frequency), Bluetooth 및 IrDA(Infrared Data Association)가 있다. 그리고 미들웨어 기술로는 JINI, UPNP(Universal Plug and Play), Havi(Home Audio Video Interoperability), OSGI가 있다.

본 논문에서는 홈 제어 네트워크로 PLC를 기반으로 한 LonWorks를 이용한다. LonWorks의 통신 방식은 정보 기반 통신(Information-based communication) 으로 기존의 명령어 기반 통신(command-based communication)과 구분된다. 각 디바이스는 네트워크 변수 (Network Variable : NV)라는 특별한 형태의 입출력 오브젝트를 이용하여 디바이스 간의 바인딩 정보만을 바꿈으로서 통신을 다시 설정할 수 있다. 이러한 네트워크 변수는 LonMark협회에서 표준적으로 정하여 표준 네트워크 변수(Standard Network Variable Type: SNVT)를 미리 정의함으로써 여러 밴더의 디바이스가 함께 운용될 수 있는 상호개방성을 제공한다. 모든 디바이스는 Neuron Chip을 탑재하고 있으며 이 칩은 LonWorks 디바이스의 핵심이라 할 수 있는 LonTalk 프로토콜을 가지고 있어 LonWorks 통신을 가능하게 해준다. 각 디바이스 개발자는 해당 디바이스의 동작을 기술하기 위해서 Neuron-C를 이용해서 동작을 구현한다. Neuron-C에서 NV의 동작 및 이벤트를 구체화하고 이에 따른 전체 네트워크 동작은 NV의 바인딩에 의해서 결정된다.(손영성, 2001)

OSGI는 가정 내에서의 어떤 프로토콜이라도 관계없이 OSGi Service Gateway Specification을 만족하는 service gateway 라면 지원이 가능하다. OSGi에서 서비스(service)는 특정 기능을 수행하는 자바 인터페이스와 실제 구현 객체를 말하고, 번들(bundle)은 그러한 서비스를 제공하기 위한 기능적 배포단위이다. 번들 컨텍스트(bundle context)란 번들의 life cycle을 관리하는 번들의 실행환경을 의미한다. 번들의 life cycle에서 번들 상태는 install, uninstalled, resolved, starting, stopping, active가 있다.(임동찬, 2001)

2.2 멀티 에이전트 플랫폼

FIPA는 에이전트들뿐만 아니라 에이전트를 기반으로 하는 어플리케이션들 사이에도 상호작용 하는 메시지 표준과 공개된 문서에 의해 인텔리전트 에이전트를 개발하는 방법을 정의하는 국제적인 조직이다. 즉, FIPA의 목적은 에이전트 어플리케이션들과 에이전트 시스템들과의 상호작용 하는 에이전트 표준을 만드는 것이다.(FIPA 2000 specification)

FIPA OS는 AMS(Agent Management System), DF(Directory Facilitator), MTS(Message Transport System), ACC(Agent Communication Channel)로 구성된 플랫폼이다. AMS는 에이전트 플랫폼의 필수적인 컴포넌트 중의 하나이고, 에이전트의 생성, 삭제, 어떠한 에이전트가 동적으로 등록되어야 하는지 그리고 어떠한 에이전트가 다른 에이전트 플랫폼으로부터 이주하는지에 대한 감시 등의 작업을 관리한다. DF는 에이전트 플랫폼의 필수적인 컴포넌트 중의 하나로 다른 에이전트에게 옐로우 페이지(yellow page) 서비스를 제공한다. 에이전트는 다른 에이전트와 에이전트 통신 언어(Agent Communication Language)를 사용하여 전송 규약(Transport Protocol)을 통해 메시지를 전달하는데 이때 MTS는 에이전트 구분자(Agent ID)를 보고 해당 에이전트에게 라우팅(routing)한다. 에이전트간의 통신을 위해 ACL(Agent Communication Language) 메시지를 사용한다.

건강관리 멀티 에이전트 시스템에서 FIPA OS는 에이전트와 Library가 동작하는 기본 플랫폼의 역할을 한다. 에이전트와 Library는 실행되면서 AMS와 DF에 자신을 등록하며, ACL 메시지를 만들어 MTS와 ACC를 통해 원하는 에이전트나 Library로 전송한다.

3. 전체 시스템 구성

3.1 시나리오

건강관리 멀티 에이전트 시스템의 동작을 알기 위해서 하나의 시나리오를 제시 한다. 세계 여성들의 관심사인 다이어트를 예를 들어 건강관리 멀티 에이전트 시스템의 구체적인 동작 모습을 알 수 있다.

가. 사용자가 건강관리 에이전트에게 다이어트를 할 것임을 알려준다. 건강관리 에이전트의 다이어트에 관한프로그램을 열어서 아래 사항을 입력한다.
   - 성별, 나이, 키, 몸무게, 맥박수, 체내지방수치, 혈압, 직업, 다이어트 하고 싶은 부위, 조절하고 싶은 부위와 수치(cm, inch, kg), 목표기간, 취미생활, 좋아하는 것과 싫어하는 것 (음식, 활동), 출근시간, 퇴근시간, 평균 수면 시간, 하루 활동량

나. 건강관리 에이전트는 입력된 사항과 이전에 다이어트를 한 자료가 있으면 두 자료를 참조하여 사용자에게 적합한 다이어트 방법을 제시하고 건강관리 에이전트가 관여할 수 있는 범위를 결정한다.

다. 사용자는 제시된 방법 중에 가장 자신에게 적합한 것(default로 운동과 음식조절은 함께 선택)을 선택하여 다이어트 시작 날짜를 입력하고 그 날짜부터 건강관리 에이전트는 관리를 시작

한다.

라. 건강관리 에이전트는 매일 체내지방수치와 몸무게를 사용자에게 일정한 시간에 측정할 것을 요구하여 다이어트 상황을 체크한다.

마. 건강관리 에이전트는 매일 하루에 먹은 음식의 종류, 양, 운동정도를 사용자에게 묻는다.

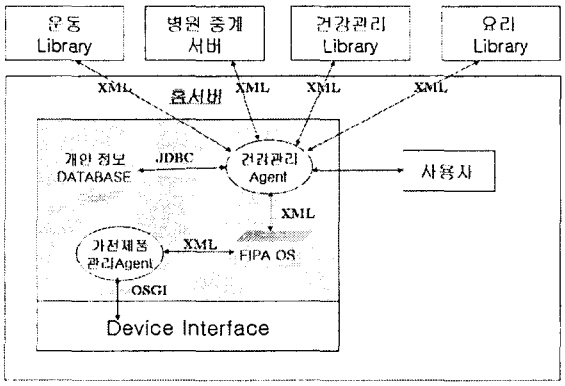
바. 며칠간의 정보로 진행상태가 미진할 경우에 아래와 같은 강도 높은 관여를 실시한다.
- 러닝머신을 정해진 거리만큼 뛰지 않을 때는 사용자의 취미 생활을 제한하기, 음식 소비가 다이어트 요구량보다 큰 경우 음식소비액 동결, 냉장고 문을 열 때마다 경고음 울리기, TV시청 시간이 너무 길 경우 TV화면에 경고 문장 표시.

사. 현재의 몸 상태를 혈압계, 온도계, 혈당측정기, 맥박계 등의 기본 건강도구를 사용하여 측정해 계속 다이어트를 할 것인지 여부를 사용자에게 물어본다. 만약 몸 상태가 안 좋을 시에는 다이어트를 중단시키고, 병원에 진료요청을 한다.

아. 다이어트를 성공했을 경우는 현재 상태를 유지하기 위한 방법을 제시하고 정기적으로 체중, 체내지방수치, 혈압 등을 측정하게 하여 이런 수치가 비만에 가까워지면 다이어트를 다시 권유 한다.

3.2 시스템 구성

<그림 3-1> 시스템 구성도



건강관리 에이전트는 사용자로부터 건강관리에 대한 요청을 처리하고, 중요한 사항이 있을 때 마다 사용자에게 정보를 제공하는 역할을 한다. 또한 홈 서버에 있는 가전제품관리 에이전트와 사용자들의 모든 정보를 저장한 데이터베이스를 직접 관리한다. 데이터베이스를 따로 관리하는 에이전트가 있으면 편리하지만 이 논문에서는 건강관리 에이전트가 직접 제어하도록 한다. 건강관리 에이전트는 병원중계 서버, 건강관리 Library, 요리 Library, 운동 Library와 메시지를 주고받음으로써 필요한 정보를 사용자에게 알려 준다.

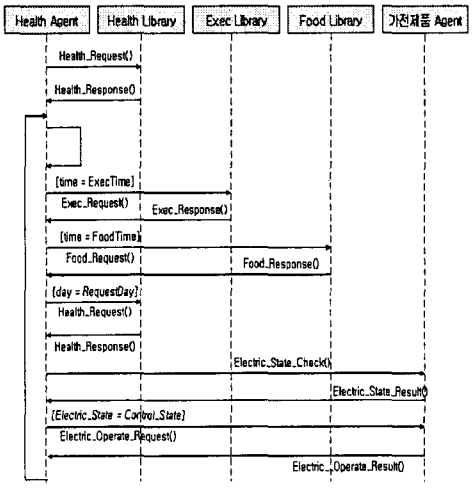
가전제품관리 에이전트는 가전제품을 직접 관리하는 에이전트로 OSGI를 이용하여 디바이스 영역을 관리 하게 된다. 가전제품관리 에이전트는 항상 집안의 가전제품의 상태에 대한 정보와 누적된 정보를 보유하고 다른 에이전트나 사용자에게서 오는 요청을 처리한다. 예를 들어, 가전제품의 ON/OFF, 현재 가전제품 상태, 누적된 정보, 각 가전제품만의 특성화된 동작에 대한 요청이 있을 수 있다. 내부적으로는 가전제품의 누적된 전기료를 계산하여 알려 줄 수 있고, 전기료를 최대한 적게 사용하도록 가전제품을 관리하는 역할을 할 수도 있다.

에이전트들과 각 Library와의 통신에서 사용하는 FIPA에서 정의한 ACL 메시지의 형태는 아래와 같다.

```
(request
  :sender
    (agent-identifier :name dummy@foo.com
      :addresses (sequence iiop://foo.com/acc))
  :receiver (set (agent-identifier :name ams@foo.com
      :addresses (sequence iiop://foo.com/acc)))
  :language FIPA-SL0
  :protocol FIPA-Request
  :ontology FIPA-Agent-Management
  :content (action (agent-identifier
          :name ams@foo.com
          :addresses (sequence iiop://foo.com/acc))
      (register (ams-agent-description
          :name (agent-identifier
              :name dummy@foo.com
              :addresses
                  (sequence iiop://foo.com/acc))
          :state active))))
```

FIPA OS는 기본적으로 ACL 메시지를 처리하는 API를 제공하기 때문에 위의 예제와 같은 text문서를 만들지 않아도 된다. 본 논문에서는 ACL 메시지의 content 부분을 text로 만들지 않고 XML문서를 파싱한 DOM 객체로 content에 실어서 메시지를 보내게 된다. 메시지를 받은 쪽에선 파싱된 DOM 객체를 처리하기 때문에 XML 문서를 다시 파싱 할 필요가 없다.

3.3 프로그램의 순차적 흐름



건강관리 에이전트는 사용자의 요청에 따라 건강관리 Library로 Health_Request() 메시지를 보내고 건강관리 Library는 받은 메시지를 분석하여 사용자에 적절한 관리 방식을 Health_Response()메시지에 실어 보낸다. 또한, 사용자가 운동해야 할 시간이나 식사 시간에 맞춰 운동 Library, 요리 Library에게 메시지를 보내어 운동 정보와 요리 정보를 받고 특정 날짜가 경과 하면 사용자의 변화된 정보를 건강관리 Library에게 보내어 Health_Response() 메시지를 받는다.
가전제품관리 에이전트는 건강관리 에이전트의 요청에 따라 현재 가전제품의 상태를 알려주고 건강관리 에이전트가 사용자의 건강 상태에 따라 가전기기를 제어할 필요가 있다고 판단을 하면 이를 가전제품관리 에이전트에게 알려주고 가전제품관리 에이전트는 가전기기를 제어하게 된다.

4. 각 모듈별 상세 기능과 동작

4.1 건강관리 에이전트

4.1.1 기능

건강관리 에이전트는 본 논문에서 구현하고자 하는 중심 에이전트로서 인터페이스를 통해 사용자와 1:1 대화를 한다. FIPA Specification을 만족하는 플랫폼이 포함된 홈 서버 디스플레이, PDA, 휴대폰, TV, 냉장고, 모니터 등 다양한 매체를 통해 사용자로부터 요구사항을 입력받고, 필요한 정보를 알려 준다. 현재 홈 서버와 PDA에서의 개발은 이루어졌지만, 다른 매체에서의 실험은 이루어지지 않았다.

사용자가 건강관리 에이전트에게 관리하고 싶은 건강을 요청 하면, 건강관리 Library에서는 사용자의 조건에 맞게 건강 조절 방식을 추천한다. 건강관리 에이전트는 이 정보를 바탕으로 건강관리 Library, 요리 Library, 운동 Library에게 필요한 정보를 받고, 현재 사용자의 누적된 정보를 분석하여 사용자의 건강을 관리한다. 만약 사용자의 행동을 조절할 필요가 있다면 가전제품관리 에이전트에게 메시지를 전달하여 가전기기들을 제어를 한다. 건강관리 Library에게 전달하는 메시지를 통해 추천 운동을 요청하는 시간, 추천 음식을 요청하는 시간, 몸무게를 측정하는 시간, 금일의 음식 섭취량과 운동량을 기입하는 시간, 기타 설정을 하는 시간을 사용자의 생활(출근시간, 퇴근시간, 잠자는 시간 등)에 맞추어 설정하게 된다.

건강관리 에이전트는 기본적인 의료지식을 가지고 있다. 가정에서 쉽게 측정할 수 있는 체온, 몸무게, 혈압 등의 수치의 변화를 감지해서 몸의 이상유무를 판단한다. 갑작스런 몸의 변화가 있을 때는 가까운 병원이나 등록된 원격진료 병원에 상태를 알려주고, 예약을 한다.

건강관리 에이전트에 필요한 사용자의 다양한 정보는 홈 서버에 내장된 데이터베이스를 이용한다. 데이터베이스에는 기본적인 사용자 정보, 기타 Library와 연동하면서 발생하는 설정 정보, 서비스 정보가 저장된다. 건강관리 에이전트는 사용자의 매일 변화되는 몸무게, 체지방수치, 체온을 데이터베이스에 저장하고, 이 정보를 종합 분석함으로써 사용자의 건강을 판단하는 기본 자료가 된다.

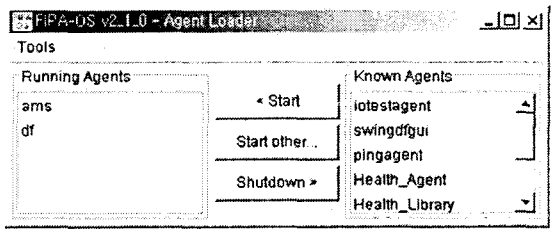
건강관리 에이전트는 FIPA OS 플랫폼 위에서 동작하는 에이전트이다. FIPA OS는 현재 Java 언어로 구현되었기 때문에, Java로 제작하였다. 현재 데이터베이스는 mysql을 사용하였고, JDBC를 통해 데이터를 관리한다. 가전제품 관리 에이전트와 기타 Library와는 ACL 메시지를 통해 통신한다.

4.1.2 동작

건강관리 에이전트는 Fipa os v2.1을 사용하여 에이전트 틀을 만들었다. Fipa os의 profile 폴더의 loader.profile 파일 안에 아래의 내용을 추가하여 에이전트의 클래스를 Fipa os에 등록한다.

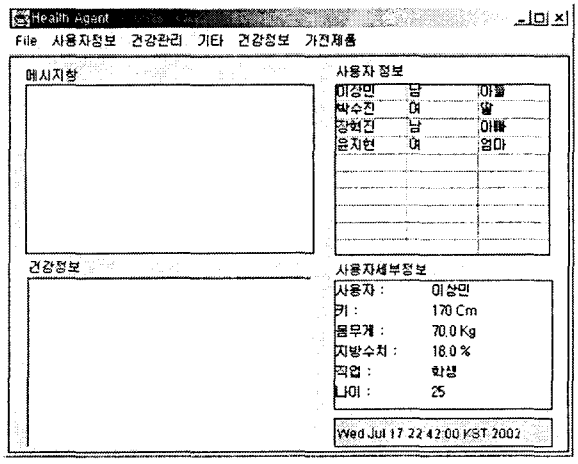
```
<agent Description className =
"fipaos.ha.HealthAgent.HealthAgent"
agentName="Health_Agent" owner="fipaos"
start="false" />
```

<그림 4-1> Fipa os 실행



Fipa os를 실행하면 Known agents에 Health_Agent 가 보이고 Health_Agent를 클릭하고 start버튼을 누르면 건강관리 에이전트가 실행된다.

<그림 4-2> 건강관리 에이전트 실행

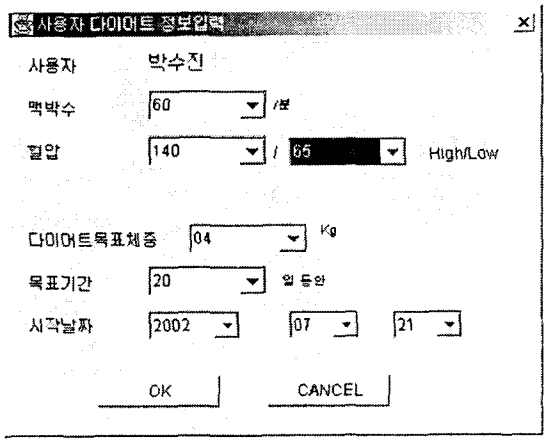


위의 창에서 사용자 정보는 홈 서버의 Database에 저장된 내용이다. 사용된 사용자 정보 Database Table 은 아래와 같다.

```
CREATE TABLE user_information
(Number INT PRIMARY KEY not null
auto_increment ,
U_Name VARCHAR(255),
U_Birthday DATE,
U_Sex TINYINT,
U_Height TINYINT UNSIGNED,
U_Weight FLOAT,
U_Fat_Rate FLOAT,
U_Part VARCHAR(20),
U_Job VARCHAR(30));
```

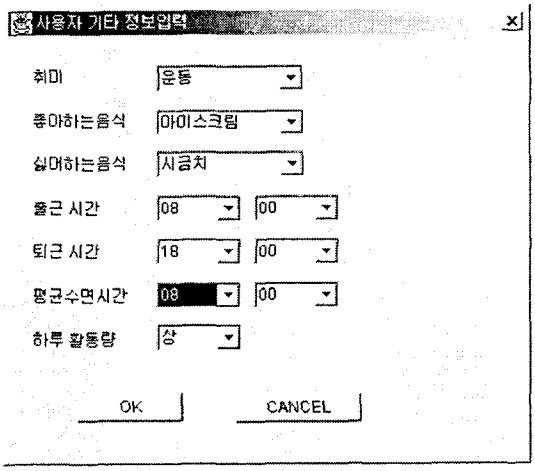
건강관리 메뉴에서 다이어트를 실행하면 아래의 창이 뜨고, 사용자가 다이어트에 필요한 정보를 묻게 된다.

<그림 4-3> 사용자 다이어트 정보 입력



정보를 입력하고 OK 버튼을 누르면 아래의 창이 뜬다.

<그림 4-4> 사용자 다이어트 기타 정보 입력



위의 두개의 창에서 입력된 정보로 건강관리 Library에 다이어트 요청을 하게 된다. 또한 이 정보는 Database에 저장된다. Database Table 구조는 아래와 같다.

```
CREATE TABLE user_health_information
(Number INT PRIMARY KEY not  null
auto_increment,
U_Name VARCHAR(255),
Request_Health VARCHAR(50),
U_Pulse INT,
U_High_Blood_Pressure INT,
U_Low_Blood_Pressure INT,
U_Diet_Goal INT,
U_Diet_Term INT,
U_Start_Day DATE,
U_Start_Weight FLOAT,
U_Start_Fat_Rate FLOAT,
U_Interest VARCHAR(50),
U_Like VARCHAR(50),
U_DisLike VARCHAR(50),
U_Office_Attendance_Hour INT,
U_Office_Attendance_Min INT,
U_Office_Leaving_hour INT,
U_Office_Leaving_Min INT,
U_Sleeping_Term_Hour INT,
U_Sleeping_Term_Min INT,
U_Activity_Rate VARCHAR(10),
Is_Complete INT);
```

위의 정보들을 이용하여 건강관리 Library 에게 다이어트 요청을 하면 요청의 결과로 사용자의 제한 몸무게를 설정하게 된다. 사용자는 매일 정해진 시간에 체중계를 이용하여 몸무게와 지방수치를 측정해야 한다. 그리고 밤에는 금일 식사량과 활동량을 입력하게 된다. 금일 식사량과 활동량은 정확한 수치가 아니라 상, 중, 하로、나누어진 대강의 정보를 입력한다. 몸무게와 활동량은 아래의 Database Table의 형태로 이루어져 있다.

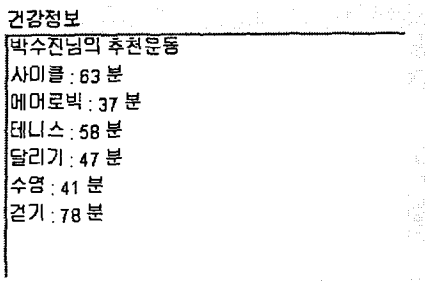
```
// 사용자의 매일의 몸무게와 fat_rate를 저장
CREATE TABLE user_data_diet
(Number INT PRIMARY KEY not null
auto_increment,
U_Name VARCHAR(255),
U_Today DATE, today_weight FLOAT,
today_fat_rate FLOAT);

// 사용자의 매일의 식사량과 운동량을 저장
CREATE TABLE user_foodexec_data
(Number INT PRIMARY KEY not null
 auto_increment,
U_Name VARCHAR(255),
U_Today DATE,
today_break INT,
today_lunch INT,
today_dinner INT,
today_exec INT);
```

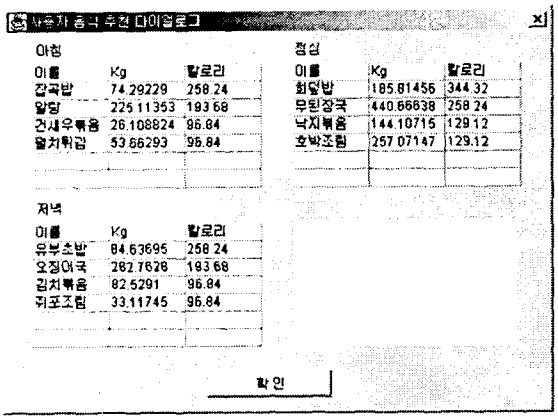
매일 아침에 사용자는 식단과 운동정보를 음식

관리 Library와 운동 관리 Library에게 요청을 하고, 이 정보들을 매일 받게 된다. 받은 정보들은 아래의 형태로 보이게 된다.

<그림 4-5> 사용자 건강정보



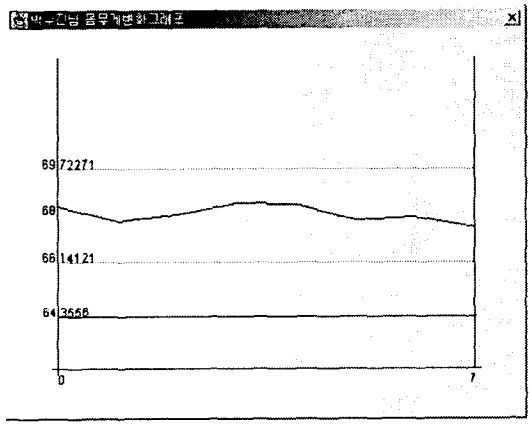
<그림 4-6> 사용자 음식 추천



위의 정보들도 Database에 저장한다.

매일 사용자가 체중계로 입력하는 몸무게의 정보를 사용자는 얼마나 변하는지 잘 알 수 없기 때문에 도표를 통해 살펴 볼 수 있게 했다. 도표는 아래의 그림과 같다.

<그림 4-7> 몸무게 변화 그래프



건강관리 에이전트는 계속 가전제품 관리 에이전트에 메시지를 보내 현재 가전제품들의 상태를 계속 감시한다. 만약 사용자가 건강관리 Library에서 받은 제한 몸무게를 넘어갈 경우 TV, 냉장고, 에어콘등 다양한 가전기기들을 제어하게 된다. 가전기기들을 제어하는 시나리오는 다양한 방식이 있을 수 있고, 현재는 TV와 냉장고로 구현되었다.

4.2 건강관리 Library

4.2.1 기능

건강관리 Library는 모든 건강에 관한 지식을 보유하고 있는 Library이다. 건강관리 에이전트에서 보내주는 사용자의 신체조건과 현재 건강 상태를 의료지식을 참고하여 건강관리에 필요한 정보를 사용자에게 전달한다.

4.2.2 동작

사용자가 다이어트에 대한 정보를 입력하면 그 정보가 아래와 같은 XML형태로 건강관리 library로 보내진다.

```
<health_library>
  <request>다이어트</request>
  <user_information>
    <name>박수진</name>
    <age>24</age>
    <sex>여</sex>
    <height>164</height>
    <weight>60</weight>
  </user_information>
  <diet_information>
    <pulse>NULL</pulse>
    <high_blood_pressure>NULL</high_blood_pressure>
    <low_blood_pressure>NULL</low_blood_pressure>
    <fat_rate>NULL</fat_rate>
    <diet_goal>5</diet_goal>
    <diet_goal_term>30</diet_goal_term>
    <start_day>2002-08-01</start_day>
  </diet_information>
  <etc_information>
    <interest>NULL</interest>
    <like>NULL</like>
    <dislike>NULL</dislike>
    <office_attendance>08:00</office_attendance>
    <office_leaving>20:00</office_leaving>
    <sleeping_term>10:00</sleeping_term>
    <activity_rate>상</activity_rate>
  </etc_information>
  <diet_etc_information>
    <diet_part>NULL</diet_part>
    <diet_rate>NULL</diet_rate>
  </diet_etc_information>
</health_library>
```

사용자가 건강관리 에이전트에서 다이어트에 대한 정보를 입력하면 XML 형태로 건강관리 Library로 보내진다. XML에 저장되어 있는 정보를 분석하여 하루 동안 섭취해야 하는 칼로리와 운동으로 소비해야 하는 칼로리를 계산하고, 특정 날짜

(본 논문에서는 7일) 동안 다이어트를 했을 때 건강관리 에이전트에서 컨트롤해야 하는 최대 몸무게와 최소 몸무게를 계산하여 사용자에게 보내준다. 지정된 날짜가 되면 건강관리 에이전트로부터 받은 새로운 request 메시지의 내용으로 칼로리와 몸무게를 계산하여 보내준다.

하루 동안 섭취해야 하는 칼로리는 기초 대사량, 활동대사량, 식품 이용을 위한 에너지를 더한 값에서, 비만인의 경우 기본으로 섭취해야 하는 칼로리에 500~1300kcal 적게 섭취해야 하므로, 중간값인 800kcal 뺀 값으로 한다.(이원재, 1997)

운동으로 소비해야 하는 칼로리는 사용자가 목표 기간 동안 몸무게를 감소시키려 할 때 필요한 소비 칼로리에서 음식 조절로 소비할 수 있는 칼로리를 뺀 나머지로 한다. 이 칼로리는 운동 Library에서 운동 종류별로 운동 시간을 계산 하는데 사용된다.

최대 몸무게는 다이어트 전의 몸무게에서 0.1% 추가한 값이고, 최소 몸무게는 계획대로 다이어트를 했을 때의 몸무게에서 원래 몸무게의 0.1% 감소한 몸무게이다. 만약 다이어트 후 사용자의 몸무게가 최대 몸무게 보다 크면 계획대로 다이어트를 하지 않은 것으로 간주하고, 최소 몸무게보다 작으면 무리한 다이어트로 간주한다. 이것으로 사용자가 다이어트를 제대로 실행하고 있는지 확인할 수 있다.

보낼 정보의 XML형식은 아래와 같다.

```
<exercise_library>
 <respond>운동</respond>
 <user_information>
  <name>박수진</name>
  </user_information>
 <health_information>
  <h_calorie>550</h_calorie>
 </health_information>
 <food_information>
  <f_calorie>2000</f_calorie>
 </food_information>
 <diet_control_information>
  <control_high_weight>61</control_high_weight>
  <control_low_weight>58</control_low_weight>
  <re_request_day>7</re_request_day>
 </diet_control_information>
</exercise_library>
```

4.3 요리 Library

4.3.1 기능

요리 Library는 요리에 관한 지식을 보유하고 있는 Library이다. 사용자가 원하는 주 재료를 바탕으로 다양한 방법의 요리법을 제시한다. 실제 건강과 음식사이에는 많은 관련이 있으므로 Library는 건강식 위주로 구성된다. 또한 요리 Library는 계절과 재료의 가격 변동, 그리고 건강과 관련해서 분류하며, 사용자의 다양한 요구에 맞는 요리법을 제시한다.

4.3.2 동작

사용자가 다이어트를 시작하게 되면 식단을 건강관리 Library에서 관리 하는데, 사용자가 입력한 정보와 건강관리 Library에서 계산한 하루 동안 섭취해야 하는 칼로리를 이용하여 식단을 구성한다. 이 칼로리는 아침, 점심, 저녁을 3:4:3의 비율로 나누어 섭취 하는 것으로 가정한다. 식단은 어떤 종류의 밥을 하루에 몇 g을 섭취해야 하는지, 밥을 섭취하였을 때 칼로리는 얼마인지를 계산하여 건강관리 에이전트로 보낸다. 식단을 만들 때 필요한 음식 정보는 음식의 칼로리량, 음식량, 영양성분에 대한 내용이 아래와 같은 형식으로 데이터베이스에 저장되어 있다.(식품분석표)

```
Number INT Category INT,
Code SMALLINT,
Weight INT,
NameKor VARCHAR(255),
NameEng VARCHAR(255) ,
Energy FLOAT,
Albumen FLOAT,
Fat FLOAT,
Carbohydrate FLOAT,
VitaminA FLOAT,
VitaminB1 FLOAT,
VitaminB2 FLOAT,
VitaminC FLOAT,
Calcium FLOAT ,
Phosphorus FLOAT,
Iron FLOAT,
Natrium FLOAT,
Potassium FLOAT,
Cholesterol FLOAT,
Fiber FLOAT
```

각 아이템은 random하게 선택하고 정해진 칼로리에 맞게 음식의 양을 조절한다.

Food library로 보내는 메시지는 아래와 같다.

```
<food_library>
 <request>식단</request>
 <user_information>
  <name>박수진</name>
 </user_information>
 <calorie_information>
  <calorie>1500</calorie>
 </calorie_information>
</food_library>
```

food library에서 보내는 메시지는 아래와 같다. 여기서 time은 아침식사는 0, 점심은 1, 저녁은 2로 정해놓았다.

```
<food_library>
 <respond>식단</respond>
 <user_information>
   <name>박수진</name>
  <date>2002-07-15</date>
 </user_information>
 <food_information>
  <name>쌀밥</name>
  <cal>313.200012</cal>
  <quan>100</quan>
  <time>0</time>
  <name></name>
  <cal></cal>
  <quan></quan>
  <time>1</time>
 </food_information>
</food_library>
```

4.4 운동 Library

4.4.1 기능

운동 Library는 운동에 관한 지식을 보유하고 있는 Library이다. 건강관리 Library에서 계산한 하루 동안 운동으로 감소시켜야 하는 칼로리에 따라 특정한 운동을 얼마나 해야 하는지를 계산한다. 사용자는 운동 Library가 추천하는 여러 가지 운동 중에 하나를 선택하여 하루에 정해진 시간만큼 할 수 있다. 또한 스포츠 센터에서만 할 수 있는 운동은 사용자에게 특정 스포츠 센터를 추천해줌으로써 가능해진다.

4.4.2 동작

운동의 종류는 사이클, 에어로빅, 테니스, 달리기, 수영, 걷기로 여섯 가지를 제시하였고, 각각 하루에 몇 분씩 운동을 하면 주어진 칼로리가 소비되는지를 계산하여 사용자에게 추천한다.

각 운동에 대한 운동 시간을 계산 하는 식은 아래와 같다.

운동시간 = 하루에 운동으로 소비해야 하는 칼로리
/운동했을 때 1분동안 소비되는 칼로리
***사용자 몸무게**

exec_library로 보내는 메시지는 아래와 같다.

```
<exercise_library>
 <request>운동</request>
 <user_information>
  <name>박수진</name>
  <weight>60</weight>
 </user_information>
 <exec_information>
  <h_calorie>283.33334</h_calorie>
 </exec_information>
</exercise_library>
```

운동시간을 계산하여 건강관리 에이전트로 보내는 메시지는 아래와 같다.

```
<exercise_library>
 <respond>운동</respond>
 <user_information>
  <name>박수진</name>
 </user_information>
 <exec_information>
  <exec_sort>
   <name>사이클</name>
   <minutes>30</minutes>
  </exec_sort>
  <exec_sort>
   <name>에어로빅</name>
   <minutes>40</minutes>
  </exec_sort>
  <exec_sort>
   <name>테니스</name>
   <minutes>30</minutes>
  </exec_sort>
  <exec_sort>
   <name>달리기</name>
   <minutes>40</minutes>
  </exec_sort>
  <exec_sort>
   <name>수영</name>
   <minutes>40</minutes>
  </exec_sort>
  <exec_sort>
   <name>걷기</name>
   <minutes>60</minutes>
  </exec_sort>
 </exec_information>
</exercise_library>
```

5. 건강관리 멀티 에이전트 시스템의 개별화

5.1 웹서비스

웹서비스는 Service-Oriented Architecture 기반으로 하고 있다. Service-Oriented Architecture란 분산 어플리케이션을 구현할 때 쓰는 방식의 하나로서, 다이나믹한 분산 어플리케이션 구현에 사용된다. 이 구조는 주요한 역할을 하는 Service Provider, Service Consumer, Service Broker로 구성되었다. Service Provider는 웹서비스를 제공해 주는 것이고 Service Consumer는 HTTP를 이용하여 통신할 수 있는 Client를 의미 한다. Service Broker는 사용가능한 웹서비스를 등록해 놓은 레지스트리이다.

이러한 구조는 Fipa Specification 의 Service Agent, DF, Consumer Agent의 형태와 유사하다. 또한 웹서비스에서, Web Service Provider, UDDI,

Web Service Consumer의 형태와 동일하다. Fipa Specification과 웹 서비스와의 차이점이라면 Service Broker의 위치이다. Fipa Specification은 DF가 각각의 플랫폼마다 동작을 하는 것이고, UDDI는 UDDI Business Registry를 다국적 기업의 대기업이 관리한다. 하지만, DF를 확장하던지, UDDI를 세분화 하는 작은 서비스를 생각하면 그 차이는 줄어들 수 있다.

건강관리 에이전트를 웹서비스 환경에서 구현하면 Fipa Specification의 플랫폼에서 벗어날 수 있다. 즉 XML 기반으로 하는 Soap 프로토콜로 구현된 Web Service의 Service Provider와 Service Consumer의 역할을 건강관리 에이전트가 동시에 하게 된다. 사용자가 원하는 서비스를 기존의 ACL 메시지가 아니라, Soap 메시지를 보내면, 서비스 등록이 이루어지고, 원하는 건강관리 서비스가 수행된다. 기타 건강관리에 필요한 Library와의 통신도 ACL 메시지가 아니라 Soap 메시지를 이용한 웹서비스 형태가 된다.

웹서비스 형태로 제작된다는 것은 Fipa Specification과 같은 에이전트 플랫폼에 의존하지 않고도, 멀티에이전트 시스템이 가능하고, 프로그램 언어에도 종속적이지 않게 된다. 하지만, 보완할 점으로는 보안문제가 있다. 만들어진 건강관리 에이전트를 사용할 수 있는 사용자들에 대한 관리가 문제가 되고, UDDI를 이용한 건강관리 Library, 음식관리 Library, 운동관리 Library, 병원중계 서버, GIS 서비스등 다양한 서비스를 이용할 때의 인증이 문제가 된다. 따로 인증만 관리하는 서비스가 필요하고, 현재 .NET Passport가 비슷한 형태의 서비스라고 할 수 있다.

웹서비스로 가능해지는 기술 중에 무선 의료 기술, 병원과의 연계, 비만관리 센터와의 연계 그리고 GIS서비스와의 연계에 대해 아래에서 기술하였다.

5.1.1 무선 의료 기술

무선 의료 기술이란 모바일 기기를 통하여 가능한 의료 기술 서비스를 받는 것을 의미한다. 모바일 기기에서 웹서비스 환경에서 구현된 Service Provider인 건강관리 에이전트에 접속을 하여 에이전트에서 제공하는 서비스를 이용할 수 있다. 모바일 기기는 웹서비스 Consumer의 역할을 하게 되고 이를 가능하게 하는 웹서비스 게이트웨이가 존재하게 된다. 이 게이트웨이를 통하여 건강관리 서비스를 모바일 환경에서도 이용할 수 있다. 현재 모바일 웹서비스는 개발 중에 있다.

5.1.2 병원과 연계

건강관리 Library에서는 서비스와 관련된 정보를 포함하고 있는데 모든 건강에 관한 자료를 포함하기 어려우므로 병원과 연계하여 의사에게 조언을 구할 수 있다. 즉, 사용자가 자신의 질병에 대해 증상을 건강관리 에이전트에게 알려주면 건강관리 에이전트는 이를 의사에게 전달한다. 또한 체중계, 혈압기, 맥박계 등의 의료기기로 측정한 사용자의 상태도 건강관리 에이전트를 통해 의사에게 전달된다. 의사는 사용자의 증상과 측정된 자료를 통하여 진단을 할 수 있다. 이런 병원과의 연계를 통해 사용자는 시간을 들여 병원에 가지 않고서도 실내에서 의사의 진료와 처방을 받을 수 있다.

5.1.2 비만관리 센터와 연계

Service Provider의 역할을 하는 비만관리 센터와 연계가 되면 건강관리 Library, 요리 Library, 운동 Library는 건강관련 정보, 사용자에게 맞는 식단, 운동의 종류와 시간 등의 각 Library의 역할에 맞는 다양한 정보를 비만관리 센터로부터 받아 사용자에게 제공할 수 있다. 이때 각 Library는 Service Consumer의 역할을 하게 되고 비만관리 센터로부터 새로운 자료를 정기적으로 받아 사용자에게 제공할 수 있다.

5.1.3 GIS 서비스와 연계

건강관리 에이전트는 또 다른 Service Provider인 GIS 서비스와 연계하여 사용자에게 필요한 정보를 제공할 수 있다. 사용자가 병원이나 약국을 가야 한다면 건강관리 에이전트는 GIS 서비스에게 요청을 하고 GIS 서비스는 약국과 병원의 위치를 건강관리 에이전트에게 제공해준다. 즉, 사용자는 건강과 관련된 GIS 서비스를 건강관리 에이전트에서 바로 이용할 수 있게 된다.

6. 결 론

본 논문에서는 정보가전을 이용하여 홈 서버에서 동작하는 건강관리 멀티에이전트 시스템의 설계 및 구현 방법을 기술하였다. 정보가전이 현실화 되었을 때, 가전기기를 효율적으로 동작시키는 방법으로 일반적인 어플리케이션을 만든다면, 현재 컴퓨터의 프로그램들처럼 제작한 회사가 다르면 서로 정보를 공유할 수 없기 때문에 많은 자원 낭비를 초래한다. FIPA 표준으로 에이전트를 만들면 장소나 종류와 상관없이 ACL 메시지만 안다면 에이전트 사이에 대화를 통해 고차원적인 동작이 가능해진다. 건강관리 멀티에이전트 시스템은 이러한 생각을 바탕으로 제작되었고, 가전제품 관리 에이전트, GIS 에이전트, 기타 Library와 연계하여 동작한다.

FIPA OS와 OSGI가 모두 자바로 만들어졌기 때문에, 가전기기 제어는 Hardware Realtime을 만족해야 하는데, FIPA OS에서 동작하는 에이전트 사이의 메시지 전송에서 시간이 오래 걸린다. 이 문제를 해결하기 위해 FIPA OS를 .NET 환경에서 동작할 수 있도록 개발하고 또한 웹 서비스와의 연동이나 병원, 의사와의 원활한 통신을 통한 지속적인 진료를 위해 무선 의료 정보 기술과의 결합도 고려하여 발전시킬 계획이다.

참고 문헌

Huhns, Micheal N., Munidar P. Singh, Readings In Agent, Morgan KauFmann Publishers Inc.,

1998, pp.1-24.
Open Service Gateway Initiative,
http://www.osgi.org/
The Foundation for Intelligent Physical Agents, FIPA2000 specification, http://www.fipa.org/
손영성 외, LonWorks 네트워크를 이용한 원격 홈 오토메이션 시스템, 정보처리 제 8권 제 1호, 2001.
식품분석표,
http://www-2.knu.ac.kr/~fsnu/foodexp/main.htm
이원재, 과학적인 체중 조절법, 중외출판사, 1997.
임동찬 외, OSGi Service Gateway Specification 1.0에 기반을 둔 홈오토메이션 컨트롤러용 개방형 서비스 게이트웨이의 구현, 정보과학회 2001년 추계학술대회, 2001.
전호인, 송동일, 정보가전을 위한 IEEE394 기술의 적용방안, 정보처리 제8권 제1호, 2001.