

MAMI: 의료 정보 제공을 위한 멀티 에이전트 시스템에서의 에이전트 플랫폼

(MAMI: Agent Platform in a Multi-Agent System Providing Medical Information)

최 원 기 * 김 일 곤 **

(Won-ki Choi) (Il Kon Kim)

요 약 본 논문에서는 지능형 의료 정보 제공을 위한 멀티 에이전트 시스템인 MAMI(Multi-Agent system for Medical Image)의 에이전트 플랫폼의 설계 및 구현에 대해 기술한다. MAMI는 여러 의료 정보 중 의료 영상을 위한 멀티 에이전트 시스템이다. MAMI의 가장 중요한 구성 요소는 에이전트 플랫폼이고, 각 에이전트들이 동작할 수 있는 물리적 기반을 제공해 준다. MAMI는 FIPA(Foundation for Intelligent Physical Agent)가 제안하는 관리 모델을 따르고 있다. MAMI에서는 COM(Common Object Model)과 XML(eXtensible Markup Language)로 인코딩된 ACL(Agent Communication Language)을 메시지 교환에 이용하고 있다. 의료 인력을 하나의 에이전트로 개념화하여 시스템에 연결함으로써 소프트웨어 에이전트와 휴먼 에이전트가 지식을 공유하기에 적합한 물리적 기반을 제공해 주며 지능형 의료 정보 서비스를 용이하게 해준다.

Abstract This paper describe design and implementation of a medical multi-agent system platform called MAMI (Multi-Agent system for Medical Image), which provides intelligent medical information services. The most important component of MAMI is a medical multi-agent system platform that supports a physical environment that medical agents can be deployed. MAMI follows FIPA (Foundation for Intelligent Physical Agent)'s agent management reference model. In MAMI, COM (Common Object Model) and XML (eXtensible Markup Language) for encoding ACL (Agent Communication Language) are used for multi-agent communications. In MAMI, a medical staff is conceptualized as an agent and integrated with multi-agent systems. MAMI agent platform provides an infrastructure applicable to share necessary knowledge between human agents and software agents. So MAMI makes intelligent medical information services easier.

1. 서 론

의료 정보 시스템은 고도의 전문 지식을 가진 전문 의료 인력과의 연계가 필요하고, 진료 행위 역시 복잡하기 때문에 이들을 연결해 주는 자치 시스템을 필요로 한다. 멀티 에이전트 시스템은 자치성을 가진 에이전트들이 모여 서로 지식을 교환하며 자신의 목표를 추구하는 시스템이다[1]. 멀티 에이전트 시스템은 동적이고 고

도의 자치성을 요구하는 영역, 본 논문에서 적용한 의료 영역에 아주 적합하다[2].

본 논문에서는 의료 영역 중 현재 활발하게 전산화가 진행중인 의료 영상 영역에서 지능형 의료 정보 서비스를 제공하기 위해서 개발된 멀티 에이전트 시스템 플랫폼, MAMI에 대해 기술한다. 멀티 에이전트 시스템 플랫폼은 에이전트들이 활동할 수 있는 물리적 기반을 제공해주며, 에이전트들이 사용하는 언어, 통신 방법, 에이전트들 간의 관계를 결정짓는다. MAMI 플랫폼은 FIPA 규격을 근간으로 구성되어져 있으며, 의료 영역에서는 필수적인 전문 의료 인력과의 연계를 지원한다.

본 논문의 구성은 다음과 같다. 2절에서는 MAMI 플랫폼의 근간을 이루는 FIPA 규격의 특징을 기술하고, 3

* 비 회 원 : 모빌 써앤써 연구소 연구원
choiwk@mcnc.co.kr

** 정 회 원 : 경북대학교 컴퓨터학과 교수
ilkim@knu.ac.kr

논문접수 : 2001년 2월 21일

신사완료 : 2001년 7월 4일

절에서는 실제 MAMI 플랫폼이 어떻게 구성되어 있는 지에 대해 기술한다. 4절에서는 XML로 인코딩 된 ACL을 MAMI에서 어떻게 활용할 수 있는지를 기술하고, 5절에서는 의료 전문 인력과 연계 방법의 하나의 WAP 적용 방법을 기술하고, 평가와 더불어 다른 플랫폼과의 비교를 한다. 마지막으로 6절에서는 MAMI 플랫폼의 특성에 대해 논하고 향후 과제에 대해 기술한다.

2. 멀티 에이전트 플랫폼 표준

본 절에서는 멀티 에이전트 시스템 플랫폼 표준인 FIPA가 가지는 특징에 대하여 기술하고자 한다. FIPA에서는 FIPA97, FIPA98, FIPA2000을 비롯해 꾸준히 표준화 작업을 진행하고있다. 그림1은 FIPA의 규격을 관련 영역들로 분류한 것이다.

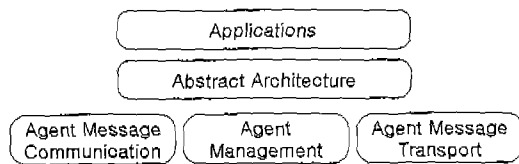


그림 1 FIPA Specification Domains

2.1 에이전트 관리 참조 모델

FIPA의 에이전트 관리 참조모델은 그림2에서 보듯이 에이전트 플랫폼을 두고 에이전트들은 플랫폼의 MTS(Message Transport System)을 통해 메시지를 교환한다. 에이전트 플랫폼은 에이전트들이 동작할 수 있는 물리적 기반을 제공해준다. 에이전트 플랫폼에는 AMS(Agent Management System), DF(Directory Facilitator) 와 같이 관리를 위한 에이전트들이 존재한다. AMS는 에이전트 관리를 담당하고, DF는 옐로우 페이지(yellow page) 역할을 담당하고 있다. MTS는 플랫폼내의 에이전트들만이 아니라 다른 플랫폼에 있는 에이전트간의 메시지 전달을 담당하고 있다. 에이전트 플랫폼간의 통신은 IIOP(Internet Inter Orb Protocol)을 이용하도록 규정되어 있다.

AID(Agent ID)를 가진 에이전트들을 위해 MTS는 메시지의 AID를 보고 해당 에이전트에게 라우팅한다. AID는 에이전트 이름을 가지며 추가적으로 전송 주소, 이름 resolution 서비스 등의 패러미터가 존재한다. 에이전트 이름은 일반적으로 에이전트의 실제 이름과 홈 에이전트 플랫폼 주소의 쌍을 '@'으로 연결해 사용한다. 전송 주소는 에이전트와 접촉할 수 있는 물리적 주소이

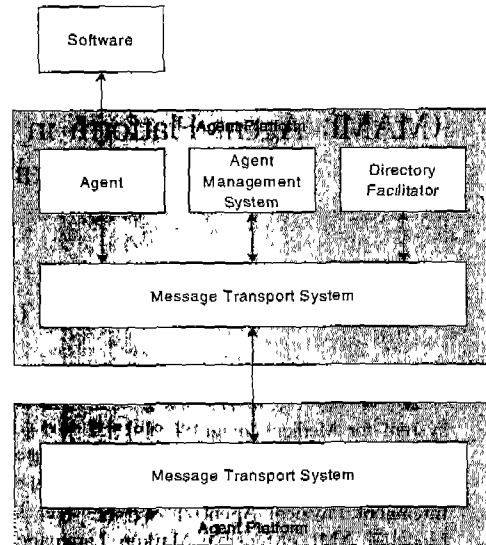


그림 2 Agent Management Reference Model

며 여러 종류의 전송 주소가 있을 경우 이들을 모두 기술해 줄 수 있다. 홈 에이전트 플랫폼은 에이전트가 등록되어 있는 에이전트 플랫폼을 의미한다. 에이전트가 플랫폼의 서비스를 제공받기 위해서는 반드시 자신을 플랫폼에 등록해야 한다. 등록은 AMS를 통해 이루어지는데, 등록 시에는 자신의 이름을 비롯한 프로필을 보내주어야 한다. 등록이 된 후에는 다른 에이전트와 메시지를 주고 받을 수 있으며 DF를 통해 자신을 광고할 수도 있다. FIPA는 관리와 관련된 온톨로지(Ontology)와 AMS/DF가 제공해야 하는 기능들을 정의하고 있다.

2.2 에이전트 통신 언어

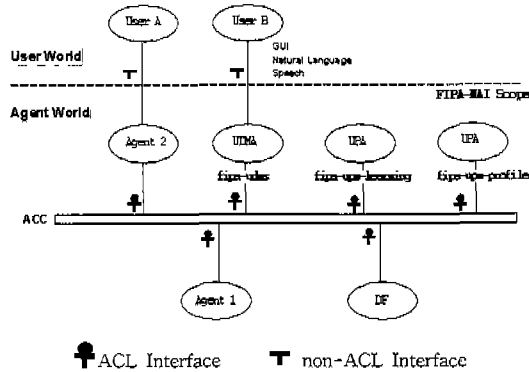
에이전트 통신 언어, ACL(Agent Communication Language)은 에이전트간의 통신에 이용되는 언어이다. FIPA에서는 메시지 계층(Message Layer)과 내용 계층(Content Layer)으로 나누어 내용 전달에 사용되는 언어는 에이전트 임의로 사용하고, 전달에 이용되는 계층만 따르면 메시지가 전달될 수 있게 해준다. 메시지 계층은 KQML(Knowledge Query Manipulation Language)과 비슷한 형태이다. KQML의 Performative와 같은 역할을 하는 Communicative Act를 가지고 있으며 메시지 전송에 필요한 파라미터들을 가지고 있다. 표1과 표2는 FIPA ACL의 CA와 파라미터들의 의미를 나타내고 있다. FIPA2000에는 내용 계층에 대한 규격과 프로토콜

1) FIPA97 Agent Communication Language Specification을 기준으로 작성

에 대한 규격이 분리되어 나타나 있다. 이외에도 메시지를 XML로 인코딩하는 DTD(Document Type Definition) 역시 제시되었다[4].

2.3 휴먼-에이전트 상호 작용

FIPA에서는 사람과 에이전트간의 상호 작용에 대한 규격을 제시하고 있다[5]. 규격에서는 그림 3과 같은 휴먼-에이전트 상호 작용 모델을 제안하고 있다. 이 모델에서는 사용자(휴먼)가 UDMS(User Dialog Management Service)를 제공하는 에이전트(UDMA: User Dialog Management Agent)와 대화를 통해 사용자가 다른 에이전트들과 정보를 교환하는 형태이다. FIPA는 UDMS 외에도 사용자 개인 프로필(User Personalization profile)을 획득하고 관리하는 UPS(User Personalization Service)를 정의하고, 이 두 서비스를 위한 온톨로지, 프로토콜, 사용자와 에이전트 대화에 사용되는 액션들의 의미를 정의하고 있다.



fipa-udms : FIPA user dialogue management service ontology
 fipa-ups-profile : FIPA user personalization profile
 fipa-ups-learning : FIPA user personalization learning ontology

그림 3 Human-Agent Interaction Reference Model

에이전트 입장에서는 사용자는 다른 에이전트와 크게 다르지 않다. 다만 UDMS를 제공하는 서비스를 통해 접촉하는 점만이 다르다. 하지만 에이전트와 사람이 사용하는 언어(자연어)는 매우 달라서 많은 문제점을 가진다. 그래서 자연어와 ACL을 동일한 방법으로 표현하는 언어의 필요성이 대두되었고, 이를 위해서 SKDL (Structured Knowledge Description Language)²⁾를 소개하고 있다.

2) SKDL은 필요한 특성과 간략한 형태만 기술되어 있다.

표 1 FIPA ACL 파라미터들

파라미터	의미
:sender	보내는 에이전트 아이디
:receiver	받는 에이전트 아이디
:content	메시지 내용
:reply-with	메시지에 응답할 때 이 메시지에 대한 것 인지를 나타낼 때 사용된다.
:in-reply-to	이 메시지가 응답하는 이전 메시지를 참조함을 나타낸다.
:envelope	메시지 전송에 관한 정보를 가지고 있다.
:language	Content에 사용되는 언어
:ontology	사용되는 온톨로지
:reply-by	응답이 요구되는 시간
:protocol	메시지에 사용되는 프로토콜
:conversation-id	CA로 이루어지는 진행중인 대화를 구분 하는데 사용된다

표 2 FIPA ACL Communicative Acts

CA	의미
Accept-proposal	이전에 제시된 제안을 받아들인다는 의미
Agree	액션 수행에 동의
Cancel	요구된 액션을 거부
Cfp	제시된 액션을 수행을 위한 제안을 요구
Confirm	제시된 명제가 참임을 알린다
Disconfirm	제시된 명제가 거짓임을 알린다
Failure	액션 수행이 실패했음
Inform	주어진 명제가 참임을 알린다
Not-understood	메시지를 이해할 수 없다는 것을 알린다
Propose	어떤 액션은 제안한다
Query-if	명제가 참인가를 문의
Query-ref	수식에 의해 참조되는 객체를 문의
Refuse	액션 수행을 거부
Reject-proposal	제안을 거부
Request	액션을 요구
Subscribe	참조되는 값의 변화를 알려주기를 요구

3. MAMI 플랫폼과 구현

MAMI 플랫폼은 그림4와 같이 MTS와 ACC, AMS, Delegate Agent, DF로 구성된다. MAMI는 에이전트 플랫폼간의 상호 운용성 제공을 위해 FIPA 규격을 따르지만, 사람과 소프트웨어 에이전트간의 유기적 결합에 중점을 두고 있다.

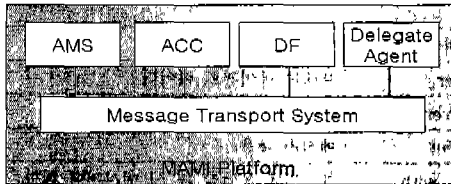


그림 4 MAMI 플랫폼 구성도

3.1 에이전트 플랫폼

플랫폼 에이전트들은 모두 메시지 큐, 자신에게 적합한 파서를 가지고 있으며, 별도의 쓰레드로 동작하고 자신의 상태를 저장할 수 있다. 각 에이전트들은 자신의 임무에 필요한 데이터베이스와 여러 기능들을 개별적으로 가지고 있다. 모든 에이전트는 메시지가 도착하면 그 메시지의 우선 순위를 판단한다. 메시지 우선 순위는 <reply-by> 태그에 있는 시간을 보고 결정하게 된다. 급하지 않은 메시지일 경우 메시지 큐에 삽입되지만, 만약 현재 수행 중인 태스크보다 더 빠른 처리가 필요하다면 현재 자신의 상태를 저장하고 새로이 도착한 메시지를 먼저 처리하게 된다. 태스크가 끝날 경우 처리가 이루어진 태스크가 있는지 조사하고 이를 먼저 처리한 뒤 메시지 큐에 있는 메시지를 처리한다. 이와 같은 처리를 위해서 적절한 위치에 체크 포인트(check pointer)를 두고 있으며 쓰레드 풀(thread pool)을 가지고 있다. 태스크의 중단은 이 체크 포인트에서만 이루어질 수 있으며 쓰레드 풀은 에이전트의 태스크 수행을 위해 생성된 쓰레드를 정해진 수만큼 가지고 있다. 체크 포인트에서는 우선 순위가 높은 태스크가 새로이 나타났는지 체크하고, 없을 경우 수행을 계속하지만, 나타났을 경우에는 자신의 상태를 먼저 저장한 뒤 쓰레드 풀에 아직 할당되지 않은 쓰레드를 찾아서 쓰레드에게 새로이 발생한 태스크를 수행하게 하고 자신은 sleep모드로 들어가게 된다. 만약 모든 쓰레드가 할당되어 있다면 현재 태스크를 먼저 수행하고 끝이 나면 가장 우선 순위가 높은 태스크부터 수행하게 된다. 태스크의 수행이 끝나면 메시지 큐와 수행이 연기된 태스크들의 우선 순위를 비교해 가장 우선 순위가 높은 태스크부터 수행한다. 에이전트에서는 한번에 하나의 태스크만이 수행이 가능하다. 태스크 수행의 필요에 따라 쓰레드를 생성할 수 있지만, 쓰레드 풀의 쓰레드를 이용할 수 없으며 태스크의 수행이 끝나면 반드시 쓰레드를 삭제해야 한다. 이와 같이 기본적으로 필요한 기능들을 MAMI에서는 에이전트 베이스 클래스로 제공해 주고 있다. 플랫폼 에이전트들은 이 베이스 클래스를 이용해 구현되었고, 자신에게 필

요한 기능들을 추가, 구현하고 있다.

플랫폼은 현재 NT 서비스 형태로 구현되어 있고, 서비스를 시작하게 되면 각 플랫폼 에이전트를 초기화한다. 초기화가 끝나면 AMS를 제외한 에이전트들은 자신을 AMS에 등록한다. Delegate 에이전트의 경우 자신이 관리하는 모든 휴먼 에이전트들을 등록한다. 이 등록 과정이 끝나면 플랫폼이 서비스를 제공을 시작한다. 현재 플랫폼이 중지될 경우 수행중인 태스크들은 모두 수행이 중지되고, 제 시작에 수행을 재개하지는 않는다.

3.2 Message Transport System

모든 에이전트들은 MTS를 통해 메시지를 주고 받으며, 전송 프로토콜로 COM을 선택하였다. 모든 에이전트들은 MTS와 연결을 위한 인터페이스와 메시지 전달을 위한 인터페이스를 구현하고 있어야 한다. MTS와 에이전트는 그림 5에 기술된 인터페이스를 각각 제공하고 있다. ConnectionInterface는 MTS가 제공하는 것으로서 최초 연결에 사용되고, ConnectionContext와 자신의 MessageInterface 포인터를 넘겨준다. 일단 연결된 에이전트는 대기 리스트에 들어가게 된다. 대기 리스트에 있는 에이전트는 아직 AMS에 등록되지 않았기 때문에 서비스를 제공받을 수 없다. 따라서 에이전트는 연결 후 AMS에 등록 메시지를 보내게 된다. AMS는 등록 메시지를 받은 후 대기 리스트에 있는 에이전트를 등록 에이전트 리스트로 올린다. 등록 에이전트 리스트에 있는 에이전트들은 메시지를 주고 받을 수 있게 된다. MessageInterface는 MTS, 에이전트 모두가 구현하고 있으며 메시지를 인자로 넘겨주면 된다.

```
//연결을 위한 인터페이스
interface ConnectionInterface : IUnknown
{
    HRESULT connect(
        [in, ref] IUnknown *thisAgent,
        [in, ref] ConnectionContext *context,
        [out, retval] MessageInterface **pMsgInt);
}
//메시지 전송을 위한 인터페이스
//에이전트, 플랫폼 둘 다 가짐
interface MessageInterface : IUnknown
{
    HRESULT sendMessage([in] BSTR m);
}
```

그림 5 MAMI Platform COM Interface

MTS는 메시지를 받게 되면 이름에 해당하는 에이전트를 등록 리스트에서 찾아 해당 에이전트에게 라우팅해준다. MTS가 처리할 수 없는 메시지의 경우 ACC에게 전달해 준다.

3.3 Agent Communication Channel

ACC는 다른 플랫폼과의 연결을 위한 기능을 가지고 있다. DCOM, IIOP를 통해서 다른 플랫폼에 메시지를 전달할 수 있다. ACC에 지원 프로토콜을 추가함에 따라 다른 에이전트 플랫폼과의 연동을 쉽게 지원할 수 있다. 또 메시지를 분석하여 비슷한 서비스를 제공하는 에이전트는 DF에서 찾아 메시지를 전달하는 기능도 가진다.

에이전트는 인터페이스를 이용해 플랫폼에 연결한 뒤 MTS를 통해 메시지를 전송할 수 있다. MTS는 receiver의 에이전트 ID를 보고 해당 에이전트에게 전송한다. 만약 receiver가 물리적으로 MTS와 연결되어 있지 않다면 AMS에 있는 delegate 항목에 등록된 에이전트에게 대신 전달한다. 이 과정을 거치고도 MTS가 전송할 수 없는 메시지는 ACC로 전송되고 ACC는 메시지를 분석하여 해당하는 에이전트로 전송한다. ACC로 메시지가 포워딩되는 경우는 대개 다른 플랫폼에 있는 에이전트에게 메시지를 보내는 경우이다. MAMI에서는 ACC가 inter-platform 통신을 담당한다.

3.4 Agent Management System

AMS는 등록된 에이전트의 정보를 저장하는 데이터베이스를 가지고 있으며, MTS도 이 데이터베이스를 참조하여 메시지를 라우팅 한다. AMS 에이전트를 관리하는 일을 하게 되는데, 기본 태스크로서 핑 태스크(ping task)를 가지고 있다. 이 핑 태스크는 에이전트들이 현재 on-line 상태인지를 체크한다. 휴먼 에이전트들은 핑 태스크의 대상이 아니다. 핑 태스크는 최하위 우선 순위를 가지고 있어서 처리할 메시지가 없는 경우에만 수행이 된다.

AMS는 에이전트를 관리하는 역할을 수행하고, 에이전트가 플랫폼의 서비스를 제공받기 위해서는 먼저 AMS에 등록해야 한다. 등록시에는 자신을 기술하는 내용(AMS Agent Description)을 보내야 한다. MAMI의 AMS 에이전트 기술에는 Name, Ownershp, Race, State, Delegate, Stylesheets가 있지만 FIPA2000 규격과는 약간의 차이가 있다. state에서 이동 에이전트를 고려하지 않기 때문에 Transit이 없고, delegate³⁾, race는 추가 항목이 있다. Delegate는 휴먼 에이전트와 소프트웨어 에이전트간의 결합을 위해서 추가되고, race는 휴먼 에이전트와 소프트웨어 에이전트를 구별한다. name 항목 역시 FIPA에서는 필수는 아니지만 MAMI에서는 반드시 필요한 항목이다. Style-Sheet 항목은

등록시에 에이전트가 보내지 않고, put-style을 하게 되면 AMS가 직접 채운다. AMS가 처리하는 연산에는 Register, Deregister, Modify, Search, Get-description, Put-style이 있다.

MAMI에서 추가된 연산은 put-style로 에이전트가 자신이 사용하는 언어의 XML Style-Sheet를 넣어주는 것이다. AMS는 웹 서버에 받은 내용을 기반으로 XML 스타일 시트(Style Sheet)를 생성해 준다. 에이전트는 자신이 사용하는 언어(language)별로 복수의 스타일 시트를 등록할 수 있다.

AMS는 연결된 에이전트의 관리를 위해 주기적으로 ping 메시지를 보낸다. MAMI의 에이전트들은 반드시 이 메시지에 응답을 보내야 한다. AMS는 3번 이상 응답하지 않는 에이전트는 연결이 끊어졌다고 판단하고 해당 에이전트를 삭제한다.

3.6 Delegate Agent

Delegate Agent는 휴먼 에이전트의 정보를 저장하고 있으며, 휴먼 에이전트와의 연결을 위해 웹과 연동하는 인터페이스를 가지고 있다. 웹 연동 인터페이스는 COM 오토메이션 인터페이스를 통해 제공되며, 필요에 따라 특정한 디렉토리에 파일을 생성해 웹을 통해 생성된 파일을 접근할 수 있게 해준다.

MAMI에서는 사람도 소프트웨어 에이전트와 동일하게 취급한다. 이를 위해서는 사람과 에이전트 플랫폼의 중간에서 매개체 역할을 하는 존재로서 위임 에이전트(Delegate Agent)가 존재한다. 위임 에이전트는 웹 서버와 연동되어 사람이 접근할 수 있는 인터페이스를 제공한다. 사람도 에이전트와 마찬가지로 에이전트 플랫폼의 서비스를 제공받기 위해서는 AMS에 등록해야 하는데, 위임 에이전트가 사람의 ID를 이용해서 AMS에 사람을 에이전트로 등록하고, 그 사람의 능력들을 DF에 등록해 광고할 수 있다. 앞으로 이렇게 등록된 사람을 휴먼 에이전트(human agent)라 지칭하고 에이전트는 인공으로 만들어진 에이전트를 지칭한다. 위임 에이전트는 휴먼 에이전트 등록시에 자신을 휴먼 에이전트의 대리자로 AMS에 등록한다. 이후에 만약 어떤 에이전트가 휴먼 에이전트에게 메시지를 보내게 되면 앞에서 기술한 MTS의 동작에 의해 위임 에이전트로 메시지가 전송되고 MTS는 웹 서버를 통해 휴먼 에이전트가 접속되었을 때 메시지를 전달하거나 푸쉬(push)를 이용해 바로 전달할 수 있다. 휴먼 에이전트의 접속 환경은 인터넷 익스플로러와 같은 XML 브라우저와 WAP 브라우저 환경으로 제한한다. 브라우저의 구분은 웹 서버를 통해 쉽게 할 수 있다. XML을 이용하는 방법은 4절에,

3) 이전 FIPA 규격에는 있었던 항목이다

WAP 이용은 5절에 기술한다. 웹 서버와 연동되는 Delegate 에이전트는 웹 서버와의 연동을 위한 인터페이스를 가지고 있다.

위임 에이전트는 COM의 로컬 서버 형태로 구현하고 오토메이션 인터페이스를 통해 ASP(Active-Server page)와 연동되어 웹 서비스를 제공해준다. 환경의 구분은 웹 서버가 브라우저 정보를 알려주기 때문에 쉽게 할 수 있다. WAP 환경에서는 여러 제약 상황 때문에 단순 통보만을 지원하고, XML 브라우저는 좀더 다양한 형태의 상호 작용이 가능하다. XML 브라우저의 경우 두 가지 형태로 구분할 수 있다. 하나는 에이전트로부터 휴먼 에이전트가 메시지를 받고 응답을 보내는 경우이고, 나머지 하나는 휴먼 에이전트가 에이전트의 서비스를 요구한 경우이다. 에이전트로부터 메시지를 받는 경우는 위임 에이전트는 다음과 같이 동작한다.

- ① 에이전트에게서 받은 메시지를 분석하고 저장한다.
- ② AMS에게서 메시지를 보낸 에이전트의 스타일 시트 주소를 가져온다. 없다면 not-understood 메시지를 보낸다.
- ③ 가져온 스타일 시트 주소와 보낸 메시지를 이용해 XML문서를 생성하고, 메시지의 CA와 프로토타입에 따라 선택할 수 있는 행동들을 버턴 형태로 문서에 추가한다. 만들어진 문서는 브라우저를 통해 휴먼 에이전트에게 보여진다.
- ④ 휴먼 에이전트는 필요한 입력을 채우고 자신이 원하는 행동을 선택한다.
- ⑤ 휴먼 에이전트가 선택된 행동과 입력을 이용해 응답 메시지를 만들어 이를 메시지를 보낸 에이전트에게 보낸다. 불른 응답할 필요가 없다면 메시지를 생성해 보내지 않는다.

휴먼 에이전트가 에이전트에게 먼저 메시지를 보내는 경우는 에이전트의 서비스를 요청할 때이고 위임 에이전트는 다음과 같이 동작한다.

- ① DF에게서 휴먼 에이전트가 필요로 하는 에이전트를 검색한다.
- ② 검색된 에이전트를 휴먼 에이전트에게 보여주고 에이전트와 원하는 서비스를 선택하게 해준다.
- ③ 선택된 에이전트의 서비스 페이지를 DF에게서 가져오고, 스타일 시트를 AMS에게 가져와 서비스 요청 XML 문서를 생성한다.
- ④ 생성된 문서에서 휴먼 에이전트는 입력이 필요한 항목을 채우고 서비스를 요청한다.
- ⑤ 입력과 서비스 요청 페이지를 이용해 메시지를 생성해 메시지를 에이전트에게 보낸다. 이때 메시지

의 Conversation-id, reply-to와 같은 파라미터는 위임 에이전트가 임의로 생성한다.

위 과정들을 살펴보면 휴먼 에이전트의 입력을 필요로 하는 경우가 있다. 사용자의 입력을 받는 화면은 스타일 시트를 이용하면 되지만, 입력을 받아 들이는 일은 위임 에이전트가 해주어야 한다. 이를 위해서는 입력이 필요한 부분을 구분해주는 공통의 XML 태그가 필요하다. MAMI에서는 <VARIABLE> 태그를 이용해 이를 구분한다. <VARIABLE> 태그의 이용에 따라 내용 계층 언어의 설계에 약간의 제약이 따르게 된다.

3.7 Directory Facilitator

DF는 에이전트 서비스를 저장하는 메타데이터를 관리하면서 에이전트 서비스를 등록, 질의할 수 있게 해준다. DF는 에이전트가 자신이 제공하는 서비스들을 광고하는 옐로우 페이지 역할을 한다. DF에 자신을 광고하기 위해서는 AMS와 마찬가지로 자신의 능력을 기술해서 등록해야 한다. 등록을 위해 필요한 항목들은 Name, Services, Protocol, Ontology, Language이다. DF가 제공하는 연산에는 Register, Deregister, Modify, Search, Get-service-page, Put-service-page가 있다. MAMI에서 DF는 소프트웨어 에이전트만이 아니라 사람도 위임 에이전트를 통해 등록할 수 있다. Put-service-page와 get-service-page는 에이전트의 서비스를 사람이 이용할 수 있게 해주는 XML 문서를 등록하고 가져오는 연산이다. service-page는 에이전트가 제공하는 서비스를 사람이 브라우저를 통해 요청할 수 있도록 하기 위해 만들어진 것이다. 위임 에이전트는 service-page를 이용해 사람이 서비스를 직접 요구할 수 있게 해준다.

4. ACL에서 XML이용

ACL에 XML을 이용한다는 것은 FIPA의 정의된 CA와 파라미터들의 의미를 그대로 유지하면서 인코딩 방법으로 XML을 사용한다는 것을 의미한다. MAMI에서 에이전트간의 통신에 사용되는 메시지는 XML로 인코딩[4]된 형태를 사용하며, FIPA의 CA와 파라미터의 의미를 그대로 따른다. 에이전트와 플랫폼의 연결은 COM(Component Object Module)을 사용한다. ACL 메시지 계층을 XML로 인코딩되면, 파서 개발이 용이하고, 새로운 ACL 구문 추가, 변경을 잘 지원할 수 있고, 웹 기술과 에이전트의 결합을 용이하게 하고, content에 사용되는 온톨로지를 공유하게 쉽게 하고, 링크 기능을 이용해서 에이전트의 정보가 있는 파일의 위치를 링크하게 되면 수신자는 송신자를 좀 더 잘 이해하게 된다[6].

그림 6은 XML로 인코딩된 메시지의 예이다. 내용 계

층을 XML로 인코딩할 경우, 위에 열거된 장점들 외에 또 하나의 장점으로 XML은 html과 달리 내용 중심으로서 뷰(view)가 분리되어 있어 여러 가지 형태로 디스플레이가 가능하다는 것이다[7]. XML 문서는 스타일 시트를 지정할 수 있으며, 이 스타일 시트에 따라 동일한 문서가 여러 가지 형태로 디스플레이 될 수 있다. XML의 이 특징은 멀티 에이전트 시스템과 휴먼 에이전트를 연동시키는데 많은 장점을 준다. 휴먼 에이전트와의 연동에서 가장 어려운 점은 에이전트 간에 이용되는 ACL을 사람이 쉽게 이해할 수 없기 때문이다. 따라서 에이전트가 사람에게 메시지를 전달할 때는 SKDL과 같이 자연어에 가까운 형태를 사용하고자 하는 시도들이 있다. 하지만 현재까지 특별한 해결방법이 없는 자연어 처리를 통한 접근 방식은 현실적으로 해결책을 제시하지 못 할 것으로 판단된다. 본 논문에서는 자연어 형태의 접근 방법 대신 인공 언어로 표현된 지식을 적절한 표현 양식으로 사람이 쉽게 이해할 수 있는 형태로 꾸미는 방법을 제안한다. 데이터베이스에 들어있는 복잡한 데이터들도 이들이 어떤 의미를 가지는지를 아는 응용 프로그램에 의해서 사람이 쉽게 이해할 수 있게 나타나는 것처럼 ACL도 잘 디스플레이한다면 사람이 쉽게 이해할 수 있을 것이다. 하지만 시스템을 구성하는 에이전트들은 아키텍처, 언어가 모두 동일할 수 없기 때문에 각 에이전트마다 자신의 ACL을 잘 디스플레이하는 인터페이스를 가지는 것은 많은 문제점이 있다. 각각의 에이전트에게 사용자 인터페이스를 구현한다는 것도 큰 오버헤드이고, 제각각으로 구현된 인터페이스를 연동하는 것은 매우 어려운 일이다. 웹 기술과 XML은 이와 같은 문제점을 해결해준다. 웹 기술은 통일된 인터페이스를 제공해주고, XML의 스타일 시트는 에이전트에게 가장 효과적으로 자신의 ACL을 디스플레이하는 방법을 제공해 준다.

에이전트가 휴먼 에이전트에게 메시지를 보낼 때 사람이 이해할 수 있는 자연어에 가까운 메시지를 보내지 말고, 자신이 사용하는 XML 형태의 메시지를 보내고, 이에 덧붙여서 사람이 쉽게 알아볼 수 있게 디스플레이 해주는 스타일 시트를 알려주는 것이다. 따라서 에이전트는 휴먼 에이전트에게 효과적으로 의미를 전달할 수 있다. 이와 같은 방법은 메시지를 받는 대상이 휴먼 에이전트 여부를 고려하지 않아도 되기 때문에, 에이전트의 부담을 줄여준다.

그림 6은 DICOM 에이전트가 휴먼 에이전트에게 의료 영상 판독을 질의하는 메시지의 예이다.

`dicom@gaint.knu.ac.kr`는 AMS에 자신의 스타일 시트를 이미 등록해둔 상태이다. 위임 에이전트는 먼저 메

```
<?xml version="1.0"?>
<!DOCTYPE mami_acl SYSTEM "mami_acl.dtd">
<message>
<messagetype> query </messagetype>
<messageparameter>
<sender>
<agent-identifier>
<name> dicom@gaint.knu.ac.kr </name>
<addresses> <sequence>
<addItem> dcom://gaint.knu.ac.kr/acc </addItem>
</sequence> </addresses>
</agent-identifier>
</sender>
<receiver> wonki@gaint.knu.ac.kr </receiver>
<language> dicom_diagnosis </language>
<protocol> fipa-query </protocol>
<reply-with> query-0011 </reply-with>
<reply-by> 30 </reply-by>
<content>
<dicom-image>
<patient>
<name> sang-wook </name>
<sex> male </sex>
<age> 25 </age> </patient>
<image link=
  http://gaint.knu.ac.kr/dicom/c0001.dcm/>
<diagnosis>
<VARIABLE> opinion </VARIABLE>
</diagnosis>
</dicom-image>
</content>
</messageparameter> </message>
```

그림 6 XML로 인코딩된 메시지 예

시지 타입이 query이고, 프로토콜이 fipa-query인 것을 보고 <VARIABLE> 태그에 있는 내용을 휴먼 에이전트가 입력해야 한다는 것을 알고 그림 7과 같은 화면을 보여주게 된다. content를 등록된 스타일 시트와 결합하면 템플릿 형태로 디스플레이된다. 그림 7의 맨 밑에 있는 버튼들은 fipa-query 프로토콜에서 query에 응답을 선택할 수 있게 해주는 것이다. 버튼들은 각각 not-understood, refuse, inform에 해당한다.

휴먼 에이전트가 일정 시간(reply-by) 동안 접속하지 않아 메시지를 보낼 수 없을 때, 위임 에이전트는 failure 응답을 보내게 된다. 예에서 나타나는 것처럼 xml과 스타일 시트의 활용을 통해 사람이 쉽게 에이전트에게 지식을 공유할 수 있다. 다른 CA들과 프로토콜들에 대해서도 각 상황에 맞는 화면을 보여주어 에이전트와 휴먼 에이전트의 통합을 MAMI에서 제공해 준다.

5. WAP의 활용과 평가

WAP은 디스플레이를 비롯해 여러 가지 제약사항을

가지고 있다. WAP과 같은 무선 통신 기술은 휴먼 에이전트와 바로 연결할 수 있는 좋은 환경을 제공해주지만, 현재는 여러 제약들에 의해 4절에서 XML을 이용하는 지식교환 방법을 지원할 수가 없다. 그래서 현재 MAMI에서는 두 가지 형태로 이용하고 있다. 하나는 간단한 자연어로 여러 지식을 전달하는 것이고, 또 하나는 메시지가 도착했음을 알리고 XML 브라우저를 통해 접속할 것을 요청하는 것이다. 에이전트가 휴먼 에이전트에게 메시지를 보낼 때 language 파라미터가 simple-natural일 경우가 전자에 속하고 나머지 언어들은 후자에 속한다. 전자의 경우 메시지를 보내는 에이전트는 받는 에이전트가 사람인 것을 인지하고 간단한 자연어 메시지를 보내준다. 그림 6과 같은 메시지가 도착하게 되면, 어떤 에이전트로부터 메시지가 도착했으며 언제까지 답을 받기를 희망한다는 것을 그림 8처럼 보여준다. 불가 항목은 해당 시간 안에 응답이 불가능할 때 선택하고 이때 위임 에이전트는 failure 메시지를 전송하게 된다.

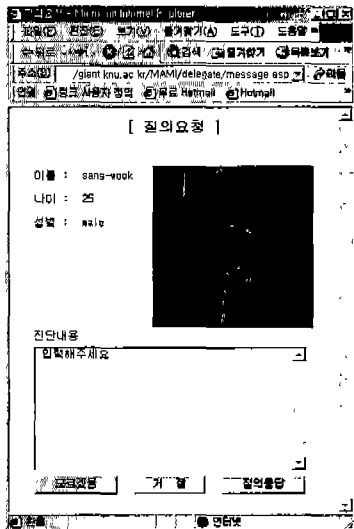


그림 7 질의 메시지 뷰 페이지



그림 8 WAP 단말기를 통한 메시지 예

지능형 의료 정보 멀티 에이전트 플랫폼과 다른 에이전트 플랫폼과의 기능적 비교를 표3에 기술하였다[8,9,10]. 표에서 나타나듯이 아직 다른 플랫폼들에서는 전문 인력과 연동보다는 소프트웨어 에이전트 간의 연동에 더 초점을 맞추고 있다.

표 3 다른 플랫폼과의 비교

	본 플랫폼	FIPA OS	Grass-Hopper
FIPA 규격	준수	준수	준수하지 않음
Protocol	COM	CORBA	CORBA TCP RMI
Mobile Agent	비지원	비지원	지원
ACL	XML encoded FIPA ACL	FIPA	
인력과의 연동	지원	지원하지 않음	지원하지 않음

기존의 소프트웨어 패러다임에서는 사람은 사용자로서 소프트웨어는 사용자에게 의해 조작되는 유틸리티 (utility)로 사람에게 종속되어 사용되어 왔다. 에이전트라는 새로운 소프트웨어 패러다임에서는 사람의 조작에 의해 종속되는 소프트웨어가 아니라 독자적으로 동작하며 사람에게 서비스를 제공하거나 필요에 따라 사람으로부터 도움을 받는 - 지시를 받는 것이 아니라 - 관계로의 변환이 필요할 것으로 생각된다. 본 플랫폼에서는 이러한 패러다임 변화를 수용할 수 있게 하기 위해 전문 인력과 소프트웨어 에이전트 연동의 한 방법을 제시하고 있다. 에이전트 시스템이 동작하는 제한된 환경을 사람이 인식하기 쉬운 형태로 변환하고 이를 사람에게 전달하는 방법을 통해 사람을 하나의 휴먼 에이전트로 모델링하고 있다.

6. 결론 및 향후 과제

MAMI는 FIPA 플랫폼을 기본 골격으로 한 의료 영상 멀티 에이전트 시스템 플랫폼으로서 XML로 인코딩된 ACL을 사용하여 에이전트간의 통신을 가능하게 해주며, 휴먼 에이전트를 이용하여 전문 의료 인력과 지식 교환을 용이하게 하기 위한 서비스들을 제공하고 있다. 사람을 휴먼 에이전트로 개념화하기 위해 위임 에이전트가 플랫폼에 있으며, 위임 에이전트는 WAP과 웹 환경을 통해 휴먼 에이전트와 접속한다. 휴먼 에이전트와 지식교환을 위해 XML과 스타일 시트를 이용하는 방법을 제안하였다.

향후 과제로는 다른 FIPA 플랫폼과의 상호 운용성을

지원해야 한다. 또 SOAP(Simple Object Access Protocol)을 이용하는 에이전트 간 메시지 전달 메커니즘을 통해 웹과의 연동을 용이하게 하는 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] Michael N. Huhns and Munidar P. Singh, Readings In Agents, Morgan Kaufmann Publishers Inc., pp.1-24, 1998.
- [2] Gerhard Weiss, Multiagent System, MIT Press, pp.79-82, 1999.
- [3] FIPA, "FIPA Agent Management Specification," <http://www.fipa.org>, 2000.
- [4] FIPA, "FIPA ACL Message Representation in XML Specification," <http://www.fipa.org>, 2000.
- [5] FIPA, "FIPA 98 Part 8 Version 1.0: Human-Agent Interaction Specification," <http://www.fipa.org>, 1998.
- [6] Benjamin N. Grosz and Yannis Labrou, "An Approach to using XML and a Rule-based Content Language with an Agent Communication Language," Proceedings of the IJCAI-99 Workshop on Agent Communication Languages (ACL-99), 1999
- [7] Simon St. Laurent, Ethan Cerami, Building XML Application, McGraw-Hill, pp.128-149, 1999.
- [8] S. Poslad, P. Buckle, R. Hadingham, "Open Source Standards and Scaleable Agencies," Barcelona, Spain, June 2000.
- [9] S. Poslad, M. Calisti, "Towards Improved trust and security in FIPA Agent Platforms," Barcelona, Spain - June 2000.
- [10] IKV++, "Grasshopper Basics and concepts," <http://www.grasshopper.de>, 2000.



김 일 곤

1980년 서울대학교 수학교육과 졸업, 1988년 서울대학교 전산학과 석사학위 취득. 1991년 서울대학교 전산학과 박사학위 취득. 1992년 ~ 현재 경북대학교 컴퓨터학과 부교수로 근무. 1997년 3월 ~ 1998년 8월까지 미국 조지타운대학교 병원 ISIS 연구소 방문연구자. 관심분야는 지능형 에이전트 시스템, 분산시스템, 의료정보학



최 원 기

1996년 경북대학교 컴퓨터학과 졸업. 1998년 경북대학교 컴퓨터학과 석사학위 취득. 2001년 경북대학교 컴퓨터학과 박사 과정 수료. 2001년 3월 ~ 현재 모빌 씨앤씨 연구소 근무. 관심분야는 지능형 에이전트 시스템, 분산 객체, 모바일 컴퓨팅

일 컴퓨팅