# 웹 상에서 동작하는 에이전트 기반 일정 관리 시스템의 설계 및 구현

유재홍°, 성미영, 채진석
시립인천대학교 전자계산학과

# Design and Implementation of an Agent-based Schedule Management System for Collaborative Work on the Web

Jaehong Yoo°, Meeyoung Sung, Jinseok Chae
jhyoo@isis.inchon.ac.kr, mysung@lion.inchon.ac.kr, jschae@isis.inchon.ac.kr
Department of Computer Science, University of Inchon

요 약

이 논문은 웹 상에서 동작하며 지능을 가지는 에이전트의 도움을 받아 효율적으로 일정을 관리할 수 있도록 도와주는 일정 관리 시스템의 설계 및 구현에 대해 기술한다. 이 시스템은 멀티 에이전트 구조를 가지고 있으며, 에이전트들 간의 통신을 위해서는 KQML(Knowledge Query and Manipulation Language)을 사용하고 있다. 이것은 시스템에 유연성과 적응성을 제공해 주며, 새로운 에이전트를 추가할 때 에이전트들 간의 이형질성을 극복할 수 있도록 해준다. 또한, 웹 환경을 이용함으로써 클라이언트 프로그램을 따로 설치할 필요 없이 어느 곳에서나 웹 브라우저를 이용하여 협동작업을 수행할 수 있다.

## 1. 서론

지난 10여 년 동안 멀티미디어 기술과 컴퓨터 네트워크 기술이 급속하게 발전하였고 이러한 기술들의 결합으로 CSCW(Computer Supported Cooperative Work)[1,2,3] 분야의 급속한 발전과 더불어 원격회의, 원격교육, 원격자문, 공동저작 등에 대한 요구가 날로 커지고 있다. 또한, 국내외의 많은 연구자들이 에이전트에 대한 연구에 관심을 쏟고 있는데, 이는 그들의 시스템을 유연성과 적응성을 갖춘 지능형으로 발전시키는데 에이전트가 중요한 역할을 할 것이라 기대하기 때문이다. 이러한 에이전트가 실세계 시스템인 협동작업 시스템에서 활약한다면 기존의 여러 소프트웨어에 존재하는 기능성을 인간에게 보다 편리한 형태로 서비스할 수 있을 뿐만 아니라, 웹 브라우저만을 가지고 실세계에서와 같이 동기적이고 협동적인 작업을 지원할 수 있을 것이다.

이 논문에서는 여러 에이전트들의 협동작업을 통해 여러 전문가들에게 동기적 협력을 보다 편리한 형태로 지원해 주며, 분산 협동 처리를 위한 에이전트들 간의 통신에 있어 가장 큰 문제점인 이형질성(heterogeneity)을 극복하기 위해 KQML을 사용하여 지식을 교환하는 기법을 제시하고 있다.

우선 에이전트 기반의 일정관리 시스템에 관한 관련 연구로서 에이전트와 KQML에 대해 간단히 살펴본 다음, 본 논문에서 제안한 전체 시스템의 구성과 에이전트들 간에 통신을 위해 사용된 KQML 메시지의 흐름에 대해 자세히 살펴보고 결론을 맺겠다.

## 2. 연구 배경

에이전트에 대한 연구는 1959년 John McCarthy의 "The Advice Taker"와 Oliver Selfridge의 "The Pandemonium Paradigm for Learning"으로부터 그 개념이 싹트기 시작하여 최근 7, 8년 동안 활발하게 연구되어 왔다. 에이전트의 개념은 인간의 방식으로 소통하고 여러 가지 일을 인간의 방식으로 처리하는 목적 지향적인 실체로 이해될 수 있으며 컴퓨터 안에서 살아가는 소프트 로봇(soft robot)이라고 할 수 있다. 일반적 에이전트는 자율성, 사회성, 이동성, 지능 등의 특성을 가진다[5]. 이 특징을 모두 갖추어야 에이전트는 아니며, 이러한 특성을 얼마나 많이 가지냐에 따라서 얼마나 능동적인가를 판단하는 척도가 된다.

### 2.1 멀티 에이전트 시스템

과거 1980년대 말까지도 에이전트 기반 시스템은 하나의 에이전트로 구성된 것이 주류였다. 그러나 사용자의 요구사항이 다양해짐에 따라 단독의 에이전트가 인지적인 측면(cognitive aspect)에서 이를 해결하기 위한 지능을 가지기는 어렵기 때문에 분산 인공지능(distributed artificial intelligence)을 이용하여 지역적으로 떨어진 다른 에이전트의 도움을 받아 처리하는 분산 협동 처리 개념을 접목시켜야할 필요가 생겨났다. 이를 멀티 에이전트(multi-agent)라 부른다.

멀티 에이전트는 응용 에이전트들 외에 조정 에이전트(coordinating agent, facilitator)라는 중개자를 통해 메시지 전달과 각 에이전트의 제어를 수행하게 된다. 여기서 조정 에이전트는 모든 응용 에이전트의 위치 정보, 각 에이전트의 처리 능력 등의 정보를 가지고 각각의 응용 에이전트를 연결해 주는 네임 서버 역할을 한다. 여기서 응용 에이전트들이 매우 다른 구조를 가지는 에이전트 환경이라면 에이전트들 사이에 인터페이스로써 표준화된 메시지 형태와 프로토콜이 필요하다.

### 2.2 KQML

멀티 에이전트 시스템의 표준화된 메시지 형태와 프로토콜에 대한 연구 중 대표적인 것에는 KQML((Knowledge Query and Manipulation Language)[4]이다.

KQML은 KIF(Knowledge Interchange Format)와 같이 ARPA(Advanced Research Projects Agency)가 후원하는 Knowledge Sharing Effort에서 나왔다. KIF는 지식의 표현을 다루는 반면, KQML은 동작하고 있는 에이전트들 사이의 메시지 포맷과 메시지 핸들링 프로토콜(message-handling protocol)에 초점을 둔다. KQML은 각각 다른 지식 베이스에서 수행될 연산들(operations)을 정의하고, 조정 에이전트를 통해 지식과 정보를 공유하려 하는 에이전트들을 위한 기초적인 구조를 제공한다.

KQML 메시지는 수행문(performative)이라고도 불리며, 각 메시지는 암시적으로 몇 가지 명세 된 행동을 수행하도록 예약되어져 있다. KQML에는 많은 수의 수행문들이 정의되어있고, 대부분의 에이전트 기반의 시스템들은 단지 작은 부분만을 지원한다. 수행문을 사용하는 에이전트는 다른 에이전트에 정보를 요구할 수 있으며, 다른 에이전트에 자신의 지식베이스에 존재하는 지식들을 가르쳐줄 수 있고, 에이전트들의 서비스를 예약할 수 있으며 그들 자신의 고유한 서비스들을 제공할 수도 있다.

## 3. 전체 시스템 구조

본 논문에서 제안하는 에이전트 기반의 일정 관리 시스템은 그림 1과 같이 여러 에이전트가 하나의 시스템을 구성하는 멀티 에이전트 구조를 가진다. 따라서 이 장에서는 전체 시스템의 구조를 살펴본 후, 이 시스템이 제공하는 기능과 각 기능을 구성하는 에이전트들에 대해 자세히 살펴보고 다음으로 에이전트들 사이에 주고받는 KQML 메시지의 흐름에 대해 살펴보도록 하겠다.

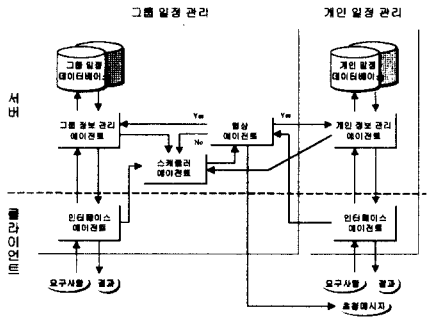


그림 1 일정 관리 에이전트

### 3.1 그룹 일정 관리

그룹 일정 관리에서는 사용자로부터 그룹 일정 열람과 함께 회의 소집에 대한 제안을 받아들여 시간을 제시하고 그룹 구성원들에게 초청 메일을 전송한 후, 응답을 받아 회의 스케줄을 생성해 주는 일을 한다. 다음은 각 에이전트의 기능에 대한 설명이다.

• 인터페이스 에이전트 - 사용자들에게 요구사항을 받아 그룹 정보 관리 에이전트에게 전달하고 해당되는 알맞은 정보를 받아서 사용자에게 보여준다.

• 그룹 정보 관리 에이전트 - 그룹의 일정 정보, 사용자의 인증 정보를 관리하고 다른 에이전트에게 제공해준다. 예를 들어, 인터페이스 에이전트의 그룹 일정 열람 요구, 사용자 인증, 스케줄러 에이전트의 그룹 정보 요구 등이 있다.

• 스케줄러 에이전트 - 그룹/개인 정보 관리 에이전트로부터 그룹/개인 스케줄 정보를 얻어 회의를 위한 적절한 시간을 추론하여 제시한다. 이 정보는 협상 에이전트에 보내진다.

• 협상 에이전트 - 그룹 멤버들에게 회의를 위한 적절한 시간과 회의 성사 여부를 전달하고, 그룹 구성원들의 회의 참석 여부의 응답을 받아 회의 소집의 성사 여부를 처리한다. 모임이 성사되지 못했으면 스케줄러 에이전트에게 해당 메시지를 전송한다, 회의가 성사되면 각 그룹 구성원에게 메일을 전송하고 그룹/개인 정보 관리 에이전트를 통해 그룹/개인 일정 데이터베이스를 갱신한다.

### 3.2 개인 일정 관리

개인 일정 관리에서는 사용자로부터 개인 일정 열람에 대한 정보나 협상 에이전트와는 별도로 다른 멤버의 모임 소집 제안에 대한 정보 등을 제시하여 주는 일을 한다. 다음은 각 에이전트에 대한 설명이다.

• 인터페이스 에이전트 - 사용자들에게 요구사항을 받아 개인 정보 관리 에이전트에게 전달하고 해당되는 알맞은 정보를 사용자에게 보여준다. 또한 다른 사용자의 회의 소집 제안이 있을 때에 정보를 사용자에게 보여준다.

• 개인 정보 관리 에이전트 - 인터페이스 에이전트의 요구사항이 개인 일정 열람이면 개인 일정 데이터베이스에서 얻어온 정보를 인터페이스 에이전트에게 전달해 준다. 요구사항이 모임 소집 제안에 대한 응답이면 협상 에이전트에게 정보를 전달한다.

## 4. KQML 적용 및 구현 환경

### 4.1 KQML 메시지의 흐름

KQML 메시지는 전체 시스템에서 이루어지는 작업들 중에서 자신의 일정 관리에서부터 회의를 소집하기까지의 과정에서 일어날 수 있는 작업들에 적용된다

KQML 메시지가 적용되는 작업을 살펴보면 다음과 같다.

- 사용자 등록 및 인증
- 개인 일정 열람
- 그룹 일정 열람
- 회의 소집 제안
- 회의 소집 협상

이 절에서는 지면상 회의 소집 협상에 KQML을 적용한 예를 설명한다. 예약 수행문(reserved performatives)과 VKB(Virtual Knowledge Base)의 개략적인 내용만을 포함하여 표현하였을 뿐, 수행문 전체 문장과 VKB의 세부적인 내용에 대해서는 명세하지 않는다. 각 작업의 진행 순서를 설명하는데 있어서 "<"와 ">"를 사용하여 VKB를 표현하였다.

### 4.2 회의 소집 협상에서 KQML 적용 예

회의 소집 협상은 회의 소집 제안 다음에 이루어지는 작업이며, KQML 메시지의 흐름은 그림 2와 같다. 스케줄러 에이전트에 의해 추측된 회의 소집 날짜를 그룹 구성원에게 전달한 후, 그 결과를 받아 그룹 구성원들은 투표를 하게 되고, 협상 에이전트는 그룹 구성원의 투표 결과를 집계한다. 정해진 정책(예, 회의 소집 조건이 구성원의 과반수 이상)에 맞으면 회의 소집 날짜가 결정되고, 이 날짜는 그룹 일정 관리 에이전트와 개인 일정 관리 에이전트에 보내져 일정 정보에 추가된다. 반대로 정책에 어긋나면 결과를 다시 스케줄러 에이전트에 보내 다른 날짜를 결정하도록 한다. 이 과정을 반복하여 그룹 구성원들이 원하는 날짜를 결정한다.

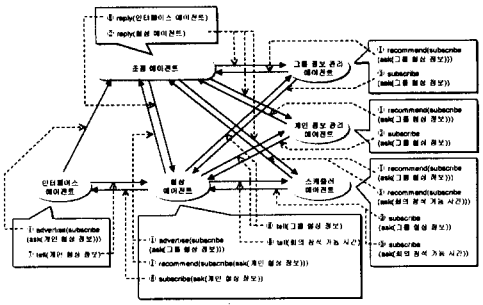

그림 2 회의 소집 협상을 위한 KQML 메시지의 흐름

각 에이전트에서 사용되는 수행문과 그들의 순서를 살펴보면 다음과 같다.

① 서버 측에 존재하는 스케줄러 에이전트, 협상자 에이전트, 그룹 정보 관리 에이전트 그리고 개인 정보 관리 에이전트는 조정 에이전트에게 다음과 같은 사실을 요청하거나 알린다.

• 스케줄러 에이전트 : <그룹 협상 정보>와 <회의 참석 가능 시간>이 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 요청한다.

• 협상 에이전트 : <그룹 협상 정보>가 갱신될 때마다 알려줄 수 있다고 알린다, <개인 협상 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 요청한다.

• 그룹 일정 관리 에이전트 : <그룹 협상 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보 요청한다.

• 개인 일정 관리 에이전트 - <그룹 협상 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보 요청한다.

② 조정 에이전트는 그룹 일정 관리 에이전트, 개인 일정 관리 에이전트, 스케줄러 에이전트에게 필요한 정보를 가지고 있는 에이전트에 대한 정보를 넘겨준다.

③ 그룹 정보 관리 에이전트와 개인 정보 관리 에이전트는 <그룹 협상 정보>가 갱신될 때마다 그 정보를 알려달라고 요청한다. 스케줄러 에이전트는 <그룹 협상 정보>와 <회의 참석 가능 시간>이 갱신될 때마다 그 정보를 알려달라고 요청한다.

④ 인터페이스 에이전트는 <개인 협상 정보>가 갱신될 때마다 알려줄 수 있다고 조정 에이전트에 알린다.

⑤ 조정 에이전트는 <개인 협상 정보>를 인터페이스 에이전트가 알려줄 수 있다고 협상 에이전트에게 알려준다.

⑥ 협상 에이전트는 <개인 협상 정보>가 갱신될 때마다 그 정보를 알려달라고 인터페이스 에이전트에게 요청한다.

⑦ 인터페이스 에이전트는 <개인 협상 정보>가 갱신될 때마다 협상 에이전트에게 그 정보를 알려 준다.

⑧ 협상 에이전트는 회의 소집 협상이 타결되면 그룹 정보 관리 에이전트와 개인 정보 관리 에이전트에게 <그룹 협상 정보>를 알려 준다. 회의 소집 협상이 결렬되면 스케줄러 에이전트에게 각 사용자들의 <회의 참석 가능 시간>을 알려주어 스케줄러 에이전트에게 다른 시간을 추론하도록 한다.

### 4.3 구현 환경 및 수행 결과

이 논문에서 제안한 시스템의 구현에 사용된 환경으로는 서버용 운영체제로 Solaris 2.6을 사용하고 클라이언트용 운영체제로 Windows 98을 사용하였다. 데이터 베이스는 객체지향 데이터베이스인 O2를 사용하였고, 개발 도구로는 JDK 1.2와 JATLite[6]를 사용하였다. JATLite는 Stanford 대학에서 Java 언어를 통해 개발한 템플리트(template)이며, 자바 언어를 사용하여 KQML 메시지를 교환하는 에이전트 프레임워크(framework) 개발을 쉽게 해준다.

본 시스템의 인터페이스는 사용자의 개인 일정과 그룹 일정 열람, 사용자의 개인 일정과 그룹 일정 등록, 그리고 회의 소집 요청을 할 수 있도록 구성되어 있으며, 개인 일정과 그룹 일정 열람 부분에서는 일별, 주별, 월별 일정 열람을 할 수 있도록 구성이 되어있다. 그림 3은 개인 일정 관리 중에서 월별 일정 관리를 위한 인터페이스를 나타낸다.

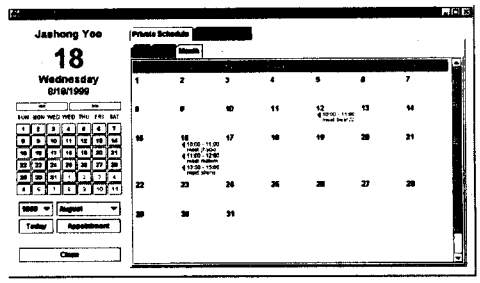


그림 3 일정관리 인터페이스

## 5. 결론

이 논문에서는 웹 기반 일정 관리 시스템과 에이전트 개념(Agency)을 결합한 시스템에 대한 연구이다. 일정 관리 시스템은 회의를 소집해주는 능동적인 객체로서 그룹과 개인의 일정을 관리하여 회의를 원하는 그룹 구성원이 다른 구성원들에게 회의에 초청할 수 있도록 지원해 준다.

본 시스템의 가장 큰 장점은 아래와 같다.

(i) 시스템의 유연성과 적응성을 기대할 수 있고, (ii) 에이전트들 간의 이형질성을 극복할 수 있다. 에이전트들 간의 통신에 있어 KQML을 사용함으로써 기존 지식 베이스의 응용 시스템이나 새로이 만들어지는 지식 베이스는 쉽게 기존의 시스템에 연결될 수 있고 그 기능성을 이용할 수 있다. 또한, (iii) 웹 환경을 이용함으로써 운영체제나 데이터베이스에 관계없이 저렴한 비용으로 쉽게 어플리케이션을 개발할 수 있고 인터넷과 연결된 곳이라면 어느 곳에서나 웹서버에 접속함으로써 협동작업을 수행할 수 있다.

이 시스템은 멀티 에이전트 환경이므로 에이전트간의 동기적 협력을 통해 복잡한 문제를 해결할 수 있는 효율적인 기법의 연구가 요구된다.

## 참고문헌

[1] Stephen Jabele, Steven Rohall, Ralph L. Vinciguerra, "High Performance Infrastructure for Visually-Intensive CSCW Applications", Proceedings on CSCW '94, ACM Press, pp. 395-403, October 1994.

[2] R. Steinmets and N. Nahrstedt, "Multimedia: Computing, Communications & Applications", Prentice Hall, pp. 854, 1995.

[3] Eric Garland and Dave Rowell, "Face-to-Face Collaboration", Byte, Vol. 19, No. 11, pp.233-242, November, 1994.

[4] Munindar P. Singh, "Agent Communication Languages: Rethinking the Principles", IEEE Computer, Vol. 31, No. 12., pp. 40-47, December 1998.

[5] "Intelligent Agents", Communication of the ACM, Vol. 37, No. 7, July 1994.

[6] http://java.stanford.edu/