

## 회의 소집을 위한 다중 에이전트 시스템의 구현\*

유재홍

인천대학교 컴퓨터공학과  
(hyoo@incheon.ac.kr)

노승진

인천대학교 컴퓨터공학과  
(sjrhn@incheon.ac.kr)

성미영

인천대학교 컴퓨터공학과  
(mysung@incheon.ac.kr)

이 연구는 웹 기반 협동작업 시스템에서 지능적인 에이전트들을 이용하여 원격 회의의 소집을 자동화함으로써 여러 참여자들에게 보다 편리한 형태의 협동작업 환경을 지원해 주는 시스템에 대한 것이다. 회의 일정을 잡는 작업을 자동화하는 것은 각 개인의 공적인 일정 뿐만 아니라 개인의 사정과 선호도 등에 대한 세심한 배려를 요구한다. 그러므로, 회의 소집 자동화는 신뢰성과 병렬성을 높이기 위해서 각 개인의 일정을 병렬적으로 별도 관리하는 작업을 지원하는 분산 처리 작업이 요구된다. 이 논문에서는 회의 소집에 들이는 시간과 노력을 최소로 줄여주는 다중 에이전트 시스템의 설계와 구현에 대하여 자세하게 소개한다. 이 시스템은 서버-클라이언트 모델에 기반하며, 서버 측에서는 스케줄 에이전트, 협상 에이전트, 개인정보 관리 에이전트, 그룹 정보 관리 에이전트, 세션관리 에이전트와 조정 에이전트가 활동한다. 클라이언트 측에서는 인터페이스 에이전트, 미디어 에이전트와 협동 에이전트가 활동한다. 이들 에이전트들은 통신은 표준화된 지식 표현 언어를 이용하여 통신하므로 분산 협동 처리를 위한 에이전트들 간의 통신에 있어 가장 큰 문제점인 이형질성을 극복할 수 있도록 해준다. 본 시스템의 회의소집 지원 에이전트들은 전진연결 알고리즘으로 추론하고 역전파 네트워크 알고리즘으로 학습하여 가장 많은 사람이 참여할 수 있는 날짜를 제안해줌으로써 회의 소집자가 회의 소집에 들이는 노력을 최소로 줄일 수 있게 해 준다.

### 1. 서론

80년대 중반에 출현한 컴퓨터 지원 협동작업(CSCW; Computer Supported Cooperative Work) (Jabele et al., 1994; Steinmets & Nahrstedt, 1995; Garland & Rowell, 1994) 분야는 이후에 있었던 멀티미디어 기술과 컴퓨터 네트워크 기술의 급속한 발전에 힘입어 최근 들어 그 연구가 더욱 활발하게 진행되고 있

다. 한편, 국내외의 많은 연구자들이 에이전트에 대한 연구에 관심을 쏟고 있다. 이는 그들의 시스템을 유연성과 적응성을 갖춘 지능형으로 발전시키는데 에이전트가 중요한 역할을 할 것이라 기대하기 때문이다. 에이전트는 자율성(autonomy), 사교성(social ability), 이동성(mobility), 지능(intelligence) 등의 특성을 가지며, 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어이

\* 이 논문은 한국과학재단 핵심전문연구(과제번호 981-0927-148-2)의 지원으로 연구되었음

다. 이러한 에이전트가 실세계의 협동작업 시스템에서 활약한다면 기존의 여러 소프트웨어에 존재하는 기능을 인간에게 보다 편리한 형태로 제공할 수 있다.

국외의 에이전트(Agents) 연구는 1990년 초반에 걸쳐 사용자 또는 그룹의 타협(user or group negotiation)을 위한 에이전트 기술(A Framework for Negotiation), 협동 에이전트들의 협동 환경(Environments for Cooperating) 정보 검색을 위한 지능적 에이전트(Domain Knowledge Agents for Information Retrieval) 기술연구, 공동 저작을 위한 에이전트 기술(Autonomous Agents in the Support of Cooperative Authorship) 연구(John, Connolly and Ernest, 1994) 등과 같이 다양한 연구활동이 있었으며, 프로그래밍과 교육 분야에서도 에이전트 개념이 효과적으로 응용되어 KidSim (Kid's Simulation)이라는 프로그래밍 에이전트나 COACH (Cognitive Adaptive Computer Help) (Communication of the ACM, 1994) 라는 교수 에이전트(teaching agent) 등이 구현되었다.

국내의 에이전트 분야에 대한 연구도 최근 들어 활발하게 진행되고 있다. 국내의 에이전트 분야의 대표적인 연구로는 검색 에이전트, 웹 에이전트, 이동 에이전트, 학습 에이전트, 적응형 에이전트, 전자 상거래를 위한 에이전트 등이 있는데, 대부분의 연구가 단독 사용자의 편리를 위한 에이전트 연구이며 협동 환경을 위한 에이전트 시스템에 대한 연구는 미약한 실정이다.

웹서비스는 웹 서버가 제공할 서비스에 대한 정보를 저장하고 있고 사용자는 원하는 경

우에 단순히 정보를 받아 보는 단 방향 서비스인 반면 화상회의(Garland & Rowell, 1994; Reinhard et al., 1994; Greenberg & Marwood, 1994; ITU-T Draft Recommendation T.120, 1996) 등의 협동작업 시스템은 여러 사용자가 시스템과 상호 작용할 수 있어야 협동이 가능하다. 현재 공동저작(collaborative writing)과 원격회의(teleconferencing) 등의 실시간 그룹 지원 응용들이 많이 존재한다. 이러한 그룹 지원 시스템들을 사용할 때 사용자들의 긴밀한 협동을 위해서 회의 소집 과정을 필요로 한다.

회의 소집 과정은 매우 따분하고 반복적이며 시간이 많이 드는 작업이다. 회의에 참여하는 각 개인들은 어떤 회의를 할만한 시간이 있음에도 불구하고 공통적으로 회의가 가능한 시간을 찾아내는 일은 매우 어렵다. 전자 메일이나 전화 등을 이용하여 공통적으로 가능한 시간을 찾아내는데 드는 통신적 지연 시간이 많고, 한 회의 일정을 잡는 과정과 동시에 다른 회의의 일정을 잡은 일이 병렬적으로 발생할 수 있기 때문에 간혹 잡은 일정이 적절하지 않거나 최적 아닐 수가 있다. 그러므로, 회의 소집 과정은 아주 적절한 자동화의 대상이다. 이미 1980년 초반부터 개인 및 조직의 일정관리에 다수의 연구들이 수행되어 왔다(Kincaid & Dupont & Kaye, 1985). 최근에는 분산 인공지능(Distributed Artificial Intelligence) 연구의 일환으로 회의 소집 자동화가 다양하게 연구되고 있다. 특히 Sandip Sen과 Edmund H. Durfee는 다단계 협상 프로토콜(multistage negotiation protocol)을 이용하는 분산 회의 스케줄링(DMS; Distributed Meeting Scheduling)

해법을 제안하였으며 통계적인 접근으로 회의 소집 자동화 모델을 정형화(formalism) 하는데도 기여하였다(Sen et al., 1998).

하나의 에이전트는 기본적으로 세 가지로 구성된다. 이 세 가지는 제어지식과 추론능력 등을 가지는 에이전트 엔진, 특정 응용분야의 작업 수행에 필요한 지식인 영역 지식, 그리고 다른 에이전트와의 협동을 통해 문제를 해결하기 위한 통신모듈이다. 예전에는 에이전트-기반 시스템이 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 접목시켜야 할 필요가 생겼다. 이를 멀티 에이전트(multi-agent)라 부르며, 이러한 멀티 에이전트 시스템이 되기 위해서는 각 에이전트 사이의 중재(negotiation)와 협력(coordination) 기법 등이 먼저 정의되어야 하고, 이들 에이전트들이 지능적으로 동작할 수 있어야 한다.

이 연구는 웹 기반 협동작업 시스템에서 지능적인 에이전트들을 이용하여 원격회의의 소집을 자동화함으로써 여러 참여자들에게 보다 편리한 형태의 협동작업 환경을 지원해 주는 시스템에 대한 것이다. 회의 일정을 잡는 작업을 자동화하는 것은 각 개인의 공적인 일정 뿐만 아니라 개인의 사정과 선호도 등에 대한 세심한 배려를 요구한다. 그러므로, 회의 소집 자동화는 신뢰성과 병렬성을 높일 수 있도록 각

개인의 일정을 병렬적으로 별도 관리하는 작업을 지원하는 분산 처리 작업이 요구된다. 이 연구에서는 시스템의 확장성을 고려해 분산 협동 처리를 위한 에이전트들 간의 통신에 있어 가장 큰 문제점인 이형질성(heterogeneity)을 극복하기 위해 표준화된 지식 교환 언어 KQML(Knowledge Query and Manipulation Language) (Munindar, 1998)를 이용하여 지식을 교환하는 기법을 사용하고자 한다. 이 방법은 기존 지식 베이스의 응용 시스템이나 새로이 만들어지는 지식 베이스를 쉽게 기존의 시스템에 연결하여 그 기능성을 이용할 수 있도록 지원해준다.

이 논문에서는 웹 기반 협동작업 시스템에서 지능적인 에이전트들을 통해 회의 소집에 들이는 노력을 최소화함으로써 여러 참여자들에게 동기적 협력을 보다 편리한 형태로 지원해 주는 시스템을 설계하고 구현한 내용에 대하여 자세하게 소개한다. 논문의 구성은 다음과 같다. 2장에서는 다중에이전트 기반의 전체 시스템 구조를 소개하고, 3장에서는 회의소집을 위한 시스템 내부 모델에 대하여 살펴보겠다. 4장에서는 시스템 내의 몇몇 에이전트들이 사용하는 추론과 학습방법에 대해 설명하고, 추론과 학습이 에이전트들에 어떻게 적용되는지를 살펴보겠다. 5장에서는 시스템을 구성하는 에이전트들 간의 주요 작업에 대한 KQML 메시지 흐름에 대해 소개한다. 6장에서는 시스템의 구현과 실험 내용에 대하여 알아보고, 마지막으로 7장에서는 연구 내용을 요약하고 앞으로의 연구 방향을 제시한다.

## 2. 시스템 구조

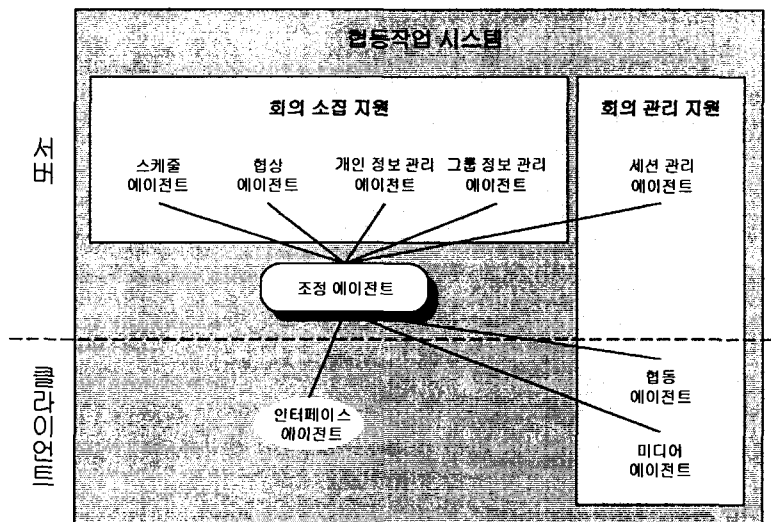
이 시스템은 그룹 내의 효율적인 협동 작업을 위해 소집부터 회의 관리까지 수행해준다. 전체 시스템의 구조는 <그림 1>과 같으며, 여러 에이전트가 하나의 시스템을 구성하는 다중 에이전트 구조를 가진다. 이 에이전트들 중 몇몇 에이전트들이 모여 회의 소집 지원, 회의 관리 지원을 위한 기능을 수행한다. 또한, 회의 소집을 지원하는 에이전트들 중에서는 참여자들이 회의에 참석할 가능성이 높은 날짜를 정하는데 도움을 줄 수 있도록 개인 일정 관리와 그룹 일정 관리를 지원하기도 한다. 그리고, 개인 일정 관리를 위해서 부가적으로 사용자 등록과 인증이 따른다.

이 에이전트들은 클라이언트 측과 서버 측에 분산되어 있으며, 서버 측에서는 스케줄 에이전트, 협상 에이전트, 개인정보 관리 에이전트, 협상 에이전트, 개인 정보 관리 에이전트, 그룹 정보 관리 에이전트

(personal information managing agent), 그룹 정보 관리 에이전트(group information managing agent), 세션관리 에이전트(session managing agent), 그리고 조정 에이전트 (coordinating agent)가 활동한다. 클라이언트 측에서는 인터페이스 에이전트 (interface agent), 미디어 에이전트(media agent), 그리고 협동 에이전트(collaborating agent)가 활동한다. 이 장에서는 시스템에서 사용되는 에이전트들과 몇몇 에이전트들에 의해 지원되는 작업에 대해 살펴보고 하겠다.

### 2.1 조정 에이전트(Coordination Agent)

조정 에이전트는 서버 측에서 활동하며, 시스템 내의 모든 에이전트의 위치 정보, 각 에이전트의 처리능력 등의 정보를 가지고 각각의 응용 에이전트를 연결해주는 네임 서버 역할을 한다. 조정 에이전트를 통해 구성되는 에



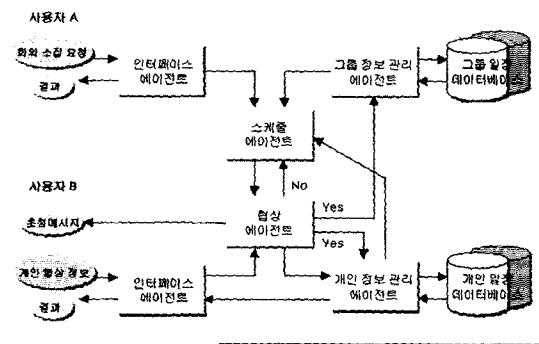
<그림 1> 전체 시스템 구성도

이전트들 간의 통신 구조에 대해서는 5장에서 설명하겠다.

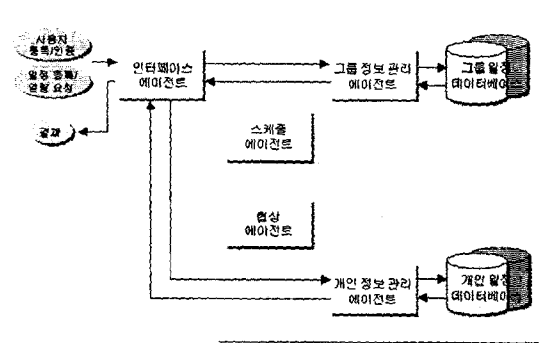
## 2.2 회의 소집 지원을 위한 에이전트 그룹

회의 소집 지원을 위한 에이전트 그룹은 그룹 정보 관리 에이전트, 스케줄 에이전트, 협상 에이전트, 개인 정보 관리 에이전트들이다. 개인의 일정은 물론 사용자 그룹의 일정을 관리함으로써 회의를 위한 준비 작업(참여자 초청, 참여자의 참가 가능 여부, 회의의 성사 여부 등)을 쉽게 해준다. 개인 일정 관리를 위해서는 부가적인 사용자 등록을 해야 하며, 사용자 그룹 일정 관리를 위해서는 해당 사용자 그룹에 가입해야 한다. 또한 매번 서버에 접속할 때마다 사용자 인증을 거쳐야 한다. 사용자 등록, 사용자 인증, 개인 일정 관리, 그리고 사용자 그룹 일정 관리는 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트가 회의 소집 지원과 함께 병행한다. 이렇게 개인 일정과 그룹 일정에서 획득된 정보는 사용자들이 많이 참여할 수 있는 회의 소집 날짜를 추론하는데 유용한 정보로 사용된다.

<그림 2>는 회의 소집 제안 및 협상 과정을 나타낸다. 회의 소집 제안은 그룹의 장(chief)이 그룹의 구성원들과 회의를 원할 때 이루어지며, 회의 소집 정보(희망 회의 소집 날짜 리스트, 참여자 리스트, 회의에 필요한 참여자 수, 회의 내용 등)와 함께 인터페이스 에이전트를 거쳐 스케줄 에이전트에 보내진다. 스케줄 에이전트는 회의 소집 정보를 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트에 보내 회의 소집 가능 날짜 리스트를 얻어온다.



<그림 2> 회의 소집 제안 및 협상



<그림 3> 사용자 등록/인증과 개인/그룹 일정 열람

스케줄 에이전트는 두 가지 에이전트로부터 얻어진 다양한 결과들을 가지고 회의 소집 정책에 부합되는 날짜 중에서 그룹 구성원들이 가장 많이 참여할 수 있는 회의 가능 날짜를 추론한다. 이 정보는 회의 소집 정보와 함께 협상 에이전트에 보내지며, 협상 에이전트는 스케줄 에이전트가 추론한 날짜와 회의 소집 정보를 개인 정보 관리 에이전트와 사용자에게 보낸다. 다른 사용자들은 메일을 확인하거나 일정관리 응용 프로그램을 이용하여 서버에 접속하여 회의 소집 요청을 확인한 후, 자

신이 참여 가능한 날짜와 함께 회의 참가 여부를 결정한다. 이 정보는 협상 에이전트에 보내지고, 협상 에이전트는 사용자가 보내온 정보를 가지고 정해진 정책을 통해 회의 소집 가능 여부를 결정한다. 회의 소집이 가능하면 개인 정보 에이전트와 그룹 정보 관리 에이전트에 회의 소집 정보를 보내어 일정을 갱신시키고, 그렇지 않으면 회의 소집을 요청한 그룹의 장에게 회의 소집 불가 메시지를 메일을 통해 보낸다.

<그림 3>은 사용자 등록, 사용자 인증 및 개인일정 열람과 그룹 일정 열람을 나타낸다. 사용자 등록은 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트에 의해 처리되고, 사용자 인증과 그룹 일정 등록 및 열람은 그룹 정보 관리 에이전트에 의해 처리된다. 또한, 개인 일정 등록 및 열람은 개인 정보 관리 에이전트에 의해 처리된다.

다음은 각 에이전트의 기능에 대한 설명이다.

#### • 개인 정보 관리 에이전트

개인 정보 관리 에이전트는 인터페이스 에이전트와 함께 그룹에 구성원이 새로 가입될 때마다 생성이 되며, 개인 일정 정보와 사용자 프로파일(profile) 정보를 관리하고 이를 필요로 하는 다른 에이전트에게 제공해 준다. 예를 들어, 사용자의 개인 일정 열람 요구를 받은 인터페이스 에이전트가 개인 정보 관리 에이전트에게 개인 일정 열람을 요구할 경우에 응답한다. 그리고, 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 보내온 회의 소집 정보를 받아서 개인 정보 및 일정관련 지식 베이스와 비교해 개인의 일정 정보를 스케줄

에이전트에게 전달한다. 또한, 회의 소집에 대한 정보가 생성되거나 갱신되었을 경우 갱신된 정보를 인터페이스 에이전트에게 전달하며, 사용자를 대신한 회의 소집 요청에 대한 응답도 수행한다.

#### • 그룹 정보 관리 에이전트

그룹 정보 관리 에이전트는 새로운 그룹이 등록될 때마다 생성되며, 그룹의 일정 정보와 사용자의 인증 정보와 사용자가 속해 있는 그룹 정보를 관리하며 이 정보를 필요로 하는 다른 에이전트에게 제공해 준다. 예를 들어, 등록된 사용자인가를 결정하는 사용자 인증의 경우, 인터페이스 에이전트가 사용자의 그룹 일정 열람을 요구하는 경우, 또는 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 필요한 그룹 정보를 요구한 경우 등에 응답한다. 그리고, 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 보내온 회의 소집 정보를 그룹 정보 및 일정관리 지식 베이스의 내용과 비교해서 그룹의 일정 정보를 스케줄 에이전트에게 전달한다. 또한, 그룹 정보 관리 에이전트는 협상 에이전트로부터 성사된 회의 정보를 받아 데이터 베이스에 저장한 후, 세션 관리 에이전트에게 전송한다.

#### • 스케줄 에이전트

스케줄 에이전트는 인터페이스 에이전트로부터 전달된 회의 소집 정보를 바탕으로 하여 참여자들의 개인 일정 정보와 그룹 일정 정보를 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트에게서 얻어 희망 회의 소집 날짜 리스트에 정의된 기간 중 그룹 구성원들이 가장

많이 참여할 수 있는 회의 가능 날짜와 시간을 추론한다. 이 추론된 정보는 협상 에이전트에 보내진다.

- 협상 에이전트

스케줄 에이전트로부터 전달된 회의 소집 날짜와 회의 참가자 명단을 받아서 해당 참가자에게 회의 소집 날짜와 시간 및 회의 내용을 전달한다. 그 다음에 참가자들의 회의 참석 여부에 대한 응답을 받아 회의 소집의 성사 여부를 결정한다. 참가자들의 응답으로 회의 참석 가능 여부 및 주어진 기간 내에 자신의 회의 가능한 날짜들이 같이 보내 질 수 있다. 회의 소집 요청 시 그룹의 장이 정한 정책(예를 들어, 회의 소집 조건이 구성원의 과반수이상)에 따라 회의가 가능한지 불가능한지를 판단한다. 회의가 성사되면 협상 에이전트는 각 그룹 구성원에게 성립된 회의 일정을 메일로 알리고, 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트를 통해 개인 일정 데이터베이스와 그룹 일정 데이터 베이스를 갱신한다. 반대로, 회의가 성사되지 못하면 회의 소집 요청을 한 그룹의 장에게 회의 소집이 불가능함을 메일로 알린다.

### 2.3 인터페이스 에이전트

인터페이스 에이전트는 개인 정보 관리 에이전트와 함께 그룹에 구성원이 새로 가입될 때마다 생성이 되며, 사용자의 작업을 편리하게 도와주는 일을 한다. 인터페이스 에이전트는 응용프로그램이 어떤 기능을 수행하는데 있어서 필요한 정보를 제공해주기 위해 사용

자와 협상을 한다. 중요한 작업은 사용자에게 정보를 제공하길 요구하고, 그렇지 않으면 인터페이스 에이전트는 작업의 중요도에 따라 미리 준비된 작업을 순차적으로 수행한다. 이런 인터페이스 에이전트의 기능은 사용자의 기호와 수행되어졌던 작업에 대한 지식을 바탕으로 한 학습을 통해 이루어진다.

### 2.4 회의 관리 지원 에이전트 그룹

회의 관리를 지원해 주는 에이전트들은 세션 관리 에이전트, 협동 에이전트, 미디어 에이전트로 구성되며, 협동작업을 효율적으로 할 수 있도록 회의 준비와 회의 관리 작업을 수행한다. 회의 준비는 회의 소집이 이루어진 후에 회의로 넘어가는 시점에서 일어나는 작업이다. 회의 관리는 회의 참가자, 회의 이름, 회의 시작 시간, 회의에 관한 방침 등과 같은 공유되는 상태 정보를 제어하는 작업이다. 따라서, 이 작업에는 회의 설정, 회의 종료, 그리고 참여 및 탈퇴 기능이 수반된다. <그림 3.4>에서 보는 바와 같이 세션 관리 에이전트는 서버에 위치하며 클라이언트에 위치한 협동 에이전트와 상호작용을 한다. 여기서, 세션 관리 에이전트는 전체 회의 정보를 관리하며, 협동 에이전트는 현재 참여중인 회의 정보를 관리한다. 이것은 회의 중에 서버의 부하를 줄이기 위함이다.

- 세션 관리 에이전트

세션 관리 에이전트는 서버에만 존재하며, 회의 시작 전후로 하여 모든 회의의 참가자, 회의 이름, 회의 시작 시간, 회의에 관한 방침 등과 같은 공유되는 상태 정보를 제어하는 일

을 한다. 3.2 절에서 살펴본 그룹 정보 관리 에이전트에 의해 확정된 회의 정보가 전달되면 해당 회의를 개설해 준다. 세션 관리 에이전트는 모든 협동 작업 회의 정보를 유지하며 아래와 같은 작업을 담당한다

- 회의 소집 협상을 통하여 예약된 회의 소집이나 클라이언트 측 협동 에이전트의 회의 소집 요청에 대한 회의 소집
- 회의 설정이나 변경 및 종료 등의 회의 제어 관리
- 원하는 회의와 연결
- 참가자 인증
- 구성원 지위 제어 관리
- 전체 참여자 회의그룹 정보 관리

• 미디어 에이전트

미디어 에이전트는 갱신된 데이터를 모든 다른 참가자들에게 전송하며 아래와 같은 일을 담당한다.

- 그룹 통신 구조 정보 유지
- 다른 미디어 에이전트 사이의 데이터 통신에 있어서 미디어 제어 관리

• 협동 에이전트

협동 에이전트는 각 협동 작업 참여자가 능동적으로 회의에 참여할 수 있게 하는 일을 담당하며 모든 클라이언트 사이트마다 존재한다. 협동 에이전트는 현재 참여 중인 협동 작업 회의 정보를 유지하며 아래와 같은 작업을 지원한다.

- 회의 소집 요청
- 회의 소집에 대한 승낙 여부 처리
- 각 사이트에서 발생한 변경 사항에 대한 멀

티캐스트 요청

- 개인 참여자 회의그룹 정보 관리
- 응용 프로그램으로부터의 회의 정보 참조 등의 요구 처리

### 3. 시스템 모델

이 장에서는 본 시스템에 대한 선명한 이해를 위하여 본 시스템의 핵심이 되는 모델을 정의해 보고자 한다. 회의 소집 시스템은 여러 에이전트들과 여러 회의로 구성되며 하나의 회의는 참여자들의 그룹으로 구성된다. 주어진  $n$  개의 회의와  $g$  개의 그룹 및  $k$  참여자가 있을 때, 하나의 회의 소집 모델은 다음과 같은 3-튜플로 표현될 수 있다.

$$S = (A, C, P)$$

여기서,

$A = \{1, 2, \dots, k\}$ 의 개인 정보관리 에이전트와  $\{1, 2, \dots, g\}$ 의 그룹정보 관리에이전트로 구성된 에이전트(agent)들의 집합이고,

$C = \{c_1, c_2, \dots, c_n\}$ 은 소집되어야 할 회의(confERENCE)들의 집합이고,

$P = \{1, 2, \dots, m\}$ 는 참여자(participant)들의 집합이다,

또한, 하나의 회의는 아래와 같은 튜플로 표현될 수 있다.

$$c = \{ title, date, time, duration, period, policy, participant \}$$

여기서,

*title* : 회의의 제목(title)을 기술하며,



*date* : 회의가 소집될 것 희망하는 날짜(date)로서, 희망 날짜 및 기간을 토대로 회의 소집일을 산출하게 된다.

*time* : 회의 소집자가 희망하는 회의 시작시간(time)이며,

*duration* : 회의의 예정시간(duration)으로, 회의 소집자는 회의가 얼마동안의 시간동안 진행될 것인지에 대한 정보를 명시하게 된다.

*period* : 기간(period)으로서, 회의가 소집되어야 하는 기간을 부여하여 그 기간내의 각 사용자 및 사용자 그룹의 일정을 참고로 한다.

*policy* : 회의 소집 시 적용되어야 할 정책(policy)이며, 1/2, 2/3, 1/1등과 같은 값들 중 하나의 값으로 표현된다.

*participant* : 특정 회의에 참석해야 하는 개인 참여자(participant) 및 그룹(group)의 집합이다.

본 시스템의 스케줄 에이전트는 이러한 회의정보를 토대로 개인 정보관리 에이전트와 그룹정보관리 에이전트의 일정을 통해 최적의 회의소집 일자 및 시간을 추론하게 된다. 그리고 추론된 일자 및 시간은 협상에이전트로 전달되어 각 참여자와의 협상(negotiation)과정을 수행하게 된다.

## 4. 추론과 학습의 적용

이 논문에서 제안하는 시스템은 추론을 위해 실시간으로 빠른 회의 소집 결과를 도출해 낼 수 있는 전진 연결(forward chaining)

(Joseph & Jennifer, 1998)을 사용하였다. 학습을 위해서는 교수 학습을 위한 가장 인기 있는 신경망 중에 하나인 역전파 네트워크(back propagation network) (Perlmutter, 1995)를 사용하였다. 이 장에서는 우리가 제안하는 시스템에서 사용한 추론과 학습 방법에 대해 간단히 살펴본 다음, 어떤 에이전트들에게 추론과 학습 방법이 적용되었으며 각 에이전트들이 어떤 임무를 수행하게 되는지에 대해 살펴보고 하겠다.

### 4.1 추론과 학습 방법

#### 4.1.1 전진 연결을 사용하는 추론

If-then 규칙은 인공 지능 응용들에서 선언적 지식 표현의 가장 인기 있는 형식으로 사용되어왔다. If-then으로 표현되어진 지식은 대부분의 사람들이 쉽게 이해하고 그 규칙들을 쉽게 읽을 수 있다. 각 규칙은 지식 베이스에서 독립형 지식의 조각 또는 정보의 단위로 보여질 수 있으며 새로운 지식은 쉽게 추가될 수 있다. 또한, 보유하고 있는 지식은 개인적인 규칙들을 생성하고 변경함으로써 단순하게 변화될 수 있다.

전진 연결은 어떤 규칙의 집합이 초기 데이터 집합으로부터 새로운 사실들을 유도하기 위해 사용되는 데이터 유도형 추론 처리이다. 전진 연결 알고리즘은 단순하고 간단한 응용 또는 규칙들의 점화에 의해 새로운 데이터를 생성하는 추론 절차로서 전진 연결은 매우 빠르다. 전진 연결을 응용한 초창기 전문가 시스템들은 R1/XCON을 포함한다. 어떤 전문가 시

시스템은 Digital Equipment Corporation's VAX 컴퓨터를 위한 구성들을 생성하기 위해 사용되어졌다. PROSPECTOR는 예상 데이터로부터 광상(mineral deposits)의 위치를 예측하기 위해 사용되어진 시스템이다. 또한, 전진 연결은 빠른 식별과 빠른 응답이 요구되는 실시간 모니터링과 진단 시스템에서 많이 사용되어졌다.

• 전진 연결 추론 사이클

추론을 위해서는 세 가지 기본 요소들이 필요하다. 규칙들과 사실들로 구성된 지식 베이스, 추론하는 동안 데이터를 저장하기 위한 작업 메모리, 그리고 추론 엔진이 그것이다. 다음은 전진 연결 추론 사이클이다.

- ① 추론 엔진에 규칙 베이스를 로드하고, 지식 베이스에 저장된 지식들을 작업 메모리로 로드한다.
- ② 부가적인 초기 데이터를 작업 메모리에 추가한다.
- ③ 전건(antecedent) 절의 규칙들을 작업 메모리에 있는 데이터와 비교하여 일치시켜서 발생된 규칙들을 결정한다. 비교 일치 단계에서 전건 절의 전부가 참이면 후건(consequent) 절 또한 참이므로 전건과 후건을 포함하는 이 절은 새로운 규칙으로 받아들여진다. 이렇게 발생된 규칙들의 집합을 충돌 집합(conflict set)이라고 한다
- ④ 충돌 집합으로부터 단일 규칙을 선택하기 위해서 본 절차 다음에 설명되는 충돌 비교 함수(conflict resolution) 절차를 사용한다.
- ⑤ 후건 절을 평가함으로써 선택했던 규칙을 점화(fire)한다. 만약 그것이 사실생성

(fact-generation) 규칙이라면 작업 메모리를 갱신한다. 만약 그것이 행동 규칙이라면 효과기(effector) 함수를 호출한다. 이것을 실행(act) 단계라고 한다.

- ⑥ 충돌 집합이 공집합이 될 때까지 3, 4, 그리고 5 단계를 반복한다.

• 충돌 비교 함수 단계

비교 일치 단계가 완성되고 충돌 집합이 생기는 즉시, 충돌 비교 함수 단계로 옮겨간다. 충돌 비교 함수는 전진 연결 추론 시스템의 행동 관점에서 보면 가장 중요한 단계이다. 충돌 집합이 비어있거나 단지 하나의 규칙을 포함하고 있을 때는 문제가 간단하다. 그러나, 많은 경우에 발생하는 규칙이 하나 이상일 것이다. 여기서 어떤 규칙을 선택해서 점화해야 하는가에 대한 몇 가지 유용한 대안이 있다.

- 충돌 집합에서 첫 번째 규칙을 선택한다.
- 가장 특이성을 가지는 규칙을 선택하거나 많은 수의 전건 절을 가지는 규칙을 선택한다. 이것은 규칙의 한 부분으로서 가장 많은 테스트 상태를 가지는 가장 특수화된 규칙을 선택하는 것이다.
- 가장 최근에 변화된 데이터를 참조하는 규칙을 선택한다. 이 방법은 타임스탬프를 붙이거나 어떻게든지 규칙들이 마지막으로 변경되어졌을 때 보여주기 위해 꼬리표를 붙여서 작업 메모리의 변화를 표시하는 방법을 필요로 한다.
- 만약 규칙이 이전 사이클에서 점화되었다면, 그것을 충돌 집합에 더하지 않는다. 점화된 규칙은 단지 한번만 점화될 수 있다.

본 연구에서는 비교 흡수 방법으로는 첫 번째와 네 번째 대안인 가장 많은 수의 전진 절을 가지는 규칙을 선택하고, 규칙이 이전 사이클에서 점화되었을 경우, 그것을 충돌 집합에 더하지 않는 방법을 사용하였다.

#### 4.1.2 역전파 네트워크를 사용하는 학습

역전파 네트워크는 교수 학습을 위한 가장 인기 있는 신경망 구조이다. 이것은 네트워크를 통하여 단 방향으로 데이터가 흐르는 순방향 연결 토폴로지(feedforward connection topology)이다. 그리고, 오류 역전파(error backward propagation)라고 불리는 기술을 사용한다. 오류 역전파 네트워크는 입력과 출력 장치들의 계층 외에 부가적으로 하나 이상의 은닉 장치들(hidden units)의 계층을 가질 수 있다. 이 은닉 계층에 존재하는 장치들은 단지 외부적인 환경이 아닌 내부에 구성된 입력 장치나 출력 장치로부터 입력을 받는다. 역전파 네트워크는 처리 장치들의 은닉 단일 계층과 함께 이 은닉 계층에 충분한 장치를 부여했을 때, 연속적인 함수를 구성하여 학습할 수 있다. 역전파 네트워크의 중요한 응용은 예측과 분류이다.

본 시스템에서는 사용자의 관심도와 중요도를 학습하기 위해서 키워드 필터(keyword filter)와 피드백 필터(feedback filter)를 사용한다. 키워드 필터는 정해진 키워드들과 각 키워드의 정해진 값을 가지고 주어진 시간에 나타난 키워드의 횟수와 그 키워드의 가중치(weight)를 곱한다. 이렇게 각 출현된 키워드마다 얻어진 값을 모두 더한 값을 결과로 사용한다. 결

과 값이 낮을수록 사용자가 한가하다는 것을 표시한다.

피드백 필터는 역전파 네트워크의 학습을 위해 주어진 시간대의 일정을 입력 값으로 하고 사용자의 피드백 정보를 출력 값으로 사용한다. 피드백 필터는 초기에 키워드 필터를 사용하여 얻어진 결과 값에 임의로 정한 피드백 값을 부여하여 인터페이스 에이전트가 역전파 네트워크를 통해 학습을 하도록 하며, 그 다음부터는 이 학습된 역전파 네트워크를 사용하여 사용자에게 결과를 제시한다. 이것이 좀더 학습을 필요로 할 경우에는 다시 평가를 해서 나온 값을 사용자가 피드백 정보로서 인터페이스 에이전트에게 전달하여 다시 학습을 하도록 한다. 이 학습 과정은 사용자의 공통적인 선호도와 중요도를 반영하는 것이다. 만약, 키워드 필터가 많은 사용자의 공통적인 선호도와 중요도를 반영하여 구성되었다면 초기에 키워드 필터를 사용한 후에는 학습하는 반복 회수가 적게 일어날 것이다.

본 연구에서는 사용자의 선호도와 중요도를 표시하기 위해 사용하는 회의 일정의 상태는 다섯 가지 타입으로 정의하였다. 이 타입은 <Vacant>, <Insignificant>, <Neutral>, <A Little Significant>, <Significant>이다. 각각 순서대로 1.0, 0.75, 0.5, 0.25, 0.0의 feedback 값을 가지며, 이것은 인터페이스 에이전트에게 전달되어 학습을 위한 피드백 값으로 사용된다.

본 논문에서 키워드 필터를 위한 키워드로는 다음과 같은 것을 사용한다.

- STime(Specific-Time)

- GSTime (Group-Specific-Time)
- PSTime(Private-Specific-Time)
- NSTime(Non-Specific-Time)
  - GNSTime(Group-Non-Specific-Time)
  - PNSTime(Private-Non-Specific-Time)
- UTime(Uncertain-Time)
  - GUTime(Group-Uncertain-Time)
  - PUTime(Private-Uncertain-Time)
- VTime(Vacant-Time)
  - GVTime(Group-Vacant-Time)
  - PVTime(Private-Vacant-Time)

이것은 약속의 종류에 따라 분류된 것이다. STime은 날짜와 시간이 정해진 일정을 나타낸다. 예를 들어, 개인의 경우에는 다른 사람과의 약속이 여기에 해당되고, 그룹의 경우에는 임시 회의나 정기 모임이 여기에 해당된다. NSTime은 약속 날짜는 정해졌으나 시간이 명확히 정해져 있지 않은 약속을 나타낸다. 이것은 기념일 등으로 하루 중에 어떤 시간대에 약속이 있다는 것을 나타낸다. UTime같은 경우는 약속일과 약속 시간을 정확히 정하지 않은 약속이다. 각 약속의 중요도에 따라 GSTime = 8, PSTime = 8, GNSTime = 3, PNSTime = 3, GUTime = 1, PUTime = 1, GVTime = 0, PVTime = 0으로 해당 값을 부여했다.

#### 4.2 추론과 학습을 하는 에이전트

이 논문에서 소개하는 몇몇 에이전트들은 추론과 학습을 수행한다. 추론은 스케줄러 에이전트와 협상에이전트가 수행하며, 학습은 개

인 정보 관리 에이전트, 인터페이스 에이전트 가서 수행한다.

##### 4.2.1 스케줄 에이전트

스케줄러 에이전트는 참여자 리스트에 존재하는 참여자들의 일정 중에 많은 참여자가 참석할 수 있는 날짜를 선택하는 작업을 수행한다. 또한, 회의가 불가능할 경우 회의를 요청한 그룹의 장에게 회의 불가능 메시지를 전달한다.

스케줄러 에이전트는 모든 사용자들의 특정 기간 동안의 일정을 그룹 일정 관리 에이전트와 개인 일정 관리 에이전트를 통해 받은 다음, 그 기간 안의 각 날짜마다 회의 소집 대상인 각 사용자의 일정이 비어 있는지 조사한다. 스케줄러 에이전트는 전진 연결을 사용하여 모든 사용자들의 일정이 비어있는 회의 가능한 날짜가 있는지를 판단한다. 회의 가능한 날짜가 있을 경우, 스케줄러 에이전트는 그 날짜를 협상 에이전트에 보내기 위한 효과기(effecter) 함수를 호출하고, 반대로, 회의 가능 날짜가 없을 경우에는, 키워드 필터를 사용하기 위한 효과기 함수를 호출한다. 회의가 가능한 날이 여럿 있을 때는 경우에는 효과기 함수 내부에서 그중 가장 빠른 날짜를 선택하여 협상에이전트에게 보낸다. <그림 4>는 이를 위한 규칙 베이스를 나타낸다.

회의 소집을 위한 빈 일정이 존재하지 않을 경우에는 키워드 필터를 사용하여 가장 가능성 많은 날을 선택한다. 키워드 필터를 사용하여 나온 결과 값을 총 참여자 수로 나누어서 임계값 (여기서는 4) 이하인 값들 중 가장 작

```

Rule1 : IF(numVacantSchedule > 0)
        THEN isVacant = yes
Rule2 : IF(numVacantSchedule == 0)
        THEN isVacant = no
Rule3 : IF(isVacant == yes)
        THEN Effector(negotiationStep)
Rule4 : IF(isVacant == yes)
        THEN Effector(keywordfilterStep)

```

<그림 4> 스케줄러 에이전트에서 사용되는 규칙 베이스

은 값을 선택한다. 같은 값이 존재할 경우에는 가장 빠른 날을 선택한다. 여기서 임계값 4 이하는 일정한 상태가 <Vacant>, <Insignificant>, <Neutral>을 포함하는 값이다. 임계값 4 이상일 경우는 스케줄러 에이전트는 회의 소집 불가능 메시지를 협상 에이전트에게 보내고, 협상 에이전트는 받은 결과를 메일을 이용하여 사용자에게 보낸다.

#### 4.2.2 협상 에이전트

협상 에이전트는 인터페이스 에이전트를 통해 참여자들의 회의 참석 여부의 응답을 받아 주어진 정책에 따라 회의 소집 성사 여부를 결정하는 일을 수행한다. 인터페이스 에이전트로부터 받은 다양하고 중복된 데이터로부터 회의 가능한 시간들을 산출하기 위해서 그룹의 장은 회의 소집을 위해 정의한 회의 소집 정책을 사용한다. 이 정책은 회의 참여 가능자가 회의 소집 요청을 받은 전체 사용자의 얼마의 비율을 가지는지를 나타내며, 여기서는 회의 소집 요청을 받은 사용자의 1/2, 2/3, 1/1을 정

책의 값으로 사용한다.

정책이 주어졌을 때 협상 에이전트는 주어진 기간 내에서 정책에 맞는 날들을 선별해 낸다. 이때 선택된 날이 정책에 맞는가를 판단하기 위해서 전진 연결을 사용한다. <그림 5>는 이를 위한 규칙 베이스를 나타낸다.

```

Rule1 : IF(participationRate > 1/2)
        THEN MeetingType = A
Rule2 : IF(participationRate > 2/3)
        THEN MeetingType = B
Rule3 : IF(participationRate == 1)
        THEN MeetingType = C
Rule4 : IF(MeetingType == A)
        THEN satisfiedMeetingType = A
Rule5 : IF(MeetingType == A)
        AND(MeetingType == B)
        THEN satisfiedMeetingType = B
Rule6 : IF(MeetingType == A)
        AND(MeetingType == B)
        AND(MeetingType == C)
        THEN satisfiedMeetingType = C

```

<그림 5> 협상 에이전트에서 사용되는 규칙 베이스

정책에 맞는 날짜를 선별해낸 후, 선별된 날짜 중에서 가장 참여도가 높은 날짜를 선택한다. 참여도가 같은 날짜가 존재한다면 그 중에서 가장 빠른 날짜를 선택한다. 주어진 정책에 맞는 날이 없을 경우 회의 소집을 요청한 그룹의 장(chief)에게 회의 소집이 이루어지지 않았음을 메일을 이용해 알린다.

#### 4.2.3 개인 정보 관리 에이전트

개인 정보 관리 에이전트는 임의의 사용자

가 부재중이거나 회의 소집 요청 메시지를 확인하지 못하여 정해진 기간 내에 회의 참가 여부를 전달 할 수 없을 경우에 사용자를 대신해서 응답을 해준다. 개인정보관리 에이전트는 인터페이스 에이전트에서 학습된 사용자 프로파일(profile)로 역전파 네트워크를 구성한 후, 그룹 정보 관리 에이전트로부터 해당 사용자가 포함된 그룹의 일정 정보와 개인 일정 정보를 얻어온다. 개인정보 관리 에이전트는 이 정보들로부터 회의 가능일정에 대한 예측 데이터를 마련하고 그 중에서 가장 높은 수치의 데이터들로 응답을 한다. 이 예측 데이터 중에서 가장 높은 수치의 데이터는 그 동안 학습된 사용자의 선호도와 중요도를 반영한 데이터로서 일정이 비어있음을 나타낸다.

#### 4.2.4 인터페이스 에이전트

인터페이스 에이전트는 사용자의 피드백 정보를 이용하여 사용자의 선호도와 중요도를 학습한다. 회의 소집 요청을 받고 회의 참가 여부를 결정하기 위하여 사용자는 인터페이스 에이전트가 제시하는 날짜별 일정 리스트를 보고 빈 날짜를 정한 후 참가 여부를 결정한다. 제시된 날짜별 일정 리스트는 사용자의 선호도와 중요도를 반영하여 한가한 일정부터 바쁜 일정의 순서로 정렬되어있으며 결과가 만족스럽지 못할 경우에는 사용자에게 의해 피드백 정보를 부여한 후 다시 학습을 시킨다. 인터페이스 에이전트는 각 그룹별 사용자 프로파일을 관리해 준다.

## 5. KQML의 적용

외부와의 정보 교환 시 누구에게 무엇을 요구해야 하는가에 대한 문제가 발생하게 된다. 즉, 어떤 정보를 어떤 에이전트로부터 얻을 수 있는가 하는 문제이다. 전체 시스템에서 이루어지는 작업을 두 범주로 나누면 회의 소집 지원을 위한 작업과 회의 관리 지원을 위한 작업으로 나눌 수 있는데, 이 장에서는 이 중 회의 소집 지원 메커니즘 중에서 가장 중요한 작업들인 회의 소집 제안과 회의 소집 협상을 위한 KQML 메시지의 흐름에 대해서만 다루도록 하겠다. 이 시스템의 기타 여러 작업에 대한 KQML 메시지 흐름 및 이에 대한 자세한 설명은 참고 문헌 (Joseph & Jennifer, 1998)을 참조하기 바란다.

이 논문에서는 KQML 메시지를 표현하는데 있어서 예약된 수행문들(reserved performatives)과 VKB(Virtual Knowledge Base)의 개략적인 내용만을 포함하여 표현하였을 뿐, 수행문 전체 문장과 VKB의 세부적인 내용에 대해서는 명세하지 않았다.

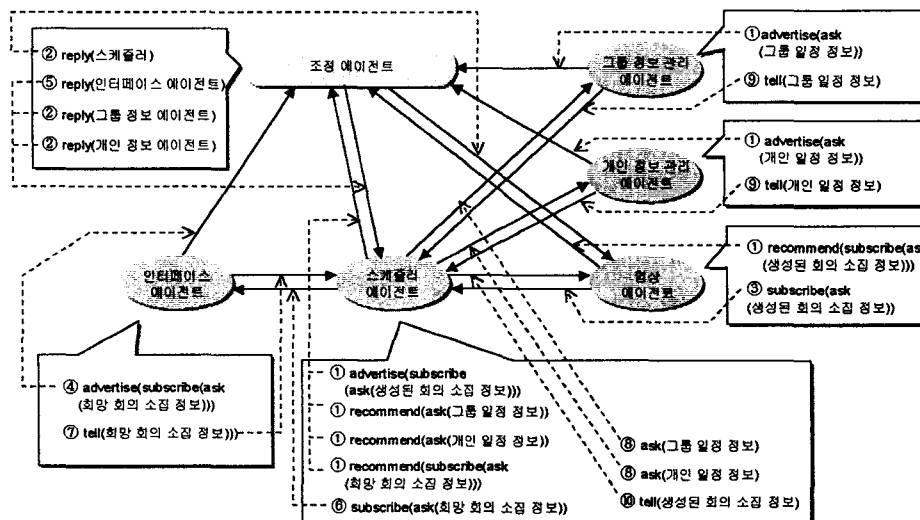
### 5.1 회의 소집 제안을 위한 KQML 메시지 흐름

회의 소집 제안은 어떤 사용자가 회의를 개최하기를 원할 경우에 이루어지는 작업이며, KQML 메시지의 흐름은 <그림 6>과 같다. 우선, 그룹의 우두머리가 원하는 회의 기간을 정한다. 이 정보는 인터페이스 에이전트를 통해 스케줄러 에이전트에 전달되고 스케줄러 에이전트는 해당 그룹의 일정과 구성원들 개인의

일정을 얻어와서 알맞은 날짜를 추측해 낸다. 이 결과는 협상 에이전트에 보내져 모든 사용자들에게 메일로 전송된다.

- ① 서버 측에 존재하는 스케줄 에이전트, 협상 에이전트, 그룹 정보 관리 에이전트 그리고 개인 정보 관리 에이전트는 조정 에이전트에게 다음과 같은 사실을 요청하거나 알린다.
  - 스케줄 에이전트 : <생성된 회의 소집 정보>가 갱신될 때마다 그 정보를 알려줄 수 있다고 알린다. <그룹 일정 정보>와 <개인 일정 정보>를 알려줄 수 있는 에이전트의 정보를 요청한다. <희망 회의 소집 정보>가 갱신될 때마다 알려줄 수 있는 에이전트의 정보를 요청한다.
  - 협상 에이전트 : <생성된 회의 소집 정보>가 갱신될 때마다 알려줄 수 있는 에이전트의 정보를 요청한다.

- 그룹 일정 관리 에이전트 : <그룹 일정 정보>를 알려줄 수 있다고 알린다.
- 개인 일정 관리 에이전트 : <개인 일정 정보>를 알려줄 수 있다고 알린다.
- ② 조정 에이전트는 스케줄 에이전트와 협상 에이전트에게 그들이 필요로 하는 정보를 가지고 있는 에이전트에 대한 정보를 넘겨준다.
- ③ 협상 에이전트가 스케줄 에이전트에게 <생성된 회의 소집 정보>가 갱신될 때마다 알려달라고 요청한다.
- ④ 인터페이스 에이전트가 <희망 회의 소집 정보>가 갱신될 때마다 알려줄 수 있다고 알린다.
- ⑤ 조정 에이전트는 스케줄 에이전트에게 <희망 회의 소집 정보>를 알려줄 수 있는 에이전트가 인터페이스 에이전트라고 알려준다.
- ⑥ 스케줄 에이전트가 인터페이스 에이전트에

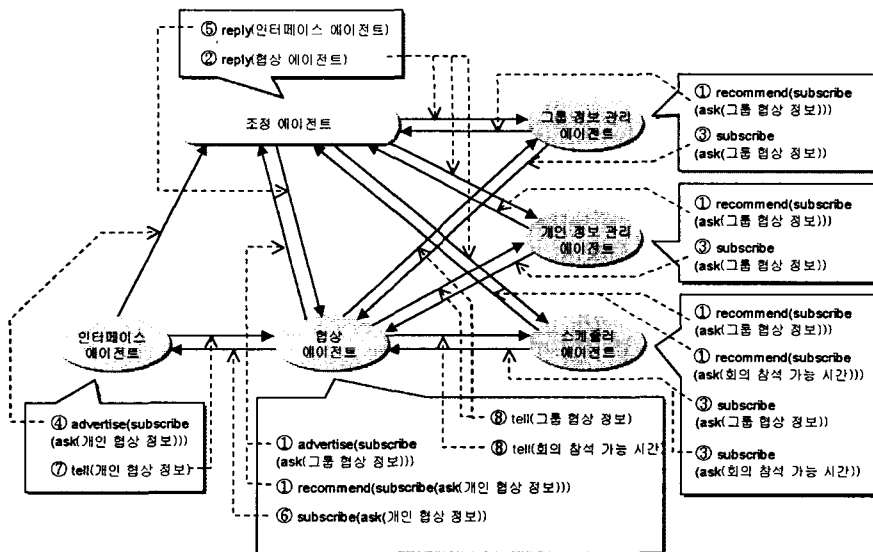


<그림 6> 회의 소집 제안을 위한 KQML 메시지의 흐름

- 계 <희망 회의 소집 정보>를 갱신될 때마다 알려달라고 요청한다.
- ⑦ 인터페이스 에이전트는 스케줄 에이전트에게 <희망 회의 소집 정보>가 갱신될 때마다 알려준다.
  - ⑧ 스케줄 에이전트는 그룹 일정 관리 에이전트에게 <그룹 일정 정보>, 개인 일정 관리 에이전트에게 <개인 일정 정보>를 알려달라고 요청한다.
  - ⑨ 그룹 일정 관리 에이전트는 <그룹 일정 정보>, 개인 일정 관리 에이전트는 <개인 일정 정보>를 스케줄 에이전트에게 알려준다.
  - ⑩ 스케줄 에이전트는 앞에서 얻은 <희망 회의 소집 정보>, <그룹 일정 정보> 그리고 <개인 일정 정보>를 가지고 생성한 <생성된 회의 소집 정보>가 갱신될 때마다 협상 에이전트에게 알려준다.

### 5.2 회의 소집 협상을 위한 KQML 메시지 흐름

회의 소집 협상은 회의 소집 제안 다음에 이루어지는 작업이며, 이 작업을 위한 KQML 메시지의 흐름은 <그림 7>과 같다. 스케줄러에 의해 추측된 회의 소집 날짜를 그룹 구성원에게 전달한 후, 그룹 구성원들은 인터페이스 에이전트를 통해 투표를 하게 된다. 협상 에이전트가 그룹 구성원의 투표 결과를 집계하는데 정해진 정책(예, 회의 소집 조건이 구성원의 과반수 이상)에 맞으면 회의 소집이 결정된다. 이렇게 얻어진 결과는 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트를 통해 개인 일정 데이터베이스와 그룹 일정 데이터 베이스에 저장된다. 반대로 정책에 어긋나면, 집계한 결과를 다시 스케줄러에 보내 다른 날짜를 결정하도록 한다. 이 과정을 반복하여 그룹 구성



<그림 7> 회의 소집 협상을 위한 KQML 메시지의 흐름



원들이 원하는 날짜를 결정한다.

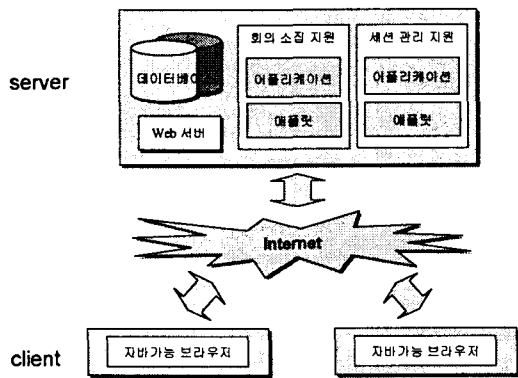
- ① 서버 측에 존재하는 스케줄 에이전트, 협상 에이전트, 그룹 정보 관리 에이전트 그리고 개인 정보 관리 에이전트는 조정 에이전트에게 다음과 같은 사실을 요청하거나 알린다.
  - 스케줄 에이전트 : <그룹 협상 정보>와 <회의 참석 가능 시간>이 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 요청한다.
  - 협상 에이전트 : <그룹 협상 정보>가 갱신될 때마다 알려줄 수 있다고 알린다, <개인 협상 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 요청한다.
  - 그룹 일정 관리 에이전트 : <그룹 협상 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, 스케줄 에이전트에게 필요한 정보를 가지고 있는 에이전트에 대한 정보를 넘겨준다.
- ③ 그룹 정보 관리 에이전트와 개인 정보 관리 에이전트는 <그룹 협상 정보>가 갱신될 때마다 알려달라고 요청한다. 스케줄 에이전트는 <그룹 협상 정보>와 <회의 참석 가능 시간>이 갱신될 때마다 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 알려달라고 인터페이스 에이전트에게 요청한다.
- ⑦ 인터페이스 에이전트는 <개인 협상 정보>가 갱신될 때마다 협상 에이전트에게 알려준다.
- ⑧ 협상 에이전트는 회의 소집 협상이 타결되면 그룹 정보 관리 에이전트와 개인 정보 관리 에이전트에게 <그룹 협상 정보>를 알려준다. 회의 소집 협상이 결렬되면 스케줄 에이전트에게 각 사용자들의 <회의 참석 가능 시간>을 알려주어 스케줄 에이전트에게 다른 시간을 추론하도록 한다.

## 6. 구현 및 활용 예

본 연구에서 제안하는 시스템은 웹 상에서 동작하는 시스템이므로 초기 실행 단계에서 HTTP 프로토콜을 사용하여 인터페이스 에이전트 애플릿을 Web 서버로부터 클라이언트의 자바가능 브라우저로 다운로드 시킨다. 그 이후의 에이전트들 간의 통신은 TCP/IP 프로토콜을 사용한다. 회의를 할 경우 클라이언트 측 협동 에이전트 애플릿과 미디어 에이전트 애플릿을 웹 서버를 통해 클라이언트의 자바가

능 브라우저에 다운로드 시킨 후에 TCP/IP 프로토콜을 사용하여 미디어 정보를 송수신한다. 또한, 복제 구조(replicated architecture)를 가지도록 하여 서버 쪽의 통신 부하를 줄이도록 구현되었다.



<그림 8> 시스템 구현 환경

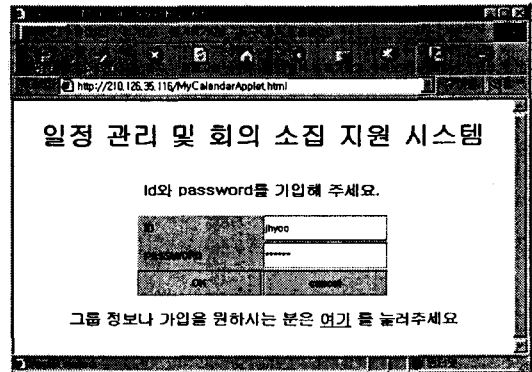
구현 환경은 <그림 8>에서 보는 바와 같이 모든 응용은 서버에 위치하고 필요한 애플릿을 클라이언트의 자바가능 브라우저에 다운로드 시켜 서버와 통신하게 하였다. 서버용 운영체제로 Solaris 2.6.1을 사용하고 클라이언트용 운영체제로는 Windows 98을 사용하였다. 데이터 베이스는 객체지향 데이터베이스인 O2를 사용하였다. 개발 도구로는 JDK 1.2, JATLite를 사용하였다.

JATLite는 Stanford 대학에서 Java 언어를 통해 개발한 템플릿(template)이며, 자바 언어를 사용하여 KQML 메시지를 교환하는 에이전트 프레임워크 개발을 쉽게 해준다. JATLite는 또한 AMR(Agent Message Router)를 제공

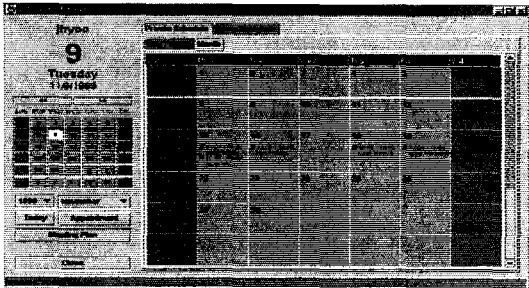
한다. 이것은 에이전트 시스템에서 조정 에이전트의 역할을 수행해 준다.

JATLite는 추상 계층(Abstract Layer), 기본 계층(Base Layer), KQML 계층(KQML Layer), 라우터 계층(Router Layer), 프로토콜 계층(Protocol Layer)의 다섯 가지 계층으로 구성된다. 가장 하위의 추상 계층은 JATLite의 구현을 위해 필요한 추상 클래스의 집합을 제공한다. 기본 계층은 TCP/IP에 기반한 기본 통신을 제공한다. KQML 계층은 KQML 메시지를 파싱하고 저장하는 기능을 제공한다. 라우터 계층은 에이전트들의 이름을 등록하거나, 메시지의 경로를 배정하거나, 큐잉(queueing)을 하는 기능을 제공한다. 프로토콜 계층은 SMTP, FTP, POP3, HTTP와 같은 표준 인터넷 서비스를 제공한다.

<그림 9>는 일정 관리 및 회의 소집 지원 시스템을 사용하기 위한 로그인 화면이다. 시스템을 사용하기 위해 사용자는 ID와 password를 기입하여 인증을 받아야 한다. 개인 일정 관리를 위해 부가적으로 사용자 등록을 해야 하며, 그룹 일정 관리를 위해서는 그룹에 가입



<그림 9> 로그인 화면

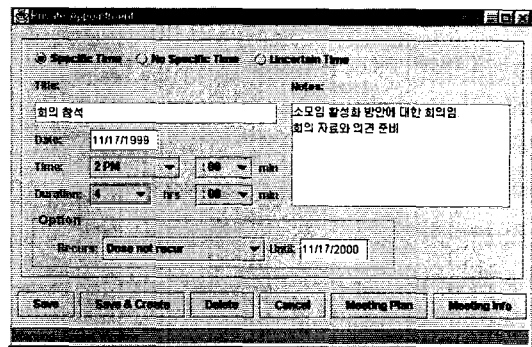


<그림 10> 일정관리 인터페이스

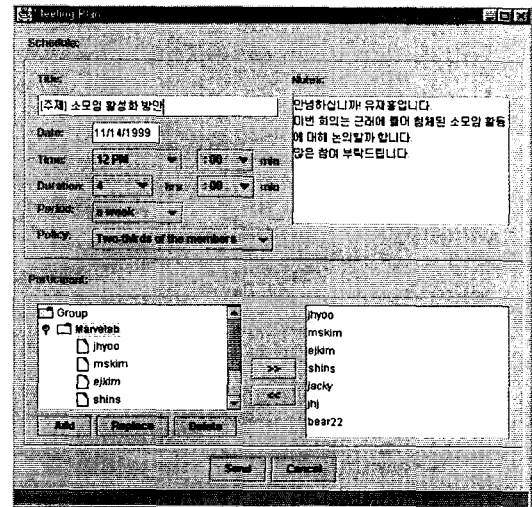
해야 한다. 또한, 매번 서버에 접속할 때마다 사용자 인증을 거쳐야 한다. 현재 시스템에 등록되어 있는 그룹 정보를 조회하거나 사용자 등록을 하기 위해서는 <그림 9>의 아래 부분에 있는 하이퍼링크로 연결된 '여기' 라는 문자를 선택하도록 하였다.

<그림 10>은 본 시스템의 회의 관리 지원을 위한 메인 인터페이스이다. 이 인터페이스는 사용자의 개인 일정과 그룹 일정 열람, 사용자의 개인 일정 등록, 그리고 회의 소집 요청 및 회의 소집 요청에 대한 응답을 처리 할 수 있도록 구성되어 있다. 개인 일정과 그룹 일정 열람 부분에서는 일별, 주별, 월별 일정 열람을 할 수 있도록 구성되어 있다. <그림 10>은 개인 일정 관리 중 월별 일정 관리를 위한 인터페이스를 보여준다.

<그림 11>은 개인 일정 갱신을 위한 인터페이스이다. 개인의 일정은 "Specific Time"과 "Non Specific Time" 그리고 "Uncertain Time"의 세 가지 타입 중에서 하나를 선택하여 등록할 수 있다. 이 인터페이스는 추가적으로 회의 소집 요청을 위해 "Meeting Plan" 버튼과 회의 소집 요청에 대한 응답을 위해 "Meeting Info" 버튼을 제공한다.

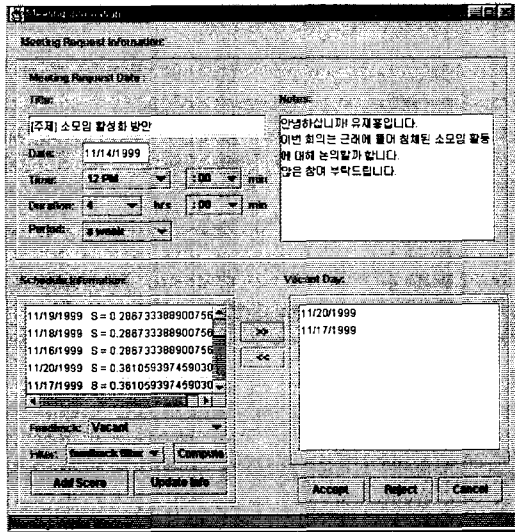


<그림 11> 일정 갱신을 위한 인터페이스



<그림 12> 회의 소집을 위한 인터페이스

<그림 12>는 회의 소집을 위한 인터페이스이다. 회의 주제, 시간대, 회의 정책 등의 회의 정보를 기입하는 부분과 회의에 참여할 사용자를 선택하는 부분으로 구성되어 있다. 이 인터페이스의 회의 정보를 기입하는 부분에 정의되어 있는 정책은 협상 에이전트가 협상하는 정책으로 사용된다.



<그림 13> 회의 소집 요청에 대한 결정을 처리하는 인터페이스

<그림 13>은 회의 소집 요청에 대한 결정을 처리하는 인터페이스이다. 이 인터페이스는 회의 소집 정보를 표시하는 부분, 에이전트를 학습시키는 부분, 그리고 회의 소집 정보에 포함된 주어진 기간 안에서 회의 참가 가능 날짜를 선택하는 세 부분으로 나뉜다. 에이전트를 학습시키는 부분은 주어진 기간 내의 날짜별 상황을 표시해 주는 리스트 박스를 제공한다. 또한, 날짜별 상황을 판단하는 기준이 되는 필터의 종류를 선택하는 기능과 필터를 사용해 나온 결과를 사용자가 판단할 수 있도록 해주는 인터페이스들을 제공한다.

회의 소집을 위한 시나리오를 살펴보면 다음과 같다.

- 그룹의 장이 그룹의 회의에 대한 계획을 세운다. <그림 10>은 일정 계획을 작성한 예이다.
- 그룹의 장이 그룹의 회의 소집을 요청한다.

<그림 11>은 지정된 날짜로부터 일주일 내의 날짜 중에서 회의 소집 날짜가 선택되어야 하며, 회의 소집 요청을 받은 사람의 2/3 이상이 회의에 참여 가능해야 회의가 성사된다는 것을 의미한다.

- 메일을 받은 구성원은 회의 소집 메일을 확인한 후, 회의 소집 요청에 대한 결정을 처리한다. <그림 13>은 인터페이스 에이전트가 두 가지 날짜를 추천했음을 보여준다. 사용자가 이 결과에 만족하면 그대로 회의 소집 요청을 승낙하고, 그렇지 않은 경우에는 결과를 수정하고 인터페이스 에이전트를 학습시킨다. 그런 다음 회의 소집 요청의 승낙/거절 여부를 결정한다.

## 7. 결론

이 연구는 웹 기반 협동작업 시스템에 에이전트 개념(agency)을 결합시킨 시스템에 대한 연구이다. 이 논문에서는 웹 기반 협동작업 시스템을 더욱 유용하게 만들어 주는 에이전트들의 주요 상호 작용 메커니즘을 제안하고 그 메커니즘의 구현에 대하여 자세하게 소개하였다. 또한, 이 논문에서는 회의에 대한 사용자의 관심도와 중요도를 반영하여 최적일시를 추론하는 구체적인 방법을 제시하였다. 제안된 시스템은 효율적인 회의 관리를 위해 회의 소집 지원과 회의 관리 지원을 위한 에이전트들로 구성되어 있다. 회의 소집 지원을 위한 에이전트들은 회의를 소집해줄 뿐만 아니라, 그룹과 개인의 일정을 관리하여 회의를 원하는 그룹

구성원이 다른 구성원들을 회의에 초청할 수 있도록 지원해 준다. 회의 관리 지원을 위한 에이전트들은 회의 소집이 결정된 이후 회의 시작 전후에 활동하며 참여자들에게 보다 편리하게 회의를 할 수 있도록 지원해준다.

본 시스템의 장점들은 아래와 같다.

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

향후 연구로는 이 시스템이 다중 에이전트 환경임을 고려하여 에이전트간의 동기적 협력을 통해 복잡한 문제를 해결할 수 있는 효율적인 기법에 대한 연구가 필요하다.

## 참고문헌

- Stephen Jabele, Steven Rohall, Ralph L. Vinciguerra (1994o). High Performance Infrastructure for visually-Intensive CSCW Applications, Proceedings on CSCW '94, ACM Press, pp. 395-403.
- R. Steinmets and N. Nahrstedt (1994o). Multimedia: Computing, Communications & Applications, Prentice Hall, pp. 854.
- Eric Garland and Dave Rowell (1994n). Face-to-Face Collaboration, Byte, Vol.19, No.11, pp.233-242.
- Walter Reinhard, Jean Schweitzer, Gerd Volksen (1994m). CSCW Tools: Concepts and Architectures, IEEE Computer, Vol. 27, No. 5, pp. 28-36.
- Saul Greenberg and David Marwood (1994o). Read Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface, Proceedings on CSCW '94, ACM Press, pp. 165-173.
- ITU-T Draft Recommendation T.120 (1996a). Data Protocols for Multimedia Conferencing
- Kincaid, C., P. Dupont, and A. Kaye (1985). Electronic calendars in the office: An assessment of user needs and current technology, ACM Transactions on Office Information Systems, 3(1):89-102.
- Sen Sandip and Edmund H. Durfee (1998). A Formal Study of Distributed Meeting Scheduling, Group Decision and Negotiation, pp. 365-389.
- Munindar P. Singh (1998d). "Agent Communication Languages: Rethinking the Principles", IEEE Computer, Vol.31, No12., pp. 40-47.
- Tom Mitchell, Rich Caruana., (1994n). Experience with a Learning Personal Assistant. Communication of the ACM, Volume 37.
- John H. Connolly and Ernest A. Edmonds (Eds.) (1994). CSCW and Artificial Intelligence. Springer-Verlag.
- Joseph P. Bigus and Jennifer Bigus (1998). Construction Intelligent Agents with Java, Wiley, pp. 73-180.
- B. Perlmutter (1995). Gradient calculations for dynamic recurrent neural networks: A survey. IEEE Transactions on Neural Networks, 6(5): 1212-1228.

Abstract

## Implementation of A Multiple-agent System for Conference Calling

Jae Hong Yoo\* · Seung Jin Rho\* · Mee Young Sung\*

Our study is focused on a multiple-agent system to provide efficient collaborative work by automating the conference calling process with the help of intelligent agents. Automating the meeting scheduling requires a careful consideration of the individual official schedule as well as the privacy and personal preferences. Therefore, the automation of conference calling needs the distributed processing task where a separate calendar management process is associated for increasing the reliability and inherent parallelism. This paper describes in detail the design and implementation issues of a multiple-agent system for conference calling that allows the convener and participants to minimize their efforts in creating a meeting. Our system is based on the client-server model. In the server side, a scheduling agent, a negotiating agent, a personal information managing agent, a group information managing agent, a session managing agent, and a coordinating agent are operating. In the client side, an interface agent, a media agent, and a collaborating agent are operating. Agents use a standardized knowledge manipulation language to communicate amongst themselves. Communicating through a standardized knowledge manipulation language allows the system to overcome heterogeneity which is one of the most important problems in communication among agents for distributed collaborative computing. The agents of our system propose the dates on which as many participants as possible are available to attend the conference using the forward chaining algorithm and the back propagation network algorithm.

---

\* Computer Science and Engineering University of Incheon