

웹 화상회의 시스템을 위한 KQML 기반의 멀티 에이전트 구조

성 미 영[†] · 유 재 홍^{††}

요 약

이 연구는 웹 기반 화상회의 시스템에서 지능적인 에이전트들을 통해 회의 소집에 들이는 노력을 최소화함으로써 여러 참여자들에게 동기적 협력을 보다 편리한 형태로 지원해 주는 시스템에 대해 연구하는 것이다. 이 시스템은 멀티 에이전트 구조를 가지며, 에이전트들 사이의 통신은 KQML을 이용한다. 이것은 시스템에 새로운 에이전트를 추가할 때에 에이전트들 사이의 이형질성을 극복할 수 있도록 해준다. 이 논문에서는 웹 화상회의 시스템을 위한 멀티 에이전트들의 전체적인 구조를 제안하고, 이들 에이전트들 사이의 KQML 메시지들의 흐름에 대하여 자세하게 소개한다.

KQML-based Multi-agent Structure for the Teleconferencing System on the Web

Mee-Young Sung[†] · Jae-Hong Yoo^{††}

ABSTRACT

Our study is focused on a multi-agent system which provides the efficient teleconferencing facility with the help of the intelligent agents who allow the participants to minimize the effort to call meeting. Our system has multi-agent architecture and use the KQML to communicate among agents. Communicating through the KQML allows us to overcome the heterogeneity when we add new agents into the system. In this paper, we propose a multi-agent structure for the teleconferencing system based on the web and describe in detail the flow of the KQML messages among agents.

1. 서 론

지난 10여 년 동안 멀티미디어 기술과 컴퓨터 네트워크 기술이 급속하게 발전하였고 이 기술들의 결합으로 최근에 있던 CSCW(Computer Supported Cooperative Work)[1, 2, 3] 분야의 급속한 발전과 더불어 원격 회의, 원격교육, 원격자문, 공동저작 등에 대한 요구가 날로 커지고 있다. 또한, 국내외의 많은 연구자들이 에

이전트에 대한 연구에 관심을 쏟고 있다. 이는 그들의 시스템을 유연성과 적응성을 갖춘 지능형으로 발전시키는데 에이전트가 중요한 역할을 할 것이라 기대하기 때문이다. 에이전트는 자율성(autonomy), 사회성(social ability), 이동성(mobility), 지능(intelligence) 등의 특성을 가지며, 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어이다. 이러한 에이전트가 실세계 시스템인 협동작업 시스템에서 활약한다면 기존의 여러 소프트웨어에 존재하는 기능성을 인간에게 보다 편리한 형태의 서비스로 제공할 수 있을 것이다.

현재, 많은 사용자들이 웹 브라우저만을 가지고 실

※ 본 연구는 한국과학재단 지정 인천대학교 멀티미디어연구센터의 지원에 의한 것임.

† 종신회원 : 인천대학교 전자계산학과 교수

†† 준 회원 : 인천대학교 대학원 전자계산학과

논문접수 : 1999년 9월 2일, 심사완료 : 1999년 10월 20일

세계에서와 같이 동기적이고 협동적인 작업을 지원하는 좋은 도구를 필요로 하고 있다. 웹 서비스는 웹 서버가 제공할 서비스에 대한 정보를 저장하고 있고 사용자는 원하는 경우에 단순히 정보를 받아 보는 단방향 서비스인 반면 화상회의[7-11] 등의 협동작업 시스템은 여러 사용자가 시스템과 상호 작용 할 수 있어야 협동이 가능하다. 이러한 필요를 충족시켜주는 좋은 협동작업 도구가 없는 지금 웹(World Wide Web)상에서 동작하며 지능을 가지는 에이전트의 도움을 받아 효율적으로 협동작업을 하는 시스템이 절대적으로 필요하다.

현재 웹 기반 화상회의 시스템은 많이 존재하지만, 이러한 시스템들은 전문가들의 공동작업이나 뚜렷한 목적을 가지는 회의 등의 긴밀한 협동을 필요로 하는 회의 시에 참여자들을 참여시키는데 많은 노력을 필요로 한다. 예를 들면, 회의 소집자가 다른 참여자들에게 일일이 연락을 하여 각 참여자의 회의 가능한 시간을 조사하고 이중 가장 적절한 시간을 회의의 시간으로 결정하는데 많은 시간과 노력을 들여야 한다.

이 논문에서는 웹 기반 협동작업 시스템에서 지능적인 에이전트들을 통해 회의 소집에 들이는 노력을 최소화함으로써 여러 전문가들에게 동기적 협력을 보다 편리한 형태로 지원해 주는 멀티에이전트 구조를 제안한다. 본 논문에서 제안하는 멀티에이전트 구조에서는 분산 협동 처리를 위한 에이전트들 간의 통신에 있어 가장 큰 문제점인 이형질성(heterogeneity)을 극복하기 위해 KQML을 이용하여 지식을 교환하는 기법을 사용하였다. 우선 웹 기반 화상회의 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 끝으로 결론을 맺겠다.

2. 관련 연구

2.1 에이전트

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 살아가

는 "soft robot"이라고 할 수 있다. 일반적 에이전트는 자율성, 사회성, 이동성, 지능 등의 특성을 가진다[5]. 에이전트의 특성에 대해 간단히 살펴보면 다음과 같다.

- 자율성 - 주위의 환경을 인식하여 어떤 행동을 취하기 위해 스스로 적절한 판단을 하는 능동성을 갖는 것을 말한다.
- 지능 - 지식 베이스, 추론 능력, 학습 능력을 가지고 사용자의 의도를 파악하여 계획을 세우고 새로운 지식을 스스로 터득하는 성질로서 자율성과 밀접한 관계를 가진다.
- 이동성 - 기존의 클라이언트/서버 개념과는 다르게 사용자가 요구한 작업을 실제 작업을 처리하는 호스트로 이동시켜 수행함으로써 수행 효율을 높이고 네트워크 부하를 줄이는 효과를 얻을 수 있다.
- 사교성 - 하나의 에이전트로는 처리하지 못하는 작업의 수행을 위해 다른 에이전트와 통신을 통해 정보를 얻는다.

위의 특징을 모두 갖추어야 에이전트는 아니며, 이러한 특성을 얼마나 많이 가지냐에 따라서 얼마나 능동적인가를 판단하는 척도가 된다.

2.1.1 멀티 에이전트 시스템(Multi-agent systems)

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. 과거 1980년대 말까지도 에이전트-기반 시스템이 하나의 에이전트로 구성된 것이 주류였다. 그러나 사용자의 요구사항이 다양해짐에 따라 단독의 에이전트가 인지적인 측면(cognitive aspect)에서 이를 해결하기 위한 지능을 가지는 어렵기 때문에 분산 인공지능(distributed artificial intelligence)을 이용하여 지역적으로 떨어진 다른 에이전트의 도움을 받아 처리하는 분산 협동 처리 개념을 접목시켜야 할 필요가 생겨났다. 이를 멀티 에이전트라 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 처리능력 등의 정보를 가지고 각

각의 응용 에이전트를 연결해 주는 네임 서버 역할을 한다. 이러한 멀티에이전트 시스템에서 에이전트들 간의 통신을 위해 다음 사항들을 고려해야 한다.

- 에이전트들은 같은 토큰(token)들의 집합 또는 같은 심벌(symbol)들을 사용하는 같은 언어로 통신해야 한다.
- 에이전트들 간에 사용되는 심벌의 의미는 동일해야 한다.
- 또한 정의된 언어에서 에이전트들 간에 통신을 교환하기 위한 방법을 개발해야만 한다.
- 시스템 내의 자원에 대한 동기화를 고려해야 한다.

위의 문제를 해결하기 위한 한가지 방법은 블랙보드 구조(blackboard architecture)를 사용하는 것이다. 블랙 보드는 다중 지식 소스(multiple knowledge sources)들을 위한 일반적인 통신 기법으로서 제어기(controller)에 의해 관리되고 조정되는 일종의 데이터 구조(data structure)이다. 이 구조를 사용하는 블랙보드 시스템은 지식 소스(knowledge source)나 에이전트들이 블랙보드 구조와 인터페이스에 매우 밀접하게 묶여 있다. 만일 단일하고 커다란 응용을 만들기를 원하거나 지식 베이스를 모듈 방식으로 만들기를 원한다면, 블랙보드 시스템이 매우 좋다. 그러나 만일 함께 작업할 블랙보드에 대하여 알기를 원하지 않거나 에이전트들이 매우 다른 구조를 가지는 환경을 원한다면, 보다 형식이 갖추어진 인터페이스인 표준화된 메시지 형태와 프로토콜이 필요하다. 이에 대하여 많은 연구가 진행되고 있으며, 그 대표적인 예가 KQML(knowledge Query and Manipulation Language)[4]이다.

2.1.2 KQML

KQML은 정보와 지식을 교환하기 위한 프로그램들과 에이전트들을 위한 프레임워크를 제공한다. KIF(Knowledge Interchange Format)와 함께, KQML은 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 맞추고 있다. KQML은 각각 다른 지식 베이스에서 수행될 연산들(operations)을 정의하는 방법을 제공하고, 조정에이전트(coordinating agent, facilitator)를 통해 지식과 정보

를 공유하려 하는 에이전트들을 위한 기초적인 구조를 제공한다.

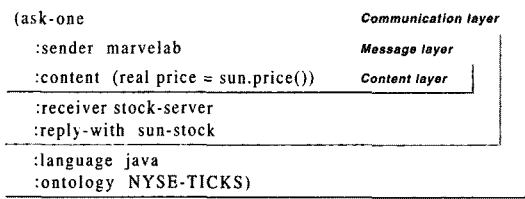
KQML 메시지는 수행문(performative)이라고도 불린다. 각 메시지는 암시적으로 몇 가지 명세된 행동을 수행하도록 예약되어 있다. KQML에는 많은 수의 수행문들이 정의되어있다. 대부분의 에이전트 기반의 시스템들은 전체 수행문의 작은 부분집합만을 지원하고 있다. 수행문을 사용하는 에이전트는 다른 에이전트에 정보를 요구할 수 있고, 다른 에이전트에게 자신의 지식베이스에 존재하는 지식들을 가르쳐줄 수 있고, 에이전트들의 서비스를 예약할 수 있다. 그리고 그들 자신의 고유한 서비스들을 제공할 수도 있다.

KQML에 대한 문서로서는 현재 1993년에 발표된 KQML 명세서[15]와 1997년에 제안된 새로운 KQML 명세를 위한 제안서[16]가 있다. 다음은 1997년에 제안된 KQML 명세를 위한 제안서에서 나타난 예약된 수행문에 대한 내용이다. 각각의 의미에 대해서는 제안서를 참고하기 바란다.

- Discourse : ask-if, ask-all, ask-one, stream-all, eos, tell, untell, deny, insert, uninsert, delete-one, delete-all, undelete, achieve, unachieve, advertise, unadvertise, subscribe
- Intervention and mechanics : error, sorry, standby, ready, next, rest, discard
- Networking and Facilitation : register, unregister, forward, broadcast, transport-address, broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

KQML은 몇몇 특별한 영역에 응용할 수 있는 의미, 개념, 그리고 관계의 명백한 명세인 온톨로지(ontology)를 사용한다. 이 온톨로지는 같은 언어로 통신하는 두 에이전트가 그 언어로 된 문장을 정확히 통역할 수 있도록 보장해 준다. 이것은 2.1.1절에서 언급한 멀티에이전트 시스템에서 에이전트들 간의 통신을 위해 고려해야 할 사항 중에 첫 번째와 두 번째의 경우이다.

KQML 메시지들은 세 개의 다른 계층에서 정보를 기호화한다. 세 개의 계층은 내용 계층(content layer)과 메시지 계층(message layer) 그리고 통신 계층(communication layer)이다. (그림 1)은 이름이 marvelab인 에이전트가 SUN 주식의 가격에 관한 요청을 하는 KQML 메시지의 예이다.

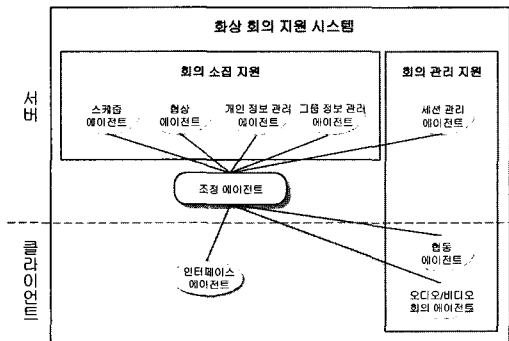


(그림 1) KQML 메시지의 예

(그림 1)의 경우에서 통신 계층은 수신자에게 정보를 요구하는 KQML 수행문 ask-one과 어떤 언어로 정의되어 있는지의 :language 그리고 온톨로지의 이름을 나타내는 :ontology로 구성된다. 메시지 계층은 송신자를 나타내는 :sender와 수신자를 나타내는 :receiver 그리고 수신자가 응답을 할 때 레이블로 사용하는 :reply-with로 구성된다. 내용 계층은 수행문이 나타내려는 의도에 관한 정보를 포함하는 :content로 구성된다.

3. 전체 시스템 구조

이 시스템은 개인과 개인, 개인과 그룹, 그룹과 그룹 간에 효율적인 화상회의를 위해 회의 소집부터 화상회의를 위한 회의 관리까지 수행해준다. 전체 시스템의 구조는 (그림 2)와 같으며, 여러 에이전트가 하나의 시스템을 구성하는 멀티 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. 여기서 개인 일정 관리를 위하여 부가적으로 사용자 등록과 인증이 따른다.



(그림 2) 전체 시스템 구성도

이 에이전트들은 클라이언트 측과 서버 측에 분산되어 있으며, 서버 측에서는 그룹 정보 관리 에이전트, 스케줄 에이전트, 협상 에이전트, 개인 정보 관리 에이전트, 세션관리 에이전트, 그리고 조정 에이전트가 활동한다. 클라이언트 측에서는 인터페이스 에이전트, 오디오/비디오 관리 에이전트 그리고 협동 에이전트가 활동한다. 이 장에서는 시스템에서 사용되는 에이전트들과 몇몇 에이전트들에 의해 지원되는 작업에 대해 살펴보고자 하겠다.

3.1 조정 에이전트

조정 에이전트는 서버 측에서 활동하며, 시스템 내의 모든 에이전트의 위치 정보, 각 에이전트의 처리능력 등의 정보를 가지고 각각의 응용 에이전트를 연결해 주는 네임 서버 역할을 한다. 조정 에이전트를 통해 구성되는 에이전트들 간의 통신 구조에 대해서는 4장에서 설명한다.

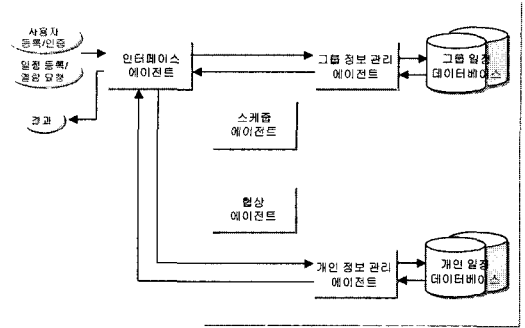
3.2 회의 소집 지원 에이전트

회의 소집 지원을 위한 에이전트들은 그룹 정보 관리 에이전트, 스케줄 에이전트, 협상 에이전트, 개인 정보 관리 에이전트들이다. 개인의 일정은 물론 그룹의 일정을 관리함으로써 화상 회의의 위한 준비 작업(참여자 초청, 참여자의 참가 가능 여부, 회의의 성사 여부 등)을 쉽게 해준다. 이를 위해 부가적으로 개인 일정을 위해 사용자 등록을 해야하며, 그룹 일정 관리를 위해서는 그룹에 가입해야 한다. 또한 매번 서버에 접속 시 사용자 인증을 거쳐야 한다. 사용자 등록, 사용자 인증, 개인 일정 관리 그리고 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 추천하는데 유용한 정보로 사용된다.

(그림 3)은 회의 소집 제안 및 협상 과정을 나타낸다. 회의 소집 제안은 다른 사용자들과 회의를 원할 때 이루어지며, 회의 소집 정보(희망 회의 소집 날짜 리스트, 참여자 리스트, 회의 내용 등)와 함께 인터페이스 에이전트를 거쳐 스케줄 에이전트에 보내진다. 스케줄 에이전트는 회의 소집 정보를 바탕으로 참여자들의 개인 일정 정보와 그룹 일정 정보를 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트에게서 얻어

사용자의 희망 회의 소집 날짜 리스트에 맞는 날짜를 추천한다. 이 정보는 회의 소집 정보와 함께 협상 에이전트에 보내지며, 협상 에이전트는 스케줄 에이전트가 추천한 날짜와 회의 소집 정보를 개인 정보 관리 에이전트와 사용자에게 보낸다. 다른 사용자들은 메일을 확인하거나 일정관리 응용 프로그램을 사용하여 서버에 접속하여 회의 소집 요청을 확인한 후, 회의 참가 여부를 결정한다. 이 정보는 협상 에이전트에 보내지고, 협상 에이전트는 사용자의 보내온 정보를 통해 회의 소집 가능 여부를 결정한다. 회의 소집이 가능하면 개인 정보 에이전트와 그룹 정보 관리 에이전트에 회의 소집 정보를 보내고, 그렇지 않으면 다시 스케줄 에이전트가 다른 날짜를 추천한다. 이 과정은 회의 소집 날짜가 정해지거나 회의 소집을 요청한 사용자가 회의 소집을 취소할 때까지 반복된다.

위에서 설명한 회의 소집 과정에서 스케줄 에이전트가 참여자들의 일정 중에 빈 일정 날짜를 찾아내고, 참여자들의 회의 참석 여부의 응답을 받아 회의 소집 성사 여부를 판단하는 과정에서 추천 알고리즘이 사용된다. 추천 시에 발생하는 충돌 집합으로부터 단일 규칙을 선택하기 위해서 사용되는 충돌 비교 함수(conflict resolution) 과정에서는 전건(antecedent)절의 논리곱의 수가 가장 많은 규칙을 선택하는 방법을 사용한다. 그러나 만일 규칙이 이전 사이클에서 점화되었다면, 그 규칙은 충돌 집합에 더하지 않는다.

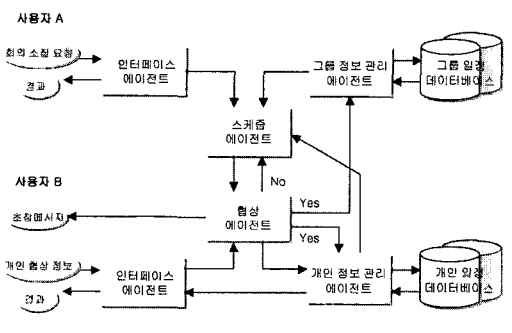


(그림 4) 사용자 등록/인증과 개인/그룹 일정 열람

같은 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 일정 정보를 관리하고 이를 필요로 하는 다른 에이전트에게 제공해준다. 예를 들어, 사용자의 개인 일정 열람 요구를 받은 인터페이스 에이전트가 개인 정보 관리 에이전트에 개인 일정 열람을 요구할 경우, 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 필요한 개인 정보 요구의 경우 등에 응답한다. 또한, 회의 소집에 대한 정보가 생성되거나 갱신되었을 경우, 이 정보를 인터페이스 에이전트에게 전달한다.
- 그룹 정보 관리 에이전트**
 그룹의 일정 정보와 사용자의 인증 정보와 사용자가 속해 있는 그룹 정보를 관리하며, 이 정보를 필요로 하는 다른 에이전트에게 제공해준다. 예를 들어, 사용자의 그룹 일정 열람이 인터페이스 에이전트를 통해 그룹 정보 관리 에이전트에 요구되었을 경우, 해당 그룹의 정보를 보려하는 사람이 그 그룹에 소속된 사용자인지를 결정하는 사용자 인증의 경우, 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 필요한 그룹 정보 요구의 경우 등에 응답한다. 또한 협상 에이전트에서 성사된 회의 정보를 받아 데이터 베이스에 저장한 후, 세션 관리 에이전트(3.3절)에게 전송한다.
- 스케줄 에이전트**
 그룹의 일정 정보와 사용자의 인증 정보를 관리하



(그림 3) 회의 소집 제안 및 협상

(그림 4)는 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 열

며, 이 정보를 필요로 하는 다른 에이전트에게 제공해준다. 예를 들어, 사용자의 그룹 일정 열람 요구를 받은 인터페이스 에이전트가 그룹 정보 관리 에이전트에 그룹 일정 열람을 요구할 경우, 해당 그룹의 정보를 보려하는 사람이 그 그룹에 소속된 사용자인지를 결정하는 사용자 인증의 경우, 스케줄 에이전트가 적절한 회의 소집 날짜를 추천하기 위해 필요한 그룹 정보 요구의 경우 등에 응답한다.

● 협상 에이전트

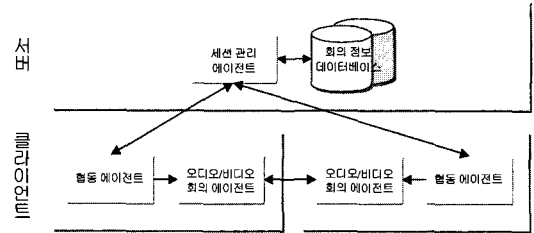
협상 에이전트는 스케줄 에이전트가 알려준 회의 소집 날짜와 회의 참가자 명단을 받아서 해당 참가자에게 소집 날짜와 회의 내용을 전달한다. 이 정보는 일정에 임시적으로 추가되며, 다른 회의 소집 요청 등에 있어서 일정으로 간주된다. 그후, 참가자들의 회의 참석 여부에 대한 응답을 받아 회의 소집의 성사 여부를 결정한다. 여기서 참가자들의 응답으로는 회의 참석 가능과 회의 참석 불가능이다. 회의 참석 불가능일 경우, 회의 참가 불가능 사유와 회의 가능 날짜를 같이 보낸다. 여기서 참여자가 회의 중간에 참여하는 부분 참여는 고려하지 않는다.

정해진 정책(예, 회의 소집 조건이 구성원의 과반수 이상)에 어긋나서 모임이 성사되지 못하면 스케줄 에이전트에게 회의 참가 불가능 명단과 그들의 회의 가능 날짜를 전송하여 다른 날짜를 추천하도록 한다, 반면, 회의가 성사되면 각 그룹 구성원에게 메일을 전송하고 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트를 통해 개인 일정 데이터베이스와 그룹 일정 데이터 베이스를 갱신한다.

3.3 회의 관리 지원 에이전트

회의 관리를 지원해 주는 에이전트들은 세션 관리 에이전트, 협동에이전트, 오디오/비디오 회의 에이전트로 구성되며, 화상회의를 효율적으로 할 수 있도록 회의 준비와 회의 관리 작업을 수행한다. 회의 준비 회의 소집이 이루어진 후에 화상회의로 넘어가는 시점에서 일어나는 작업이며, 회의 관리는 회의 참가자, 회의 이름, 회의 시작 시간, 회의에 관한 방침 등과 같은 공유되는 상태 정보를 제어하는 작업이다. 따라서 회의 설정, 회의 종료 그리고 참여 및 탈퇴 기능이 수반된다. (그림 5)에서 보는 바와 같이 세션 관리 에이전트는 서버에 위치하며 클라이언트에 위치한 협동 에이전

트와 상호작용 한다. 여기서 세션 관리 에이전트는 전체 회의 정보를 관리하며, 협동 에이전트는 현재 참여 중인 회의 정보를 관리한다. 이것은 화상회의 중에 서버의 부하를 줄이기 위함이다.



(그림 5) 회의 관리

● 세션 관리 에이전트

세션 관리 에이전트는 서버에만 존재하며, 회의 시작전후로 하여 모든 회의의 참가자, 회의 이름, 회의 시작 시간, 회의에 관한 방침 등과 같은 공유되는 상태 정보를 제어하는 일을 한다. 3.2 절에서 살펴본 그룹 정보 관리 에이전트에서 확정된 회의에 대한 정보가 전달되면 해당 회의를 개설해 준다.

세션 관리 에이전트는 모든 협동작업 회의 정보를 유지하며 아래와 같은 작업을 담당한다.

- 회의 소집
 - 회의 소집 협상을 통해서 예약된 회의 소집
 - 클라이언트 측 협동에이전트에 의한 회의 소집 요청에 의한 회의 소집
- 회의 제어 관리
 - 회의 설정이나 변경 및 종료
- 원하는 회의와 연결
- 참가자 인증
- 구성원 지위 제어 관리
- 전체 참여자 회의그룹 정보 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 멀티캐스트 요청
- 개인 참여자 회의그룹 정보 관리
- 응용프로그램으로부터의 회의 정보 참조 등의 요구 처리
- 오디오/비디오 관리 에이전트
 - 오디오/비디오 관리 에이전트는 각 참여자의 실제 음성(오디오)과 모습(비디오)을 모든 다른 참가자들에게 전송하며 아래와 같은 일을 담당한다.
 - 그룹 통신 구조 정보 유지
 - 미디어 제어 관리
 - 다른 오디오/비디오 회의 에이전트사이의 오디오/비디오 데이터 통신
 - 오디오와 비디오간의 동기화

3.4 인터페이스 에이전트

사용자를 지원하기 위한 에이전트는 인터페이스 에이전트이며, 사용자의 작업을 편리하게 도와주는 일을 한다. 인터페이스 에이전트는 응용프로그램이 어떤 기능을 수행하는데 있어서 필요한 정보를 제공해주기 위해 사용자와 협상을 한다. 만약 어떤 작업의 수행에 있어서 사용자가 필요한 정보를 제공하지 않을 경우 해당 작업의 중요도에 따라 다른 작업을 수행한다. 중요한 작업은 사용자에게 정보를 제공하길 요구하고, 그렇지 않으면 인터페이스 에이전트는 미리 준비된 작업을 순차적으로 수행한다. 이런 인터페이스 에이전트의 기능은 사용자의 기호와 수행되어졌던 작업에 대한 지식을 바탕으로 학습을 통해 이루어진다.

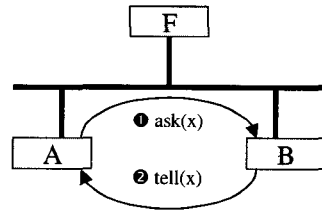
인터페이스 에이전트의 학습을 위해 약속의 종류를 날짜와 시간이 정해진 약속, 날짜는 정해졌으나 시간이 정해지지 않은 약속, 날짜는 정해지지 않았으나 시간은 정해진 약속과 날짜와 시간이 정해지지 않은 약속인 네 가지로 분류한 다음, 이것을 다시 그룹과 개인 일정에 각각 적용시켜서 모두 여덟 가지의 약속으로 분류하였다.

날짜와 시간이 정해진 일정은 개인의 경우 다른 사람과의 약속이나 그룹의 경우 임시 회의나 정기 모임 등이다. 약속 날짜는 정해졌으나 시간이 명확히 정해져있지 않은 약속은 기념일 등으로 하루 중에 어떤 시간대에 약속이 있다는 것을 나타낸다. 날짜는 정해지지 않았으나 시간은 정해진 약속은 여러 날 중에서 꼭 그 시간대가 아니면 안 되는 약속을 나타내며, 회의

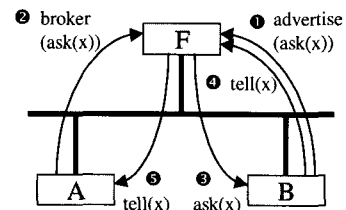
소집 요청에 의해 생성된 임시적인 일정도 여기에 포함된다. 날짜와 시간이 정해지지 않은 약속은 날짜는 정해지지 않았으나 시간은 정해진 약속에 비해 더 막연하고 더 미래에 있을 약속을 나타내는 것이다. 인터페이스 에이전트는 이 여덟 가지의 약속에 대한 사용자의 선호도와 중요도를 학습한다.

4. 에이전트들 간의 KQML 메시지 흐름

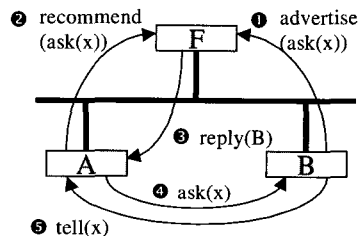
외부와의 정보 교환 시 누구에게 무엇을 요구해야 하는가에 대한 문제가 발생하게 된다. 즉, 어떤 정보를 어떤 에이전트로부터 얻을 수 있는가 하는 문제다. KQML은 (그림 6), (그림 7), (그림 8)의 구조를 지원하며, 그림에서 A와 B는 응용 에이전트를 나타내며, F는 조정 에이전트를 나타낸다.



(그림 6) A가 B를 알고 있어, B에게 직접 질의한다.



(그림 7) A는 F에게 의뢰한다. F는 대신 B에게 질의하여 결과를 A에게 돌려준다.



(그림 8) A는 F에게 물어 보아 B의 위치를 알아낸 후 B에게 직접 질의한다.

(그림 6)은 자신이 필요로 하는 정보를 얻을 수 있는 다른 에이전트의 질의 응답 능력에 대해 이미 알고 있는 경우이다. (그림 7)은 정보를 필요로 하는 에이전트들이 facilitator의 위치만을 알고 있으며, facilitator는 모든 에이전트의 질의 응답 정보를 가지고 있는 경우이다. 이 경우 모든 에이전트의 메시지들이 facilitator를 통해 전달됨으로써, facilitator의 능력에 따라 메시지에 대한 신뢰성, 응답성을 보장받을 수 있다. (그림 8)은 정보를 필요로 하는 에이전트들은 facilitator의 위치를 알고 있으면서 facilitator에게 정보를 제공해 줄 수 있는 에이전트의 위치를 얻는 경우이다. 이 경우 facilitator 혹은 다른 에이전트에게 질의의 해결에 대해 도움을 직접 청하거나, 같이 상호 협조하면서 이를 해결해 나갈 수 있다. 이러한 방법은 에이전트가 필요로 하는 적절한 정보를 수집하는데 크게 효율적일 수 있기 때문에 (그림 7) 보다 개방형 시스템 환경에 더 적합한 방법이다. 본 논문에서는 (그림 8)의 방법을 사용하여 KQML 메시지의 흐름을 나타내었다.

이 장에서는 전체 시스템에서 이루어지는 작업을 두 범주로 나눈 회의 소집 지원을 위한 작업과 회의 관리 지원을 위한 작업에 대하여 KQML 적용한 예를 그림으로 나타내며, 회의 소집 지원을 위한 작업 중에서 사용자 등록 및 인증에 대해서만 KQML 메시지의 흐름을 상세하게 설명한다. 나머지 작업들에 대한 KQML 메시지의 흐름은 지면 관계상 간단하게 설명한다.

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 않는다. 그리고 KQML 메시지의 흐름을 나타내는 이 절의 그림들에서 사용된 기호들에 대해 설명하면 다음과 같다.

- 각 에이전트와 함께 위치하는 설명 상자 안에는 각 에이전트들이 다른 에이전트들에 전달할 수 있는 수행문들을 나타낸다.
- VKB를 “<”와 “>”를 사용하여 나타내었다.
- 수행문 앞에 붙어있는 숫자 기호는 수행문들이 전달되는 순서를 나타낸다.
- 점선은 수행문의 적용을 나타내며, 실선은 수행문의 전달 방향을 나타낸다.

4.1 회의 소집 지원을 위한 KQML 메시지 흐름

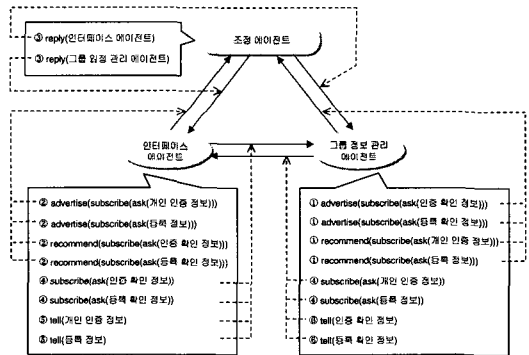
이 절에서는 3.2절에서 살펴본 사용자 등록 및 인증,

그림 일정 열람, 개인 일정 열람, 회의 소집 제안 그리고 회의 소집 협상에 대해서 KQML 메시지를 적용한다.

각 작업에서의 KQML 메시지의 흐름을 살펴보면 다음과 같다.

(1) 사용자 등록 및 인증

사용자 등록은 자신의 인적 사항, ID, 패스워드 등을 기입하여 회원으로 가입하는 작업이며, KQML 메시지의 흐름은 (그림 9)와 같다. 따라서 주어진 회원 가입 절차에 따라 자신의 등록 정보를 기입하고 인터페이스 에이전트를 통해 등록 정보를 그룹 일정 관리 에이전트에 보낸 후 그룹 일정 관리 에이전트로부터 회원 가입에 대한 확인을 받는다. 회원으로 등록되어 있는 경우, 사용자 인증은 자신의 ID와 패스워드를 기입하여 자신이 회원임을 입증 받아야 한다. 회원으로 등록되어 있지 않으면 사용자 등록을 해야한다.



(그림 9) 사용자 등록 및 인증을 위한 KQML 메시지의 흐름

각 에이전트에서 사용되는 수행문들과 그들의 순서를 사용자 인증과 사용자 등록으로 나누어 살펴보면 다음과 같다.

• 사용자 등록

- ① 서버 측에 존재하는 그룹 정보 관리 에이전트는 자신이 <등록 확인 정보>가 갱신될 때마다 알려줄 수 있으며, <등록 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 원한다고 조정 에이전트에게 알린다.
- ② 클라이언트 측에 있는 인터페이스 에이전트는 자신의 <등록 정보>가 갱신될 때마다 알려줄 수 있으며, <등록 확인 정보>가 갱신될 때마다 알

려줄 수 있는 에이전트의 정보를 원한다고 조정 에이전트에게 알린다.

- ③ 조정 에이전트는 각 에이전트에게 필요로 하는 정보를 가지고 있는 에이전트에 대한 정보를 넘겨준다.
- ④ 인터페이스 에이전트는 <등록 확인 정보>가 갱신될 때마다 알려달라고 그룹 정보 관리 에이전트에게 요청한다. 그룹 정보 관리 에이전트는 <등록 정보>가 갱신될 때마다 알려달라고 요청한다.
- ⑤ 인터페이스 에이전트는 <등록 정보>가 갱신될 때마다 그룹 정보 관리 에이전트에게 제공해 준다.
- ⑥ 그룹 정보 관리 에이전트는 인터페이스 에이전트에게서 받은 정보를 가지고 등록 절차를 거쳐서 나온 결과로 <등록 확인 정보>가 갱신될 때마다 인터페이스 에이전트에게 제공하여 준다.

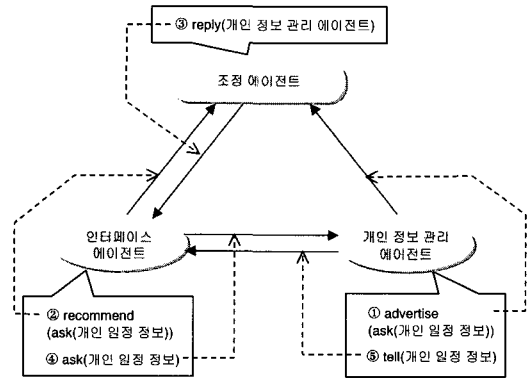
● 사용자 인증

- ① 서버 측에 존재하는 그룹 정보 관리 에이전트는 자신의 <인증 확인 정보>가 갱신될 때마다 알려줄 수 있으며, <개인 인증 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 원한다고 조정 에이전트에게 알린다.
- ② 클라이언트 측에 있는 인터페이스 에이전트가 <개인 인증 정보>가 갱신될 때마다 알려줄 수 있으며, <인증 확인 정보>가 갱신될 때마다 알려줄 수 있는 에이전트에 대한 정보를 원한다고 조정 에이전트에게 알린다.
- ③ 조정 에이전트는 각 에이전트에게 필요로 하는 정보를 가지고 있는 에이전트에 대한 정보를 넘겨준다.
- ④ 인터페이스 에이전트는 <인증 확인 정보>가 갱신될 때마다 알려달라고 그룹 정보 관리 에이전트에게 요청한다. 그룹 정보 관리 에이전트는 <개인 인증 정보>가 갱신될 때마다 알려달라고 요청한다.
- ⑤ 인터페이스 에이전트는 <인증 확인 정보>가 갱신될 때마다 그룹 정보 관리 에이전트에게 알려 준다.
- ⑥ 그룹 정보 관리 에이전트는 인터페이스 에이전트에게서 받은 정보를 가지고 인증 확인 절차를 거쳐서 나온 결과로 <인증 확인 정보>가 갱신될 때마다 인터페이스 에이전트에게 알려 준다.

(2) 개인 일정 열람

개인 일정 열람은 사용자가 자신의 일정을 확인하고자 할 때 행해지는 작업이며, KQML 메시지의 흐름은 (그림 10)과 같다. 인터페이스 에이전트는 개인 정보

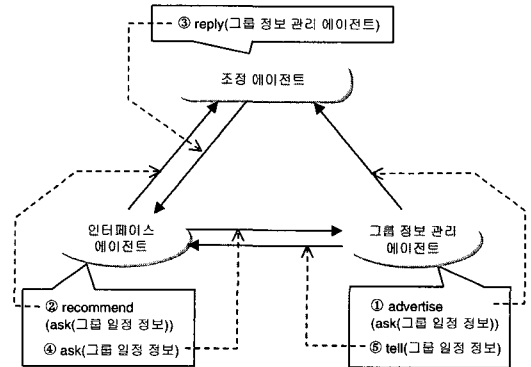
관리 에이전트에 저장되어 있는 그룹 일정을 얻어다가 사용자에게 제시해 준다.



(그림 10) 개인 일정 열람을 위한 KQML 메시지의 흐름

(3) 그룹 일정 열람

그룹 일정 열람은 사용자가 그룹 행사 등의 일정을 확인하고자 할 때 행해지는 작업이며, KQML 메시지의 흐름은 (그림 11)과 같다. 인터페이스 에이전트는 그룹 정보 관리 에이전트에 저장되어 있는 그룹 일정을 얻어다가 사용자에게 제시해 준다.

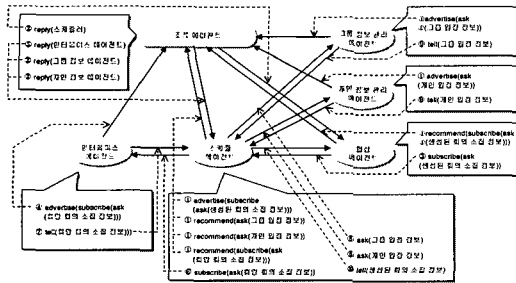


(그림 11) 그룹 일정 열람을 위한 KQML 메시지의 흐름

(4) 회의 소집 제안

회의 소집 제안은 어떤 사용자가 회의를 개최하기를 원할 경우 이루어지는 작업이며, KQML 메시지의 흐름은 (그림 12)와 같다. 우선, 그룹의 우두머리가 원하는 회의 기간을 정한다. 이 정보는 인터페이스 에이전트를 통해 스케줄 에이전트에 전달되고 스케줄

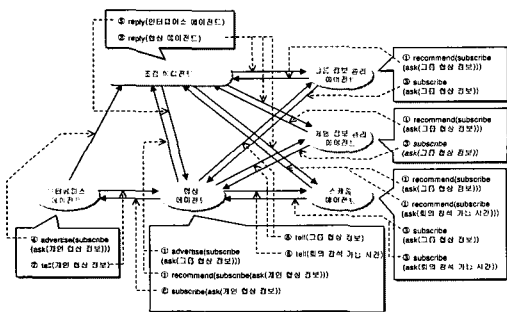
에이전트는 해당 그룹의 일정과 구성원들 개인의 일정을 얻어서 알맞은 날짜를 추측해 낸다. 이 결과는 협상 에이전트에 보내져 모든 사용자들에게 메일로 전송된다.



(그림 12) 회의 소집 제안을 위한 KQML 메시지의 흐름

(5) 회의 소집 협상

회의 소집 협상은 회의 소집 제안 다음에 이루어지는 작업이며, 이 작업을 위한 KQML 메시지의 흐름은 (그림 13)과 같다. 스케줄 에이전트에 의해 추측된 회의 소집 날짜를 그룹 구성원에게 전달한 후, 그룹 구성원들은 인터페이스 에이전트를 통해 투표를 하게 되고, 협상 에이전트는 그룹 구성원의 투표 결과를 집계한다. 정해진 정책(예, 회의 소집 조건이 구성원의 과반수 이상)에 맞으면 회의 소집이 결정되고, 이렇게 얻어진 결과는 개인 정보 관리 에이전트와 그룹 정보 관리 에이전트를 통해 개인 일정 데이터베이스와 그룹 일정 데이터 베이스에 저장된다. 반대로 정책에 어긋나면 결과를 다시 스케줄 에이전트에 보내 다른 날짜를 결정하도록 한다. 이 과정을 반복하여 그룹 구성원들이 원하는 날짜를 결정한다.



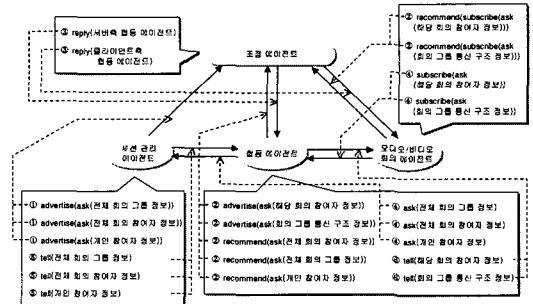
(그림 13) 회의 소집 협상을 위한 KQML 메시지의 흐름

4.2 회의 관리 지원을 위한 KQML 메시지 흐름

이 절에서는 3.3절에서 살펴본 회의 관리와 회의 소집에 대해서 KQML 메시지를 적용한다.

(1) 회의 관리

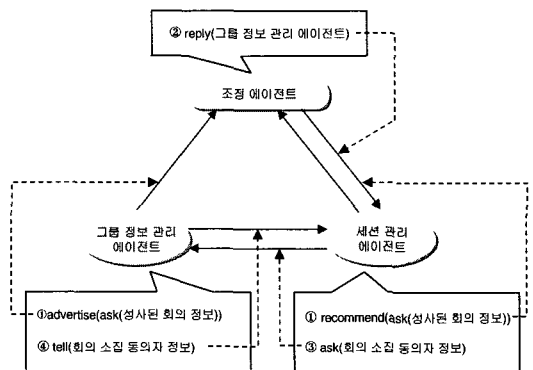
회의 관리의 회의 시작전후로 하여 회의 참가자, 회의 이름, 회의 시작 시간, 회의에 관한 방침 등과 같은 공유되는 상태 정보를 제어하는 일이다. 이를 위한 KQML 메시지의 흐름은 (그림 14)와 같다.



(그림 14) 회의 관리를 위한 KQML 메시지의 흐름

(2) 회의 준비

회의 준비는 일정 관리에서 회의 소집이 완료되어 화상회의로 넘어가는 시점에서 일어나는 작업이며, 세션 관리 에이전트는 일정 관리에서 생성된 회의 정보를 그룹 정보 관리 에이전트에게서 얻는다. 이 정보는 회의 설정, 회의 참가자 인증, 구성원 지위 제어 관리 등의 회의 관리에 유용한 정보로 사용되며, 이를 위한 KQML 메시지의 흐름은 (그림 15)와 같다.

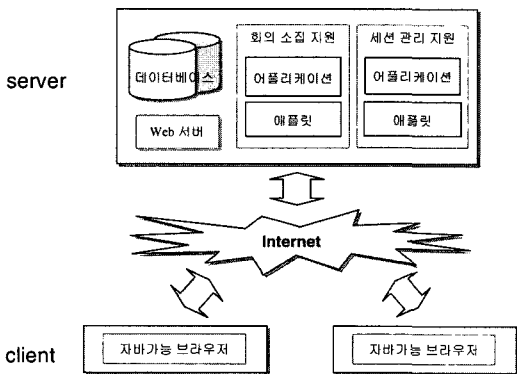


(그림 15) 회의 준비를 위한 KQML 메시지의 흐름

5. 구현 환경

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. 화상 회의를 할 경우 클라이언트 측 협동 에이전트 애플릿과 비디오/오디오 회의 에이전트 애플릿을 Web 서버를 통해 클라이언트의 자바가능 브라우저에 다운로드 시킨 후에 TCP/IP 프로토콜을 사용하여 오디오/비디오 정보를 송수신하며, 복제 구조(replicated architecture)를 가지도록 하여 서버 쪽의 통신 부하를 줄이도록 설계되었다.

구현 환경은 (그림 16)에서 보는 바와 같이 모든 응용은 서버에 위치하고 필요한 애플릿을 클라이언트의 자바가능 브라우저에 다운로드 시켜 서버와 통신을 한다. 서버용 운영체제로 Solaris 2.6.1을 사용하고 클라이언트용 운영체제로는 Windows 98을 사용한다. 데이터 베이스는 객체지향 데이터베이스인 O2를 사용한다. 개발 도구로는 JDK 1.2, JMF 2.0[17], JATLite[18]을 사용한다.



(그림 16) 시스템 구현 환경

JMF(Java Media Framework)는 자바 애플리케이션과 애플릿에서 사용되는 오디오, 비디오와 같은 실시간 데이터를 동기화하고 재생하는 통합된 구조를 제공해 주며, Media Display, Media Capture/Creation 그리고 Conferencing에 필요한 세 가지 구성 요소로 이루어져 있다. JMF는 비디오 데이터를 위해 H.261, H.263

등을 지원하며, 오디오 데이터를 위해 G.723, G.728 등을 지원한다. 또한, 실시간 데이터를 전송하는 응용에 적합한 RTP(Real-time Transport Protocol)를 제공한다. JMF에서는 RTP를 사용하여 통신하는 응용들 간의 결합을 RTP 회의(session)라 하며, 유니캐스트(unicast)를 수행하는 여러 개의 RTP 회의를 사용함으로써 IP 멀티캐스팅 없이 여러 그룹의 회의에서 멀티 캐스팅을 지원할 수 있다. 현재 JMF는 자바 소프트웨어, 실리콘 그래픽스, 인텔사가 공동으로 개발 중이다.

JATLite는 Stanford 대학에서 Java 언어를 통해 개발한 템플릿(template)이며, 자바 언어를 사용하여 KQML 메시지를 교환하는 에이전트 프레임워크 개발을 쉽게 해준다. 또한, JATLite는 AMR(Agent Message Router)를 제공한다. 이것은 멀티에이전트 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)인 다섯 가지의 계층으로 구성된다. 가장 하위의 추상 계층은 JATLite의 구현을 위해 필요한 추상 클래스의 집합을 제공한다. 기본 계층은 TCP/IP에 기초를 둔 기본 통신을 제공한다. KQML 계층은 KQML 메시지를 파싱(parsing)하고 저장하기 위한 기능을 제공한다. 라우터 계층은 에이전트들을 위한 이름 등록, 메시지의 경로를 지정하거나 큐잉(queuing)을 하는 기능을 제공한다. 프로토콜 계층은 SMTP, FTP, POP3, HTTP와 같은 표준 인터넷 서비스를 제공한다.

6. 결 론

이 연구의 목적은 웹 기반 화상회의와 에이전트 개념(agency)을 결합한 시스템에 대한 연구하는 것이다. 본 연구에서는 효율적인 회의 지원을 위해 회의 소집과 회의 관리를 편리하게 도와주는 에이전트들을 구성하였다. 본 논문에서 제시하는 시스템에서는 지능적인 에이전트가 가장 많은 사람이 참여할 수 있는 날짜를 제안해줌으로써 회의 소집자가 회의 소집에 들이는 노력을 최소화 줄일 수 있게 해 준다. 또한 회의 관리를 지원해 주는 에이전트들은 회의 소집이 결정된 이후 회의 시작 전후에 활동하며 참여자들에게 보다 편리하게 화상회의를 할 수 있도록 도와준다. 이 논문에서는

웹 기반 화상회의 시스템을 더욱 유용하게 만들어 주는 멀티에이전트들의 상호 작용 메커니즘을 제안하였다. 특히 주요 기능별 KQML 메시지 흐름에 대하여 자세하게 소개하였다.

본 시스템의 장점은 아래와 같다.

(i) 지능적 에이전트들을 통해 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 에이전트 간의 동기적 협력을 통해 참여자들의 부분 참여 가능 시간 등의 복잡한 문제를 해결할 수 있는 효율적인 기법에 대한 연구가 요구된다. 또한, 향후 폭 넓은 사용을 위해서는 인터넷 화상회의의 표준안인 H.323의 권장 내용을 포함할 수 있도록 화상회의 시스템을 수정하려 한다.

참 고 문 헌

[1] Stephen Jabele, Steven Rohall, Ralph L. Vinci-guerra, "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, p.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, No12., pp.40-47, December 1998.

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

[6] Jonathan Grudin, "Computer-Supported Cooperative Work : History and Focus," IEEE Computer, Vol.27, No.5, pp.19-26, May 1994.

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

[8] "CU-SeeME Pro," see "<http://www.wpine.com/Products/CU-SeeMe-Pro/index.html>."

[9] Walter Reinhard, Jean Schweitzer, Gerd Volksen, "CSCW Tools : Concepts and Architectures," IEEE Computer, Vol.27, No.5, pp.28-36, May 1994.

[10] Saul Greenberg and David Marwood, "Real Time Groupware as a Distributed System : Concurrency Control and its Effect on the Interface," Proceedings on CSCW '94, ACM Press, pp.165-173, October 1994.

[11] ITU-T Draft Recommendation T.120, Data Protocols for Multimedia Conferencing, August 1996.

[12] 송경준, 민병의, 황승구, 박치향, "분산협동 가상현실 미들웨어 개발", 정보과학회지 제15권 제11호, pp.20-24, 1997.11.

[13] Chris Greenhalgh and Steven Benford, "MASSIVE : A Collaborative Virtual Environment for Teleconferencing," ACM Transactions on Computer-Human Interaction, Vol.2, No.3, pp.239-261, 1995.9.

[14] Olof Hagsand, "Interactive Multiuser VEs in the DIVE System," IEEE Multimedia, Vol.3, No.1, pp.30-39, 1996.

[15] Tim Finin et al., "DRAFT Specification of the KQML Agent-Communication Language," The DARPA Knowledge Sharing Initiative External Interfaces Working Group, June 1993.

[16] Yannis Labrou and Tim Finin, "A Proposal for a new KQML Specification," TR CS-97-03, February 1997,

[17] HREF <http://www.javasoft.com/products/java-media/index.html>.

[18] HREF <http://java.stanford.edu/>.



성 미 영

e-mail : mysung@lion.inchon.ac.kr

1982년 서울대학교 식품영양학과
(학사)

계산통계학과(계산학 전공)
부진공

1987년 프랑스 INSA de Lyon 컴
퓨터공학과(공학석사)

1990년 프랑스 INSA de Lyon 전산학과(공학박사)

1990년~1993년 한국전자통신연구소 컴퓨터연구단 선
임연구원

1993년~현재 인천대학교 전자계산학과 부교수

관심분야 : 멀티미디어 협동 컴퓨팅, 에이전트, 멀티미디
어 저작, 음성 인터페이스



유 재 홍

e-mail : jhyoo@isis.inchon.ac.kr

1998년 인천대학교 전산학과
(공학사)

1998년~현재 인천대학교 전자계
산학과 석사과정

관심분야 : 멀티미디어 공동작업,
지능 에이전트, 가상현실