

분산 환경에서 공동 작업을 위한 그룹 관리 서버의 설계 및 구현

임 재 흥*

Design and Implementation of a Group Management Server for Cooperative Works in a Distributed Environment

Jae-Hong Yim

Abstract

This paper proposes a group management server which provides the information of existing groups and administrates the group session activity and the floor of shared application.

The architecture of the group management server uses two layer model. In a high layer, group manager provides the information of all working groups. In a low layer, session manager administrates the session activity and the floor of shared application.

Group agent and session agent as the user interface to access the two manager is also implemented.

1. 서론

컴퓨터와 통신 기술의 발달로 개인의 업무 능력이 증대되었을 뿐 아니라, 공동의 업무나 협조 관계에 있는 집단이 컴퓨터를 이용하여 공동 작업을 효율적으로 수행할 수 있게 되었다. CSCW (Computer Supported Cooperative Work)는 컴퓨터 공동 작업을 지원하기 위한 응용과 그 기반 플랫폼을 의미하며, 80년대 후반부터 커다란 관심의 대상이 되었다. CSCW는 지리적으로 분산된 환경에 있는 사용자들이 공동 작업을 수행할 수 있는 공유 응용 개발과 함께 조직적이고 체계적인 업무 관리와 의사 결정 시스템을 구축하여 왔다^{1, 3, 4, 5, 8)}.

컴퓨터를 이용해서 그룹 공동 작업을 하는 경우에는 그 작업 환경의 특성상 여러 가지 고려해야 할 사항들이 있다. 이는 일반적인 업무 환경과는 달리 지리적으로 분산되어 있는 사용자들이 컴퓨

* 한국해양대학교 이공대학 전자통신공학과

터를 이용하여 공동 작업을 수행하기 때문에, 의사 교환 방법이 제한적인 상황에서 작업의 목표와 작업 진행에 대한 조직적인 업무 관리가 필요하고, 필요한 정보와 자원들을 관리하고, 공유할 수 있도록 하여야 작업의 효율을 높일 수가 있다. 따라서 공동 작업 그룹들이 보다 효율적으로 업무를 수행하기 위해서는 공동 그룹에 대한 지능적이고 체계적인 관리가 필요하다. 가령 사용자들은 공동 그룹의 구성원과 작업 목표와 각 구성원의 역할과 책임을 알게 됨으로써 보다 효율적으로 공동 활동에 참여할 수가 있게 된다⁷⁾.

그룹 관리 서버는 이런 그룹들에 대한 구성 요소와 정보의 관리를 수행한다. 가령 사용자들은 그룹들에 대한 구성 요소와 정보를 얻음으로써 자신이 참여할 그룹을 결정하게 되고, 또한 특정 그룹의 참여자들은 자신들이 참여하는 그룹의 공동 작업에 관련되는 정보를 얻음으로써, 공동 작업을 효과적으로 수행하게 된다. 더 나아가 그룹 관리 서버는 그룹의 세션 활동을 관리하고, 공유 응용에 대한 입력 권한인 발언권을 관리하게 된다.

본 논문에서는 분산 환경에서 효율적인 공동 작업을 위한 그룹 관리 서버를 설계, 구현한다. 그룹 관리 서버는 2 레벨 모델을 이용하여, 상위 레벨에서 전체 그룹을 관리하는 그룹 관리자(group manager)와, 하위 레벨에서 세션 활동과 응용의 발언권 관리를 하는 세션 관리자(session manager)로 구성된다. 서버로서 동작하는 그룹 관리자와 세션 관리자를 워크스테이션에 탑재하였으며, 클라이언트로서 이들 관리자를 제어할 수 있는 그룹 에이전트(group agent)와 세션 에이전트(session agent)를 X-Window와 MS-Windows 3.1 환경에서 구현한다.

본 논문의 2장에서는 그룹 관리 서버의 기능 및 구조에 대하여 논하고, 3장에서는 그룹 관리 서버의 설계 및 구현에 대하여 기술한다. 마지막으로 4장에서 결론을 서술한다.

2. 그룹 관리 서버의 기능 및 구조

그룹웨어 시스템은 크게 그룹의 구성과 정보 및 그룹 세션 활동을 지능적으로 관리하는 그룹 관리 부분과 응용의 공유와 발언권 제어를 위한 응용 공유 서버 부분으로 나누어 진다. 본 논문에서는 그룹 관리 서버를 구축하고자 한다. 그룹 관리 서버의 기능은 크게 다음 세가지로 나누어 볼 수 있다.

2.1 그룹 관리 서버의 기능

2.1.1 다중 그룹 관리

여러 그룹이 동시에 존재하는 상황에서는, 각 그룹들에 대한 정보를 관리하고 제공해야 할 필요가 있다. 다중 그룹 관리라고 하는 것은 전체 네트워크상에 존재하는 작업 그룹들의 구성과 자원에 대한 조직적이고 체계적인 관리를 의미한다. 사용자들은 그룹 관리 서버를 통해 그룹의 구성과 작업에 대한 정보를 제공받을 수 있고, 특정 그룹 활동에 참여를 요청할 수 있으며, 그룹의 작업 상황과 목표를 알게됨으로써 그룹 협동 작업을 효율적으로 수행할 수 있게 된다. 그룹 관리 서버가

사용자들에게 제공해야 할 정보는 다음과 같다.

1) 그룹의 구성과 협동 작업에 관련된 정보

그룹은 보통 참여자들과 그들이 수행할 작업 대상과 목표, 자원 정보들로 구성된다. 따라서 협동 작업을 효율적으로 수행하기 위해서는, 그룹 관리 서버가 그룹 협동 활동과 관련되는 정보들을 관리하고 제공해야 한다. 특히 지리적으로 분산된 환경에서 참여자들이 협동 작업을 할 때에는 그룹 정보의 관리와 정보 제공에 대한 필요성이 커진다. 가령, 사용자들은 작업 그룹에 관련해서 그 작업의 책임자가 누구이며, 또한 작업에 참여하는 사람들은 누구이며, 그들의 역할과 작업의 목표는 무엇이며, 관련된 세부적인 작업들의 상호 연관성 등의 정보를 알 수 있어야 한다.

2) 공동 자원의 관리

작업의 규모가 큰 경우에는, 그룹 관리자가 그 그룹과 관련된 자원을 관리해야 한다. 사용자들은 공동 작업의 중간 산물과 결과를 공유함으로써, 불필요한 작업을 피하게 되고, 보다 효율적으로 협동 작업을 수행할 수 있게 된다. 가령 작업과 관련된 문서라든지, 화일 등을 공유하고, 작업 일정이나 작업 진척과 관련된 메시지 등을 공유하고 교환한다면 업무 능력을 보다 향상시킬 수 있게 된다.

2.1.2 세션 관리

그룹 활동은 그룹의 생성과 예약, 개시, 진행, 휴정 및 종료의 절차를 거치게 되는데 이를 하나의 세션이라고 할 수 있다. 세션은 그룹 단위로 이루어지게 되며, 이를 관리하는 것을 세션 관리라 한다. 작업 그룹은 의장의 역할을 맡은 사람이 예약을 하게 되며, 진행 시간 및 그룹 세션 목표와 작업 대상을 결정하게 된다. 의장은 정해진 시간에 그룹 활동을 개시하며, 그룹 활동은 정해진 작업 목표대로 참여자들이 자신의 역할을 수행함으로써 이루어진다. 작업의 목표를 완수하게 되면 그 그룹의 세션은 종료된다. 또한 작업이 여러 세션으로 구성되어 나누어진 경우에는 세션은 종료되지 않고, 대기나 휴정 상태에 들어가게 된다.

세션 관리에는 또한 참여자 관리가 포함된다. 참여자는 그룹이 구성될 때 주로 결정되며, 의장의 참여 요청에 의해서든지, 혹은 구성원들의 자발적인 참여에 의해서 결정된다. 참여자 관리에서 가장 중요한 문제가 되는 것은 세션이 진행 중인 상황일 때, 참여자들의 가입과 탈퇴를 자연스럽게 처리하는 것이다. 세션이 진행 중일때, 참여자가 새로 가입한 경우에는 주어진 윈도우 시스템이 다른 참여자들과 동일하게 만들어야 한다. 곧 공유된 응용이 새로운 참여자에게도 동일한 상태로 제공이 되고, 발언권의 권한이 부여되어야 한다. 이를 위해서는 응용 프로그램의 공유 메카니즘에 대한 고찰이 필요하다.

2.1.3 공유 응용에 대한 발언권 제어

여러 사용자들이 응용 프로그램을 공유하는 경우에는 그 응용 프로그램에 대한 참여자들의 발언권 제어가 필요하다. 발언권은 사용자의 응용 프로그램에 대한 입력 권한이다. 발언권 제어 방식에는 방임형, 선점형(preemptive), 의장형, FIFO형, 토큰형 등이 있다.

2.2 그룹 관리 서버의 구조

기존 CSCW 환경에서의 그룹 관리는 주로 그룹의 세션 활동과 그룹 공동 작업을 지원하는 세션 관리를 의미하였다. 그러나 다수의 그룹이 동시에 존재하는 다중 그룹 환경에서는 세션 관리뿐 아니라 전체 그룹 관리를 제공해야 한다. 전체 그룹 관리를 해 주어야 사용자들은 존재하는 그룹들에 대한 정보를 제공받을 수가 있다. 이와 같은 기능을 고려하여 본 논문에서 설계하고 제안한 그룹 관리 서버의 기능은 다음과 같다.

- 그룹의 예약과 그룹의 구성과 자원에 대한 정보 입력
- 그룹 상태에 따른 그룹 관리와 정보의 관리
- 그룹에 참여할 참가자들의 그룹 참여 신청과 등록
- 그룹 활동의 개시와 휴정과 종료
- 그룹 활동에 필요한 공유 응용에 대한 발언권 제어와 관리
- 발언권 제어 서버와 응용 공유 서버의 구동

이러한 기능들을 제공하기 위하여 본 논문에서 설계, 구현한 그룹 관리 서버는 그룹 관리 레벨을 두개의 레벨로 나누고, 다중 그룹 관리 기능과 세션 관리 기능을 구별하였다. 상위 레벨에서는 전체 그룹 관리를 하는 그룹 관리자를 두고 하위 레벨에서는 각 그룹의 세션들을 관리하는 세션 관리자를 두었다. 그림 1은 그룹 관리 서버 모델을 나타내며, 그림 2는 그룹 관리 서버가 구축되고 그룹 세션이 개시되었을 때, 각 시스템에서 구동되는 경우를 나타낸다.

그룹 관리자는 전체 그룹의 관리를 담당하며, 전체 시스템에서 하나만 존재한다. 또한 각 시스템마다 하나의 그룹 에이전트가 위치하여 사용자들이 그룹 관리자에게 그룹 정보를 요청하거나 그룹 등록과 그룹 참여를 요청할 때 사용된다. 이를 위하여 그룹 에이전트는 GUI(Graphical User

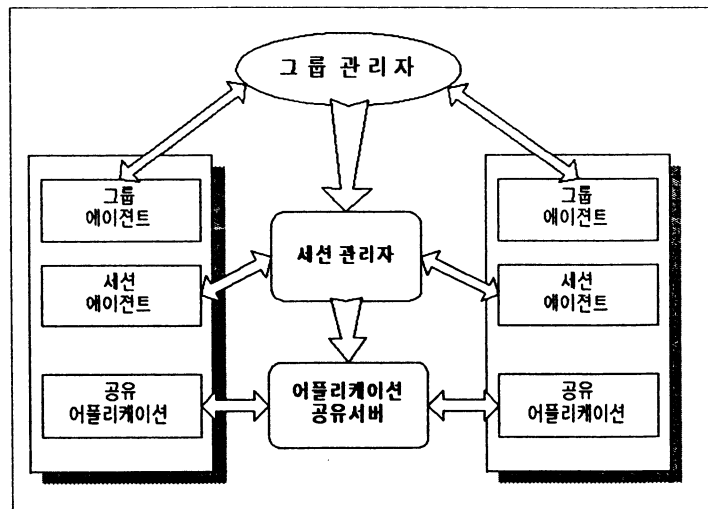


그림 1. 그룹 관리 서버 모델

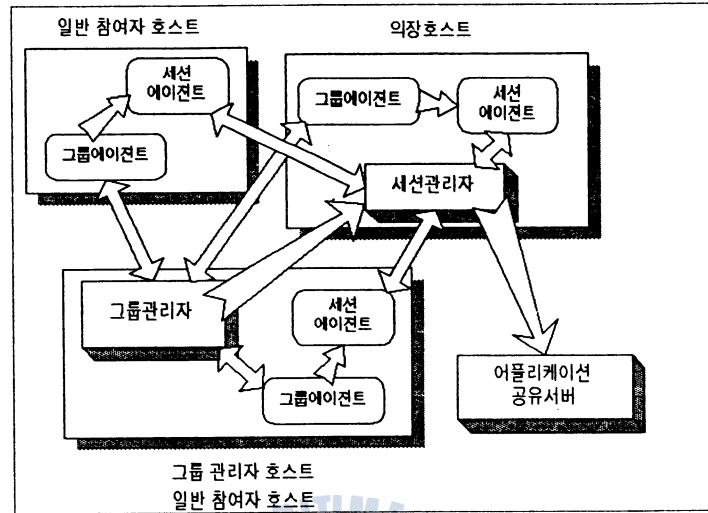


그림 2. 그룹 관리 서버의 구동 상황

Interface) 형태의 인터페이스를 제공한다. 세션이 시작되면 세션 관리자가 구동되고, 사용자는 세션 에이전트를 통해 응용 프로그램에 대한 발언권을 요청한다. 세션 관리자는 의장이 결정한 발언권 제어 방식에 의해 각 응용 프로그램에 대한 발언권을 결정하고, 이 정보를 응용 공유 서버에게 제공한다. 응용 공유 서버는 발언권을 가진 사용자의 입력만 처리하게 된다. 그룹 관리자와 세션 관리자에 대한 세부적인 기능은 다음과 같다.

2.2.1 그룹 관리자

그룹 관리자는 전체 그룹들을 관리한다. 그룹들은 그 세션 상태에 따라서 관리된다. 곧 그룹은 예약 상태, 진행 상태, 휴정 상태로 나눌 수 있고, 그룹 관리자는 이 세 가지 상태에 의해 그룹들을 관리한다. 그룹 관리자는 각 그룹들의 구성 요소와 정보를 관리한다. 그룹 관리자의 기능을 분류해 보면 다음과 같다.

1) 그룹 관리

그룹 관리는 현재 작업 중인 그룹들과 예약, 휴정된 그룹들에 대해 이루어진다. 그룹 관리자는 그룹들의 그룹 정보와 구성 요소에 대한 등록 요청을 처리하고, 이들 정보에 대한 정보 제공의 기능을 한다. 그룹 구성 요소와 정보에는 그룹 타이틀과 의장, 의장 호스트, 작업 목표, 참여자 역할, 그룹 활동 개시 시간, 기타 그룹 활동과 관련된 메모나 노트 등이 있다. 또한 그룹 활동이 종료된 그룹에 대해서는 그룹을 삭제시킨다.

2) 그룹 참가자 등록

그룹에 참여하고자 하는 사람은 그룹 에이전트를 통하여 그룹 관리자에게 그룹 참여를 요청하며, 그룹 관리자는 요청에 따라 참여자를 그룹에 등록시키고 그들의 호스트에 대한 정보와 역할을 등록시키게 된다.

3) 세션 관리자와 응용 공유 서버의 관리

그룹 관리자는 세션이 개시될 때, 세션 관리자에게 그룹 구성 요소와 정보를 제공하여 세션이 개시될 수 있도록 초기화시키며, 세션의 변동 요소를 관리하게 된다. 가령 그룹의 종료나 그룹 휴정 등의 상태 변동을 관리하며, 의장이나 참여자나 그룹 세션 활동의 변경 등의 변동 요소를 관리하게 된다. 동시에 응용 공유 서버에게 참여자와 호스트 정보를 제공하여 응용 프로그램의 공유를 초기화시키고, 새로운 참여자 등록이나 탈퇴가 있을 때, 이들 변동 사항을 응용 공유 서버에게 제공하게 된다. 그룹 관리자는 진행 중인 그룹에 대해서도 참여자가 새로 등록할 수 있도록 설계되었다.

2.2.2 세션 관리자

세션 관리자는 활동중인 그룹에 하나씩 존재하며 그룹 활동의 세션을 관리하게 된다. 세션 관리자는 세션 활동이 시작되면서 구동되고, 세션 활동이 끝남과 동시에 종료된다. 세션 관리자의 기능은 크게 두 가지로 구분되며, 그 하나는 그룹 활동과 관련된 세션 관리이며, 다른 하나는 응용 프로그램의 발언권 관리 기능이다. 이들의 기능은 다음과 같다.

1) 세션 관리

그룹 활동은 의장이 세션을 개시하게 되면 시작된다. 의장은 정해진 시간에 세션을 시작하며, 세션의 진행과 휴정, 세션의 종료에 대한 제어 권한을 가지게 된다. 세션이 진행 중인 때도 새로운 참여자의 가입이 허용되며, 동일한 상태의 응용 프로그램을 공유하게 된다.

2) 응용 프로그램의 발언권 제어

세션 관리자는 공동 작업시 이용되는 공동 응용 프로그램에 대한 발언권을 제어한다. 응용 프로그램의 발언권은 사용자들이 공동 응용 프로그램을 이용할 때, 각각의 응용에 대한 입력 권한으로서, 발언권이 있는 사용자의 입력 메시지만이 응용 공유 서버에게 전달된다. 세션 관리자는 응용 공유 서버에게 응용 프로그램에 대한 발언권을 가진 사람의 이름과 호스트를 제공하며, 응용 공유 서버는 이 발언권 정보를 토대로 발언권을 가진 사용자의 입력만 받아들이고, 다른 참여자들의 입력 정보는 무시하게 된다. 현재 응용 프로그램에 대한 발언권 제어 방식은 의장이 결정하게 되며, 다음과 같은 발언권 제어 방식이 있다.

- 선점형(preemptive) : 발언권을 요청하는 사용자에게 발언권이 즉시 넘어간다.
- 선입선점형(FIFO) : 사용자들의 발언권은 요청 순서에 의해 큐에 저장되며, 발언권을 소유하고 있는 사용자가 발언권을 포기할 때, 큐의 가장 처음에 있는 사용자가 발언권을 넘겨받게 된다.
- 의장형(master) : 세션의 의장이 발언권을 임의로 제어하게 된다.
- 방임형(free) : 발언권은 모든 사용자에게 주어진다.

3. 그룹 관리 서버의 설계 및 구현

본 논문에서의 그룹 관리 서버 개발 환경은 워크스테이션과 PC를 LAN으로 연결하여 TCP/IP (Transmission Control Protocol/Internet Protocol)를 이용하였다. 워크스테이션에서 운영되는 UNIX의 경우에는 커널(kernel)에 TCP/IP가 내장되어 있으며, PC에서는 TCP/IP 스택이 구현된 winsock 패키지를 사용하였다. 따라서 TCP/IP를 사용하는 경우에는 특별한 제약이 없는 한 시스템의 확장이 가능하다.

3.1 그룹 관리자

그룹 관리자는 각 사용자들의 인터페이스 역할을 하는 그룹 에이전트와 통신하며, 각 에이전트로부터 요청이 있는 경우에 필요한 그룹 구성 요소와 정보를 제공하거나, 등록 요청을 받아 들어 그룹과 참여자들을 등록시킨다. 또한 그룹 관리자는 공유 응용 프로그램을 관리하는 응용 공유 서버에게 참여자의 이름과 호스트 정보를 제공하며, 세션 관리자에게 필요한 그룹 정보와 구성 요소를 제공하여 세션 관리를 초기화시킨다.

그룹 관리자가 그룹 정보를 저장하는 데이터 구조와 패킷 구조는 다음과 같다.

3.1.1 데이터 구조

그룹 관리자는 다음과 같은 구조체에 그룹의 정보를 저장하며, 그룹들은 각각 RESERVED, WORKING, SUSPENDED 상태에 따라 관리된다.

```
struct  master {
    char  name[MAX_NAME];
    char host[MAX_HOST];
};
struct  participant {
    char name[MAX_NAME];
    char host[MAX_HOST];
    struct participant *next;
};
typedef struct  group {
    char title[MAX_TITLE];
    struct master  chair;
    char subject[MAX_SUBJECT];
    char time[MAX_TIME];
    struct participant *p_head;
```

```

struct participant *p_tail;
char file[MAX_FILE];
struct group *next;
} GROUP;

```

3.1.2 패킷 구조

그룹 관리자와 그룹 에이전트간에 주고 받는 메시지의 패킷 구조는 다음과 같다.

```

struct add_group {
char title(MAX_TITLE);
char master(MAX_NAME);
char host(MAX_HOST);
char subject(MAX_SUBJECT);
char time(MAX_TIME);
};
struct _join_request {
char title(MAX_TITLE);
char name(MAX_NAME);
char host(MAX_HOST);
};
struct _memo_save {
char title(MAX_TITLE);
int length;
};
union _union_data {
struct add_group add_data;
struct _join_request join_data;
struct _memo_save memo_save;
char group_title(MAX_TITLE);
};
typedef struct message {
int head;
int flag;
union _union_data m_data;
} MESSAGE;

```

3.2 세션 관리자

세션 관리자는 세션 에이전트로부터 발언권 요청을 받으며, 이 요청을 큐에 저장하고 자신이 가

분산 환경에서 공동 작업을 위한 그룹 관리 서버의 설계 및 구현

지고 있는 발언권 제어 방식에 의해 발언권 정보를 응용 공유 서버에게 제공한다. 사용자들은 세션 에이전트를 통해 세션 관리자에게 발언권을 요청하거나 포기하게 된다.

3.2.1 데이터 구조

세션 관리자는 응용 프로그램에 대한 발언권을 요청한 회의 참가자와 호스트에 대한 정보를 관리한다. 참가자와 호스트 정보는 큐에 저장되며, 큐는 실행되는 응용 프로그램 마다 관리된다. 큐의 구조는 다음과 같다.

```
typedef struct queue {
    char name[MAX_NAME];
    char host[MAX_HOST];
    char data[MAX_DATA];
    struct queue *next;
} QUEUE;
```

3.2.2 패킷 구조

세션 관리자와 세션 에이전트간에 주고 받는 메시지의 패킷 구조는 다음과 같다.

```
typedef struct _floor_message {
    int head;
    int flag;
    int app;
    char name[MAX_NAME];
    char host[MAX_HOST];
    char token_name[MAX_NAME];
    char token_host[MAX_HOST];
} FLOOR_MESSAGE;

typedef struct _queue_message {
    int head;
    int flag;
    int floor_mode;
    char token_name[MAX_NAME];
    char token_host[MAX_HOST];
    char queue[2*MAX_USER][MAX_NAME];
} QUEUE_MESSAGE;
```

세션 관리자가 응용 공유 서버와 주고 받는 패킷 구조는 다음과 같다.

```
typedef struct _floor_inform {
```

```

int head;
int flag;
int app;
char name(MAX_NAME);
char host(MAX_HOST);
} FLOOR_INFORM;
typedef struct _member_list {
int head;
char user(2*MAX_MEMBER)(MAX_NAME);
} MEMBER_LIST;
    
```

3.3 사용자 인터페이스

3.3.1 그룹 에이전트

그룹 에이전트는 그룹의 상태에 따라 전체 그룹에 대한 리스트와 그룹의 구성 요소와 그룹 정보를 보여 준다. 그림 3에서 보듯이 그룹 에이전트는 그룹의 세 가지 상태에 따라 그룹의 리스트를 보여 주며, 그룹들의 구성 요소와 정보를 보여 주게 된다.

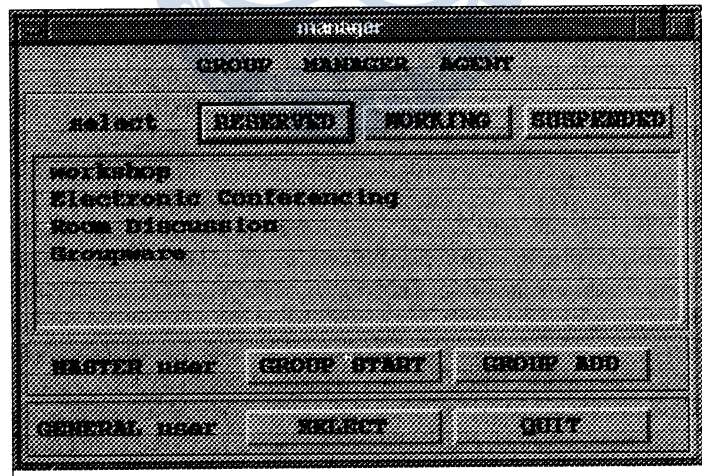


그림 3. 그룹 에이전트

3.3.2 세션 에이전트

세션 에이전트는 의장과 일반 참여자의 권한이 차별될 수 있도록 하였다. 의장은 세션의 시작과 휴정과 종료를 제어하며, 그룹 구성 요소의 변경을 제어할 수 있다. 또한 의장은 세션이 시작되면 자신의 호스트에 세션 관리자와 응용 공유 서버를 구동시키게 된다.

분산 환경에서 공동 작업을 위한 그룹 관리 서버의 설계 및 구현

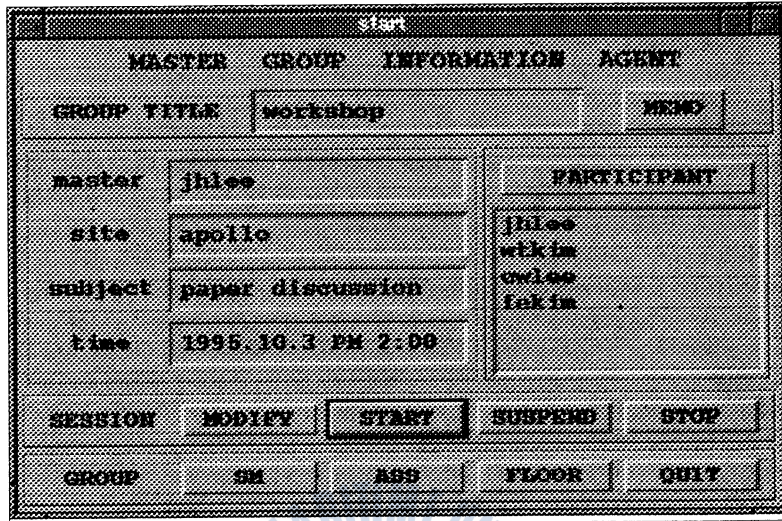


그림 4. 세션 에이전트

발언권은 응용 프로그램마다 관리되어야 하며, 세션 에이전트는 각 응용 프로그램에 대한 발언권 제어 방식과 현재 발언권 소유자와 큐에 등록된 사용자 리스트를 보여주게 된다. 또한 의장은 발언권 제어 방식을 선택할 수 있는 권한을 가지게 되며, 의장형인 경우 의장은 임의대로 발언권을 사용자에게 넘겨주게 된다.

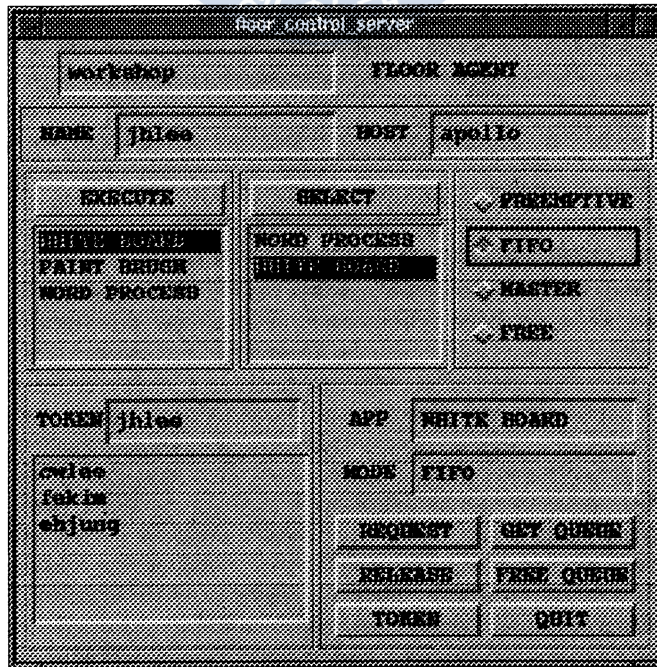


그림 5. 발언권 제어

4. 결 론

본 논문에서는 다중 그룹 환경에서 그룹 관리 서버를 설계, 구현하였다. 그룹 관리 서버는 2 레벨 모델을 이용하여, 상위 레벨에서 전체 그룹을 관리하는 그룹 관리자를, 하위 레벨에서 세션 활동과 응용 프로그램의 발언권 관리를 하는 세션 관리자를 구현하였다. 서버로서 동작하는 그룹 관리자와 세션 관리자를 워크스테이션에 탑재하였으며, 클라이언트로서 이들 관리자를 제어할 수 있는 그룹 에이전트와 세션 에이전트를 X-Window와 MS-Windows 3.1 환경에서 구현하였다. 그룹 관리 서버는 다중 그룹 환경에서 전체 그룹들의 구성 요소와 정보를 관리하며, 각 그룹들에 대한 세션 활동과 응용 프로그램 발언권 제어를 수행한다. 향후 연구 과제로서는 그룹 정보와 자원을 객체화하고 이를 관리하는 디렉토리 서비스에 대한 연구가 필요하다.

참고 문헌

- 1) Thomas Kreifelts, Wolfgang Prinz, "ASCW: An Assistant for Cooperative Work," COOCS'93 Proc. Conf. on Organizational Computing Systems, 1993.
- 2) T. Ohmori, K. Maeno, S. Sakarta, "Distributed Cooperative Control for Application Sharing Based on Multiparty and Multimedia Desktop Conferencing System: MERMAID," Proc. of 12th International Conference on Distributed Computing Systems, 1992.
- 3) Michael Altenhofen, Jurgen Ditrich, Rainer Hammerschmidt, "The BERKOM Multimedia Collaboration Service," ACM Multimedia, 1993.
- 4) Hania Gajewska, Jay Kistler, Mark S. Manasse, David D. Redell, "Argo: A System for Distributed Collaboration," ACM Multimedia, 1994.
- 5) Vinod Anupam and Chandrajit L. Bajaj, "Shastra: Multimedia Collaborative Design Environment," IEEE Multimedia, 1994.
- 6) Kankanahalli Srinivas, Ramana Reddy, Aliasghar Babadi, "MONET: A Multi-Media System for Conferencing and Application Sharing in Distributed Systems," CERC Technical Report Series Research Note, CERC-TR-RN-91-009, 1992.
- 7) Ting-Peng Liang, Hsiangchu Lai and Nian-Shing Chen, "When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks," IEEE COMPUTER, May 1994.
- 8) Chang-Wook Lee, Yong-Jin Park, "Open Platform for Group-Working in ODP environments," ICCS 1995.
- 9) Ian Smith, Elizabeth Mynatt, "What You See Is What I want: Experience With the Virtual X Shared Window System," Technical Report, Jorzea Tech. College, 1994.
- 10) Richard Bentley, Tom Rodden, Peter Sawyer and Ian Sommerville, "An Architecture for tailoring cooperative multi-user displays," CSCW 92 Proceedings.
- 11) Neil Williams, Gordon S Blair, "Distributed Multimedia Applications: A Review," Computer Communications, Vol. 17, No. 2, Feb. 1994.
- 12) Stephen Zabele, Steven L. Rohall, Ralph L. Vinciguerra, "High Performance Infrastructure for Visually-intensive CSCW Applications," CSCW 1994.
- 13) Walter Reinhard, Jean Schweitzer, and Gerd Volksen, "CSCW Tools: Concepts and Architectures," IEEE COMPUTER, 1994.