

# 소프트웨어 개발 생산성 향상을 위한 공동 작업 플랫폼 설계

조 성 빈<sup>†</sup> · 김 진 석<sup>†</sup>

## 요 약

최근에 개발되고 있는 소프트웨어들은 규모가 거대해지고 기능도 복잡 해지고 있다. 이처럼 복잡한 시스템의 개발을 위해 장기간에 걸쳐 서로 다른 전문가들의 협력 작업과 개발팀 구성원간에 정보 공유, 상호의존적 업무 수행, 공동의 산출물 생성 등의 작업을 지원하는 CSCW(Computer Supported Cooperative Works) 개념을 소프트웨어 개발 방법에 통합하는 연구가 진행 중에 있다. 본 논문에서는 소프트웨어 개발에 있어서의 공동 작업 특성과 분산 컴퓨팅 환경에 의한 공동 작업을 지원하여 생산성을 향상 시키는 CSCW 개념을 적용하여 정보 통신 소프트웨어 개발을 지원하는 하부 구조로서 다자간 공동작업 플랫폼인 SPACE(Software Platform for distributed Application sharing under Cooperative Environment)를 설계하였다. SPACE는 Windows 95 환경 하에서 완전 분산 구조를 가지고 있으며, 개발 도구를 공유하게 하는 공유 기법으로 중앙 집중형 구조를 갖는 GUI 공유 방식과 복제 구조를 지원하기 위하여 이벤트 공유 방식을 사용하고 있다.

## A Design of Cooperative Works Platform for Software Development Productivity

Sung Been Cho<sup>†</sup> · Jin Suk Kim<sup>†</sup>

### ABSTRACT

Today's Software Systems are becoming bigger, larger and more complex than ever before. To develop larger-scale projects, it is required that many experts of different fields participate and cooperate in the same project. So, it is an applied area of CSCW(Computer Supported Cooperative Works) that centers around methodologies and tools that enable cooperative works by geographically distributed people engaged in all aspects of product development. In this paper, we designed a multi-user cooperative works platform, SPACE(Software Platform for distributed Application sharing under Cooperative Environment) as a infrastructure that support to CSCW based system development for telecommunication and information system. SPACE has a fully distributed architecture under Windows 95 environment, has an application sharing mechanism enabling collaborative use of interactive application adapt to a mixed GUI sharing technology which capture GUI and screen information, and also, an event sharing technology that has a replicated architecture.

### 1. 서 론

최근의 정보화 사회로의 급격한 변화 속에 계속적

인 기술개발에 대한 경쟁이 그 어느 때보다 치열하게 전개되고 있다. 특히, 최근에 개발되고 있는 소프트웨어의 규모가 거대해지고 기능이 복잡해지고, 그리고 인간 사회 사이의 상호 교류가 더 많아지고 밀접해짐에 따라 지리적으로 분산된 전문가 그룹들 사이에 컴

<sup>†</sup> 정 회 원 : 한국전자통신연구원  
논문접수: 1997년 6월 24일, 심사완료: 1997년 11월 17일

퓨터와 네트워크를 사용한 새로운 형태의 공동 작업 형태로 변화하고 있다.

소프트웨어를 효율적으로 개발하고 품질을 관리하기 위한 기술개발은 1980년 후반과 1990년대부터 CASE(Computer Assisted Software Engineering) 연구에 초점을 두고 진행되어 왔다. 그러나 초기의 연구는 CASE 도구가 소프트웨어 개발을 위한 명세, 설계, 구축을 돕는 도구로서 구성원들간의 공동작업은 고려하지 않은 고립형으로 단위 사용자만을 지원하여 왔다.

DeMarco와 Lister는 “대규모 프로젝트에서 시스템 개발자들은 다른 사람과 작업하는데 70% 이상의 시간을 소비하고 있다.”는 연구 결과와 Jones는 “팀 활동이 비용의 85%를 차지하고 있다.”는 연구 결과를 발표하였다[2][8].

이처럼 복잡한 시스템 개발은 장기간에 걸쳐 서로 다른 전문가들의 협력 작업을 요구하고 있으며, 정보 공유, 상호의존적 업무 수행, 공동의 산출물 생성 등의 작업을 지원하도록 하는 CSCW(Computer Supported Cooperative Works) 개념을 소프트웨어 개발 방법에 통합하는 연구가 진행 중에 있다[2].

CSCW는 1986년 개최한 학술대회에서 새로운 분야로 대두되었으며, 사회적 활동의 대부분이 조직의 협조 작업(Cooperative Works)이고, 이러한 작업들이 정보 시스템 및 통신 시스템으로 지원되면(Computer Supported) 그 생산성 및 창조성을 비약적으로 높일 수 있다는 것이다. CSCW의 연구 영역은 그룹 작업 그 자체의 연구인 그룹 프로세스와 그룹 작업을 지원하는 기술 두 가지로 나눌 수 있다.

그룹 프로세스에서는 그룹의 프로세스를 이해하기 위하여 인간 개인의 특성, 기능, 지식이 그룹 작업에 미치는 영향에 관한 연구, 그룹의 크기, 구성, 운영 등의 조직적인 관점에 관한 연구, 그룹 작업을 필요로 하는 시스템 설계 방법에 관한 연구, 사람, 자원, 작업의 관리 운영 방법에 관한 연구 등이 있다[18].

본 논문은 종래의 고립형 도구의 영역과는 달리 분산 컴퓨팅 환경에서 네트워크를 매체로 데이터와 메시지를 주고 받으며 동작하는 정보 통신용 소프트웨어 개발을 지원하여 대규모 소프트웨어 개발 시 지리적으로 분산된 팀 구성원, 팀 대 팀의 협력 작업을 서로 공유할 수 있는 상호 의존적인 작업 환경을 제공

하여 지리적으로 분산됨으로써 발생하는 작업 손실을 줄여 생산성을 향상시키는 다자간 공동작업 플랫폼인 SPACE(Software Platform for distributed Application Sharing under Cooperative Environment)를 설계하였다.

SPACE는 분산 환경에서 다수의 정보 통신 시스템 개발자가 개발 도구와 정보를 공유하여 동시에 공동작업을 할 수 있는 플랫폼으로써 다음과 같은 특성을 가지고 있다.

- 정보 통신 시스템을 개발하는 최종 사용자가 스스로 정의한 프로세스 중심의 개발 단계와 이에 관련된 도구의 사용을 지원한다.
- 시스템에 의해 고정된 특정한 개발 방법론 및 도구를 지원한다.
- 분산된 사용자 간에 시스템 개발 작업과 병행하여 화상 회의 등 의사 전달 수단을 지원한다.
- 개발 방법론 및 도구에 대한 깊은 지식이 없는 일반 사용자가 GUI 등의 시각 기반의 인터페이스를 통하여 정보 통신 시스템을 개발할 수 있도록 설계되어 있다.
- 자체 개발된 개발 도구는 물론, 기존의 개발 도구와도 호환성을 유지하고 있다.

## 2. 관련 연구

문제 해결을 위하여 소프트웨어의 컴퓨터 지원 공동작업을 가능하게 하는 접근 방법에는 3가지 접근 방법이 있다.

첫째로는 중앙 집중형 공동 작업 투명성(Centralized Collaboration-transparent) 방식으로 이는 단위 사용자의 소프트웨어 인스턴스를 특정 화면 혹은 화면 공유 매커니즘에 의해서 다자간 사용자에 공유하도록 해주는 것이다. 이 접근 방식은 공유 어플리케이션의 윈도우 인터페이스를 복제함으로써 도구들을 공유하게 한다. 다양한 출력장치에서 이벤트 스트림을 가로채고 동시에 윈도우를 구동하는 매커니즘을 사용하며 대표적인 시스템으로는 SharedX, XTV와 COMIX가 있다[1][4]. 또한, 공동작업 투명성 어플리케이션에서 발언권 제어, 주석, 텔리포인팅, 공유 워크스페이스 관리, 자발적인 상호 동작을 지원하는 윈도우 시스템으로는 MMConf, Matrix, MONET, CECED,

BERKOM과 MObView가 있다. 이 시스템들은 윈도우 공유 기술을 사용한 데스크 탑 화상회의 시스템이다. 이러한 접근 방법의 장점으로는 구현하기 쉽다. 그러나 도구가 다자간 상호 동작을 지원하지 않게 설계되었다면 사용자는 교대로 사용해야 하며, 뷰 생성을 중앙 시스템에서 해야 하고, 도구가 중앙 시스템에 있어야 하므로 네트워크 트래픽이 심하다는 단점이 있다[9][11][12][15].

두번째 접근 방법으로는 중앙 집중형 공동작업 인지(Centralized Collaboration-aware) 방식으로 이는 다자간 소프트웨어 인스턴스가 다양한 인터페이스를 구동하므로써 다자간에 공유토록 해주는 방식이다. Rendezvous와 Weasel 같은 시스템이 대표적인 시스템이며, 이 방식의 장점은 도구가 내부적으로 발언권 제어 메커니즘을 구현하므로써 동시에 다자간 사용자들의 상호 동작을 가능하게 한다. 단점으로는 이기종 분산 환경에서는 하드웨어 종속적인 특징 때문에 복잡성이 증가하게 되고, 모든 이벤트가 중앙 시스템을 통하여 처리되기 때문에 네트워크 트래픽에 문제점이 있다[7].

세번째 접근 방법으로는 복제 공동작업 인지(Replicated Collaboration-aware) 방식으로 이는 분산 환경에서 도구의 인스턴스를 가지고 있으며 각각은 국부 인터페이스가 있고 공유 태스크를 접근하여 공동작업을 하며 입출력 일치성과 어플리케이션에 대한 구동 일치성을 보장한다[6].

MMConf, LIZA와 GroupKit은 복제 구조 시스템을 구현하는 기능을 제공하고 있으며 이의 대표적인 시스템은 Rapport와 Diamond 등이 있다. 이 방식의 장점으로는 도구들 내부에 발언권 제어 메커니즘 구현이 가능하며 동시에 다자간 사용자들에게 상호 동작을 가능하게 한다. 단점으로는 공유 상태와 상호 동작 상태를 유지하는 복잡성 증가와 동시 접근 제어가 어렵고 사용자 수가 증가함에 따라 수행 속도에 문제가 야기될 수 있다[13].

공동작업 분야에서의 공유 멀티미디어 환경에 대한 연구는 MONET, Rapport, CECED와 MERMAID와 같은 시스템의 결과가 있다[10]. X 윈도우를 사용하여 분산 컴퓨터 음성 어플리케이션에 대한 네트워크 투명성 시스템인 AudioFile은 단순 네트워크 프로토콜에 대한 추상화 오디오 장치 인터페이스를 제공하고 있

다[16]. 애플 매킨토시 시스템 7 Inter 어플리케이션 프로토콜 기반 KWrite 시스템은 공유 문서에 그림 파일 형태로 사용자들 사이의 상호 동작을 가능하게 하는 멀티미디어 문서에 대한 개방형 구조를 취하고 있다. 화면과 네트워크 같은 외부 엔티티에게 멀티미디어 자료 전달 기능을 가지고 있는 멀티미디어 객체 개념을 적용한 Gibbs는 멀티미디어 어플리케이션의 프로토타이핑 방법을 제시하고 있고 컴포넌트 지향 뷰를 채택하고 있다. Standalone 네트워크 기반 비디오 정보 캡처, 텔레비전 화면의 캡처와 전송 능력을 갖춘 외부 장치 처리를 하는 VidBoard 시스템이 있다[5].

동시성 공학 분야는 소프트웨어 개발의 모든 개발 단계에 참여하고 있는 지리적으로 분산된 개발자들에 의해서 공동작업 의사결정을 가능하게 하는 개발 방법과 도구에 초점을 맞춘 CSCW의 적용 분야이다. 이 분야의 주된 관심사는 공동작업, 정보 공유, 통합, 공동작업 조정 등이고 대표적인 시스템으로는 DICE (DARPA Initiative in Concurrent Engineering), PACT (Palo Alto Collaborative Testbed), SWIFT(Software Workbench for Integration & Facilitating Teams), SHARE, CORBA(Common Object Request Broker Architecture)와 DCE(Distributed Computing Environment)가 있다[3][14].

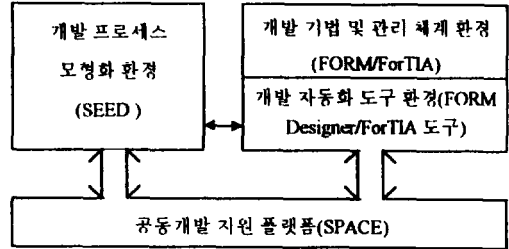
DICE는 현존하는 이기종 도구의 재사용과 결합을 강조하고 있으며 많은 프로젝트에서 상용하고 있다. PACT는 공동작업 공학 환경을 연결하고, 다른 프로젝트에서 사용 가능하게 하며, 이의 구조는 대리자(agent)로서 기능들을 캡슐화하였다. SWIFT는 그룹 문제 해결을 위한 생산성 향상을 목적으로 현재 개발 중에 있고 기존의 기능들을 재 구축함으로써 빠른 어플리케이션 구축을 가능하게 하는 지식 계층과 커널 계층으로 구성되어 있다[14]. SHARE는 정보 기술을 설계 팀 구축, 접근 그리고 설계 정보를 전달하도록 하는데 적용하는 동시성 공학 환경을 제공하고 있으며 기존의 소프트웨어 도구 지원, 멀티미디어 전자 메일, 다자간 사용자 사이의 어플리케이션 공유를 위한 윈도우 공유 기법과 ServiceMail을 사용하여 메일 기반 도구 상호 운용성을 목표로 현재 구축 중에 있다[3]. CORBA와 DCE는 특정 플랫폼과 무관하게 시스템을 구축할 수 있게 하는 국제 표준 분산 시스템이다.

SPACE는 정보통신 시스템의 컴퓨터 지원 공동작업을 가능하게 하는 접근 방식으로 중앙 집중형 공동작업 투명성 방식과 복제 공동작업 인지 방식 2가지 방식을 지원하는 복합 구조를 가진다. 소프트웨어 개발자간의 공동작업 시 공유 어플리케이션을 개발자 사이트들에서 위의 2가지 방식에서 하나의 방식을 자동으로 인지하여 공유하고자 하는 어플리케이션의 화면 공유를 통하여 공동작업을 지원하거나, 개발자 입력 이벤트를 추출하여 각 구부 어플리케이션에서 수행하도록 하여 공동작업을 지원하여 입출력 및 구동 일치성을 가능하게 한다. 또한, SPACE는 발언권 제어, 주석, 텔리폰팅, 공동 작업 개발자간의 소프트웨어 개발 시 원활한 의사소통을 위하여 화상회의, 멀티미디어 메일 기능을 가지고 있다. 이기종간 상호운용성을 위하여 데이터 회의 국제 표준인 IMTC(International Multimedia Teleconferencing Consortium) 산하의 T.120 시리즈를 근간으로 현재 프로토타입 시스템 개발 중에 있다.

**3. 분산 환경에서의 소프트웨어 공동 개발 모형**

SPACE의 상위 계층에는 소프트웨어 개발 프로세스를 프로젝트 유형에 따라 모형화하고 실행할 수 있는 환경을 제공하는 프로세스 모형화 환경인 SEED (Software Engineering Environment for software process Development)가 있다. 그리고 정보 통신 시스템의 실시간 및 분산 처리 등의 특성을 쉽게 모델링하면서 생명주기 전 개발 과정을 지원하는 개발 기법 및 관리 체계 환경이 있다. 여기에서의 개발 기법 및 관리 체계는 객체 지향 시스템 개발 방법인 FORM (Formal Object-Oriented & Reliable Method)과 정형 기법을 근간으로 하는 ForTIA(Formalism for Telematics and Information systems Applications) 방법을 제공한다. 또한, 이들의 방법을 지원하는 개발 자동화 도구 환경에는 객체지향 방법을 위한 도구인 FORM Designer와 정형 방법을 위한 도구인 ForTIA가 있다. FORM Designer의 도구에는 시스템 모형 관리기, 모형 편집기, 모형 검증기, 정보 저장소, 객체 조회기, 코드 생성기, 보고서 생성기 등이 있으며 ForTIA의 도구에는 구문 편집기, 검증기, 코드 생성기, 화일 변환기, 시험기, 시뮬레이터, 문서 생성기 등이 있다. 이

러한 환경과 SPACE와의 상호 관계는 (그림 1)과 같다.



(그림 1) 정보 통신 시스템 개발 모형  
(Fig. 1) Development model for information telecommunication system

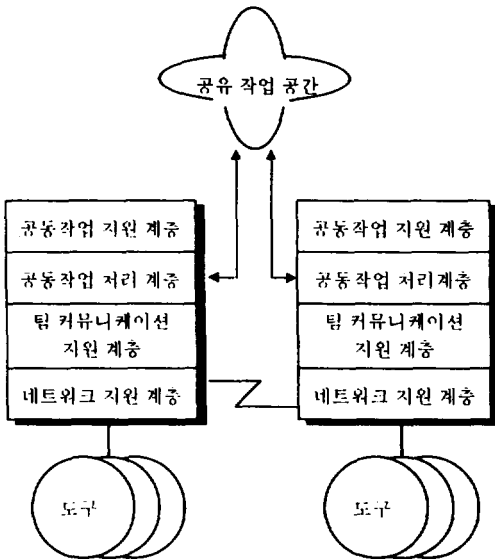
정보 통신 시스템 개발을 위한 공동개발 지원 환경인 SPACE는 개발 프로세스 모형화 환경과 개발 자동화 지원 도구 환경에서 소프트웨어 개발 생명주기에 의한 프로세스에 의해서, 혹은 단위 도구에 의해서 여러 명의 개발자들이 도구와 정보를 공유토록 하는 공동 작업을 지원한다.

SPACE는 Windows 95 환경 하에서 완전 분산 구조를 기반으로 하는 클라이언트/서버 모델을 가지고 있으며, 공유할 도구의 유무에 따라 서버, 클라이언트가 되기도 하는 유연한 구조를 가지고 있으며 GUI 공유와 이벤트 공유 메카니즘을 사용한 도구 공유를 지원한다. 서버는 SPACE의 사용자 인터페이스 처리, 공동작업 할 사용자들에 대한 네트워크 환경 구축, 공유 도구의 수행 모듈 관리와 각 클라이언트에서 발생한 사용자들의 이벤트를 수행하여 그 결과를 각 클라이언트에 보내주는 역할을 한다. 클라이언트는 SPACE의 사용자 인터페이스 처리, 공유 도구에 대한 사용자 입력을 받아 서버에게 전달하며, 서버로부터 전달된 결과를 출력하는 역할을 한다.

이러한 개념을 고려한 SPACE는 정보 통신 소프트웨어 개발을 지원하는 하부구조로써 지리적으로 흩어져 있는 개발자들이 공동으로 동시에 소프트웨어 개발 업무를 수행하고 협력하는 것을 지원해 주며, 계층적 구조 개념을 통해 공동작업에 여러 레벨의 기능 모듈을 제공하여 필요한 기능이나 성능이 최적으로 활용될 수 있도록 하였다.

#### 4. SPACE 구성

SPACE의 개념 모델 (그림 2)와 같이 내부 계층은 분산 객체 처리를 위한 CORBA를 기반으로 개념 모델과 같은 4개의 계층 즉, 소프트웨어 개발 사용자를 지원하는 공동작업 지원 계층, 공유된 도구들에서의 공동작업 수행을 지원하는 공동작업 처리 계층, 팀 커뮤니케이션 지원 계층, 멀티미디어 자료, 이벤트 메시지, 도구들의 GUI 이미지 등의 통신을 전달하는 멀티 네트워크 지원 계층으로 구성되어 있다.



(그림 2) SPACE 개념 모델  
(Fig. 2) Conceptual model of SPACE

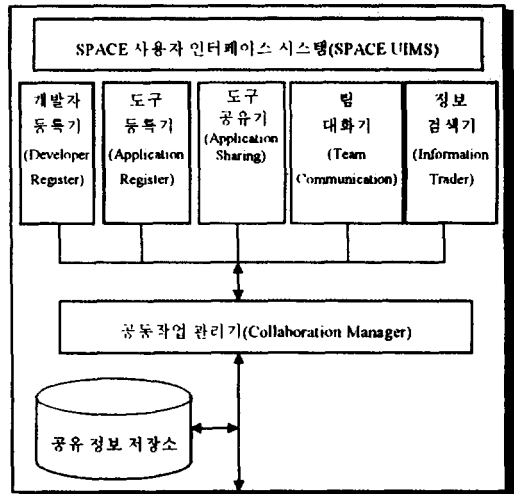
- 공동작업 지원 계층  
공동작업에 필요한 그룹, 사용자, 도구, 자료들의 정보를 관리, 적합한 사용자인지를 지원하는 계층
- 공동작업 처리 계층  
필요한 도구와 서비스를 구동하고 실질적인 공동작업의 행위인 이벤트를 처리하여 모든 공동작업자에게 그 결과를 동시에 공유하도록 지원하는 계층
- 팀 커뮤니케이션 지원 계층  
분산된 환경에서 팀간의 의사 전달(화상, 음성, 공유 워크스페이스, 메일, 화이트 보드 등)을 지

원하는 계층

- 멀티-네트워크 지원 계층  
상위 계층에서 필요한 다양한 통신 프로토콜을 지원하는 계층

#### 4.1 공동작업 지원 계층

공동작업 지원 서버 시스템은 공동작업에 필요한 그룹, 개발자, 도구, 데이터들의 정보를 관리하고 시스템에 적합한 사용자인지를 인증하는 계층으로 사용자 인터페이스, 개발자 등록기, 도구 등록기, 도구 공유기, 그룹 통신기, 정보 검색기와 공동작업 관리기로 구성되어 있으며 그 구성은 (그림 3)과 같다.



(그림 3) 공동작업 지원 계층 구성도  
(Fig. 3) Configuration of collaboration support layer

- SPACE 사용자 인터페이스 시스템  
공동작업에 관련된 모든 행위(공동작업의 동적 참여 및 탈퇴, 발언권 제어 등)를 GUI를 통하여 개발자와 시스템 사이의 의사 전달 수단을 제공한다.
- 개발자 등록기  
SPACE 시스템을 사용하려는 공동 개발자의 정보를 시스템에 등록, 기 등록자 정보의 변경, 삭제 기능을 가지고 있으며, 추가 등록자에 대한 인증 기능과 SPACE 시스템의 모든 개발자 정보를 검색하는 기능도 제공한다. 또한, 개발자들에

대한 그룹 정보를 유지하며 그룹의 등록, 삭제 그리고 변경 기능을 가진다. 각 개발자들은 다수의 그룹에 포함될 수 있다. 등록된 정보는 공유 정보 저장소에 저장한다. 개발자 정보의 데이터 구조 다음과 같다.

```
typedef UserInfo {
    char * GroupName,
    char * UserName,
    char * IPAddress,
    int NetworkProtocol,
    char * Birthday,
    char * BuseoName,
    char * TelephoneNumber,
    char * ZipCode,
    char * FaxNumber,
    int Role,
    char * Password,
    int Status }
```

정보 공유 저장소의 정보 관리를 위한 함수는 다음과 같다.

```
int Registry_Insert_Entry(int EntryType, UserInfo* userinfo)
int Registry_Update_Entry(int EntryType, UserInfo* userinfo)
int Registry_Delete_Entry(int EntryType, UserInfo* userinfo)
int Registry_Retrieval_Entry(int EntryType, UserName* username, GroupName* groupname)
```

• 도구 등록기

공동작업 시 공유할 도구에 대한 위치, 도구의 실행 모듈 패스 등의 정보를 추가, 삭제 그리고 변경할 수 있는 모듈로서 변경된 정보는 공유 정보 저장소에 저장되며 모든 사이트에서 등록된 도구의 정보를 검색할 수 있는 기능도 제공한다.

```
typedef AppInfo {
    char * GroupName,
    char * UserName,
    char * AppName,
    char * AppPath,
```

```
int AppSize,
char * AppOverview,
char * AppDate }
```

• 정보 검색기

모든 사이트에서 등록된 개발자 및 도구에 대한 정보를 검색할 수 있는 모듈로써 정보 저장소를 통하여 공동 작업자 정보, 등록 도구 정보, 팀 대화기 정보, 공동작업 정보를 가져온다.

• 도구 공유기

공동 작업을 하기 위한 입력된 정보(개발자, 공유 도구 등)와 공동 작업 개발자들의 등록 도구를 조사하여 GUI/이벤트 엔진을 선정하며, 이를 통하여 여러 명의 개발자들이 동시에 소프트웨어 개발을 가능하게 하는 환경 조성 및 공유 엔진을 구동한다.

```
typedef CoworkInfo {
    char * CoworkName,
    char * GroupName,
    char * CoworkerName,
    char * CoworkOverview,
    int MaxUserNo,
    char * CoworkerIP,
    int CoworkCharacteristic,
    int CoworkExit,
    char * CoworkApp,
    char * AppOwner,
    char * AppPath,
    char * ReserveTime,
    int Conductor,
    int Status,
    char * ConductorShip }
```

• 팀 대화기

SPACE에서 외부 시스템들을 연결하는 기능을 가지고 있으며 팀 커뮤니케이션 시스템인 공유 문서 설명기, 멀티미디어 메일, 화상 회의 단위 시스템들을 구동하게 해주는 팀 커뮤니케이션 시스템 연결기로 구성되어 있으며 실질적인 구동 시 개발자, 회의 명 등의 정보를 공동작업 관

리기에 전달하는 역할을 한다.

```
typedef TeamConfInfo {
    char * TeamConfname,
    char * GroupName,
    char * UserName,
    char * TeamConfOverview,
    char * TeamReserveTime,
    char * TeamConfConvenor }
```

• 공동작업 관리기

SPACE 사용자 인터페이스를 통하여 공유할 도구, 데이터, 공동 개발자들의 정보 등을 전달받아 공동작업에 필요한 정보를 실질적으로 관리하고 공동작업 처리기의 구동, 공동작업 수행 시 발생하는 발언권 제어, 동적인 참석과 탈퇴를 처리하는 기능과 팀 커뮤니케이션 시스템 등을 실질적으로 구동하고 관리한다.

• 공유 정보 저장소

각 계층에서 사용한 정보들을 저장하고, 해당 시스템에서 필요한 정보들을 읽어서 공동작업에 필요한 기능을 수행하도록 하는 것으로서 공동작업 정보, 개발자 정보, 팀 대화기 정보 등을 ODBC의 ACCESS DB를 사용하고 있다.

4.2 공동작업 처리 계층

필요한 도구와 서비스를 구동하고 실질적인 공동작업의 행위인 이벤트를 처리하여 모든 공동 작업자에게 그 결과를 동시에 공유하도록 지원하는 계층이다.

4.2.1 공동작업 이벤트 분류

본 시스템에서의 이벤트는 크게 SPACE 사용자 인터페이스에서 발생하는 사용자 이벤트(user event), 복제 구조하에서 발생하는 시스템 이벤트(system event)와 중앙 집중식 구조하에서의 GUI 공유 도구에서 발생하는 도구 공유 이벤트(application sharing event)로 분류된다. 이 있고, 사용자 이벤트에는 공동 작업 생성,초청,참여,탈퇴, 발언권 제어, 팀 대화기 선택 등의 이벤트가 있고, 이에 대한 해당 함수로는 회의 개회 및 종료 함수(Conference\_Create, Conference\_Invite, Conference\_Join, Conferenceconfernece\_Lock, Con-

ference\_Unlock, Conference\_Eject\_User, Confernece\_Disconnect, Conference\_Terminate), 회의 명부 함수(Conference\_Roster), 회의 구성 권한 함수(Conductor\_Assign, Conductor\_Release, Conductor\_Please, Conductor\_Give, Conductor\_Inquire) 등의 함수가 있다. 시스템 이벤트는 복제 구조를 가지는 공유 도구의 사용자 인터페이스에서 발생하는 마우스 움직임에서 발생하는 이벤트(MouseUp, MouseDown, MouseLeft, MouseRight, MouseClick, MouseButtonClick)와 키보드 입력에 대한 좌표값 및 키보드 입력 값을 가지는 UserInput 이벤트가 있다. 이에 대한 행위는 시스템 이벤트를 타 시스템에서 수행 가능한 윈도우 메시지를 생성한다. 도구 공유 이벤트는 어플리케이션에 관련된 정보를 다루는 ApplicationPDU (Protocol Data Unit), 공유 어플리케이션에서 사용하는 폰트 매칭 정보를 다루는 FontPDU, 사용자 입력 정보를 다루는 InputPDU, 텔리포닝 정보에 대한 PointerPDU, 각 시스템들간의 동기화에 대한 SynchronizedPDU, 비트맵 정보에 대한 UpdateBitmapPDU, 팔레트 정보에 대한 UpdatePallettPDU, 공유 어플리케이션 위치,크기 또는 z-order 정보에 대한 WindowListPDU, 시스템들의 자원 정보에 대한 UpdateCapabilityPDU 등의 이벤트가 있다. 이러한 이벤트들은 다음의 데이터 구조로 패킹되어 정보 교환을 하고 어플리케이션 공유를 통하여 공동작업이 가능하게 된다.

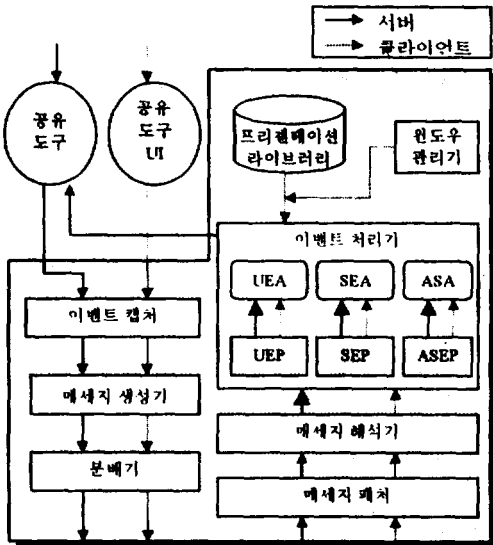
```
typedef SPACEMessage {
    int DestinationID
    int SourceID
    int Object_Identifier
    char* Object_Descriptor
    long Object_Length
    char* Object_PDU_Data }
```

4.2.2 공동작업 처리 계층 구성

공동작업 처리 계층의 구성은 도구 이벤트 캡처(capture), 분배기, 메시지 패처(Message Fetcher), 해석기, 이벤트 처리기와 윈도우 관리기로 구성되어 있으며, 그 구성은 (그림 4)와 같다. 공동작업 처리 계층은 작업 종류(도구 유무)에 따라 서버가 될 수도 있고 클라이언트가 되기도 함으로써 서버와 클라이언트의

모든 측면을 고려한 구성이다[17].

- 이벤트 캡처(event capture)  
서버와 클라이언트에서 공유 도구의 초기 화면 상태와 개발자 입력 이벤트와 입력 이벤트 처리에 대한 GUI 이미지(image)를 캡처하는 모듈
- 메시지 생성기(message Generator)  
이벤트 캡처에서 캡처한 이벤트들을 분류하여 각 이벤트에 따른 메시지를 생성하여 주는 모듈



(그림 4) 공동작업 처리 계층 구성도  
(Fig. 4) Configuration of collaboration processing layer

- 분배기(distributor)  
이벤트 캡처에서 만든 메시지를 모든 개발자들의 사이트로 분배하는 역할을 하는 모듈
- 메시지 패처(message fetcher)  
서버 및 클라이언트로부터 전달된 메시지들을 저장된 순서대로 하나씩 가져오는 역할을 하는 모듈
- 메시지 해석기(message interpreter)  
패치된 메시지를 해석하여 사용자 이벤트, 시스템 이벤트, 도구 공유 이벤트로 분류하는 모듈
- 이벤트 처리기(event processor)  
분류된 이벤트를 윈도우 관련 모듈(사용자 이벤트 액션(UEA:User Event Action), 시스템 이벤트 액션(SEA:System Event Action), 도구 이벤트

액션(ASA:Application Sharing event Action))을 이용하여 실질적으로 처리하는 모듈로서 프리젠테이션 라이브러리와 윈도우 관리를 이용한다.

- 프리젠테이션 라이브러리(presentation library)  
서버 측의 도구 수행과 동일한 결과를 디스플레이 하기 위한 윈도우 정보와 관련된 동적 라이브러리로 구성되어 있다.
- 윈도우 관리자(window manager)  
해당 사이트의 수행 중인 윈도우들의 ID(Identifier)들의 구조를 정의하고 있는 부분으로서 서버 측의 전달된 메시지에서 서버의 현재 수행 중인 해당 윈도우 ID와 클라이언트 측에서 수행 할 ID의 매핑 정보를 가지고 있다.

#### 4.2.3 공동작업 처리 알고리즘

공유 어플리케이션을 가지고 있는 서버에서의 공동작업 처리 절차는 다음과 같다.

- Release mode, 문서설명기는 구동 되지 않았으므로 초기화

- 어플리케이션 이벤트 공유를 ON로 초기화

while(엔진 종료)

```
{ while((Get_Event() == F) or (Get_Message_Event() == F))
```

```
{ Check_Event() }
```

```
switch Event
```

```
{
```

```
case System_Event :
```

```
Floor_Control ;
```

```
- Request Mode
```

```
· 자기가 floor를 가졌는지를 조사함
```

```
· 가졌다면 어플리케이션 공유 이벤트를 발생을 ON으로 Set
```

```
· 가지고 있지 않다면 어플리케이션 이벤트 공유를 OFF으로 Set
```

```
- Release Mode
```

```
· 어플리케이션 공유 이벤트를 발생을 On으로 Set
```

```
문서설명기 ;
```

```
- TRUE: 현재 수행 중인 문서설명기를 종료한다.
```

```
- FALSE: 현재 공동 작업하고 있는 어플리케이션이 있는지 조사
```

```
· 있다면 공유 어플리케이션 위에 문서설명기를
```



구동함

- 없다면 문서설명기를 구동함

case Application\_Event:

Event\_Sharing;

- 자기 사이트에서 입력한 이벤트이고, 어플리케이션 공유 이벤트가 ON 이라면 ASEP(Event) 호출
- 해당 Event를 모든 자기 사이트를 제외한 공동 작업자에게 분배함
- 해당 이벤트가 다른 사이트에서 전달된 메시지 (SPACE Queue) 라면 ASEP(Event)

GUI\_Sharing;

- ~ 자기 사이트에서 입력한 이벤트이고, 어플리케이션 공유 이벤트가 ON 이라면 ASEP(Event) 호출
- 해당 Event를 자기 사이트를 제외한 모든 공동 작업자에게 분배함
- 해당 이벤트가 다른 사이트에서 전달된 메시지 (SPACE Queue) 라면 ASEP(Event)
- 해당 Event에 대한 처리 결과를 자기 사이트를 제외한 모든 공동 작업자에게 분배함

case User\_Event:

Floor\_Control;

- request mode: 0을 1로 변환
- Floor\_Control mode가 0으로 setting 되었다면 해당 사이트의 사용자에게 floor를 할당하고 request mode로 변환함(해당 사이트에 System\_Event 발생)
- release mode: 1을 0으로 변환
- 현재 Control을 가지고 있는 사이트가 요청 검사
- True: Floor를 회수하고 release mode로 변환(해당 사이트에 System\_Event 발생)
- False: 요청한 사이트에게 메시지("사용자 000가 발언권을 가지고 있습니다")를 전달하고 해당 이벤트를 무시함

문서설명기;

- 문서설명기가 구동되었는지 조사
- True: 모든 공동 작업자들의 사이트에 구동 되어 있는 Engine Controller에 문서설명기 종료 메시지를 보냄(System\_Event 발생)
- False: 모든 공동 작업자들의 사이트에 문서설명

기 구동 메시지를 전달함(System\_Event 발생)

공동작업참여: 공동작업 참여 모듈 호출

공동작업탈퇴: 공동작업 탈퇴 모듈 호출

기타 메시지는 무시됨:

```

}
}
Get_Event()
{ - 사용자로부터 입력 이벤트를 받아들임
  - 사용자로부터 입력 이벤트가 발생했으면 Return True
  - 그렇지 않으면 Return False
}
Get_Message_Event()
{ - 각 공동작업자 사이트에서 전달되어진 메시지
  큐에서 하나의 메시지를 가져옴
- 각 메시지에 대하여 not empty라면 Return True
- 메시지 큐가 empty면 Return False }
Check_Event()
{ 이벤트를 분류(system, user, application sharing
event)하고 또한 이에 따라 세부적으로 분류함
(Controll_Floor, 문서설명기, Event_Sharing,
GUI_Sharing) }}
    
```

공동작업에 참석 중인 클라이언트에서의 공동작업 처리 절차는 다음과 같다.

- Release mode, 문서설명기는 구동 되지 않았으므로 초기화
  - 어플리케이션 이벤트 공유를 ON으로 초기화
- while(엔진 종료)

```

{
  while((Get_Event() == F) or (Get_Message_Event() == F))
  { Check_Event() }
    
```

switch Event

```

{
    
```

case System\_Event:

Floor\_Control;

어플리케이션 이벤트 공유를 서버로부터 받은 결과로 Set함

문서설명기;

- TRUE: 현재 수행 중인 문서설명기를 종료함

- FALSE: 현재 공동 작업하고 있는 어플리케이션

이 있는지 조사

- 있다면 공유 어플리케이션 위에 문서설명기를 구동함
- 없다면 문서설명기를 구동함

case Application\_Event:

Event\_Sharing;

- 자기 사이트에서 입력한 이벤트이고, 어플리케이션 공유 이벤트가 ON 이라면 ASEP(Event) 호출
- 해당 Event를 모든 자기 사이트를 제외한 공동 작업자에게 분배함
- 해당 이벤트가 다른 사이트에서 전달된 메시지 (SPACE Queue) 라면 ASEP(Event)

GUI\_Sharing;

- 자기 사이트에서 입력한 이벤트이고, 어플리케이션 공유 이벤트가 ON 이라면
- 해당 Event를 서버에게 전달함
- 해당 이벤트가 다른 사이트에서 전달된 메시지 (SPACE Queue) 라면 ASEP(Event)

case User\_Event:

Floor\_Control;

서버에게 Control-Floor이벤트를 보냄

문서설명기;

- 문서설명기가 구동되었는지 조사
- True: 모든 공동 작업자들의 사이트에 구동 되어 있는 Engine Controller에 문서설명기 종료 메시지를 보냄(System\_Event 발생)
- False: 모든 공동 작업자들의 사이트에 문서설명기 구동 메시지를 전달함(System\_Event 발생)

공동작업참여: 공동작업 참여 모듈 호출  
 공동작업탈퇴: 공동작업 탈퇴 모듈 호출

기타 메시지는 무시됨;

```
}
}
```

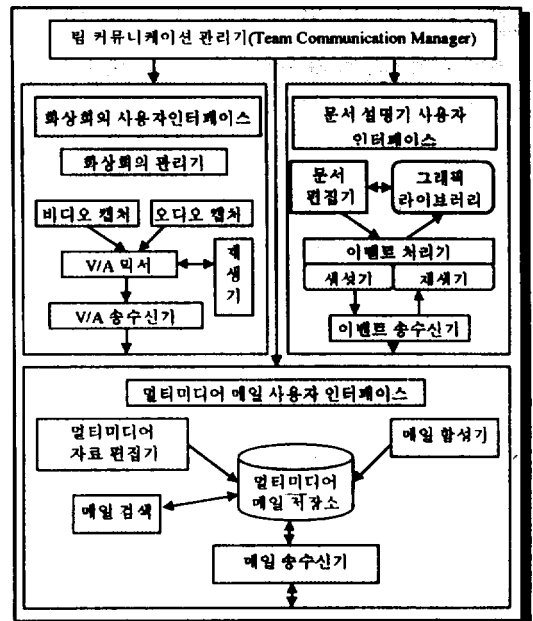
4.4 팀 커뮤니케이션 계층

분산된 환경에서 팀간의 의사 전달(화상, 음성, 멀티미디어 메일, 공유 문서 설명기 등)을 지원하는 계층으로 시스템의 구성은 (그림 5)와 같으며, 각 모듈의 기능은 다음과 같다.

팀 커뮤니케이션 시스템 중 화상회의 시스템의 처리 방법은 공동작업 처리 지원 계층의 어플리케이션 공유 이벤트 처리 방법을 적용하였고, 공동작업 문서 설명기는 시스템 이벤트 처리 방법을 적용하였다.

• 팀 커뮤니케이션 관리기

공동작업 중에 개발자들간의 커뮤니케이션에 대한 정보 (참여자, 네트워크 포트, 사용 시스템 등) 들을 관리하는 모듈



(그림 5) 팀 커뮤니케이션 구성도  
 (Fig. 5) Configuration of team communication

• 화상 회의 시스템

팀의 활동을 지원하는 대표적인 서비스로서 공동작업 참여자들의 컴퓨터 외부 환경과의 독립성 유지를 위하여 소프트웨어 영상 캡처 기법을 사용하였으며, 분산되어 있는 참여자들의 동적인 회의 참여와 회의 진행에 필요한 발언권 제어, 회의 진행 시 참가자들이 영상, 음성 매체의 동기화를 이용하여 정보를 서로 교환할 수 있는 통신 환경을 제공한다.

• 공유 문서 설명기 시스템

현재 공유된 어플리케이션을 이용하여 공동작업

수행 시 공유 어플리케이션에 영향을 주지 않고 각 참여자들에게 설명할 수 있도록 하는 서비스로서 공유 어플리케이션의 화면 캡처를 이용하며, 참여자 이벤트를 받아서 문서 설명기 시스템에서 입력할 수 있는 이벤트를 생성하고 이를 각 참여자들에게 수신한다. 수신된 이벤트는 재생기를 통하여 해석하고, 출력하는 처리 절차를 가지고 있다. 또한 채팅 기능도 제공하고 있다.

• 멀티미디어 메일 시스템

멀티미디어 메일은 텍스트, 그래픽, 이미지 그리고 음성을 포함하는 메일 문서로서의 교환 능력을 제공하는 것이며, 비동기 모드로 파일의 송수신 기능을 가진다. 멀티미디어 메일 시스템은 각 모노 미디어 데이터를 멀티미디어 자료 편집기를 통하여 생성하고, 이를 메일 합성기에 의해서 멀티미디어 메일로 합성하여 해당 참여자들에게 전송하면, 전송된 메일은 멀티미디어 메일 저장소에 저장된다. 전송된 메일은 일련의 모노 미디어 데이터의 결합으로서 메일 검색 모듈을 이용하여 멀티미디어 메일을 브라우징하고 특정 디렉토리로 저장할 수 있다.

4.5 네트워크 지원 계층

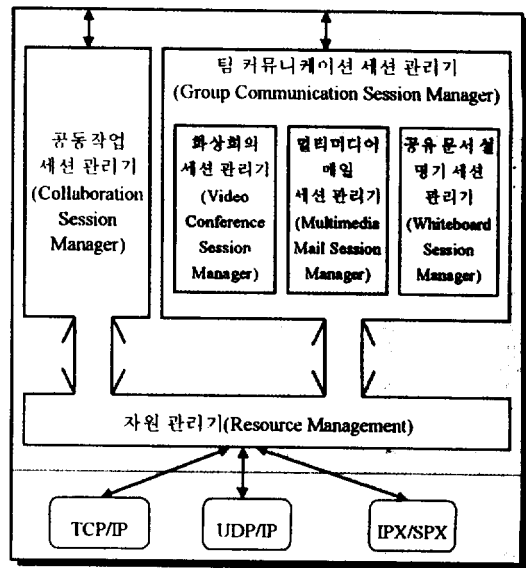
상위 계층에서 필요한 다양한 통신 프로토콜을 지원하는 계층으로 그 구성은 (그림 6)과 같으며 영역, 채널, 데이터 전송, 토큰 관리를 한다. 영역 관리에 대한 함수로는 Connect, Disconnect, Attach\_User, Detach\_User가 있으며, 채널 관리에는 Channel\_Join, Channel\_Leave, Channel\_Expel 함수, 데이터 전송에는 Send\_Data, Uniform\_Send\_Data 함수, 토큰 관리에는 Token\_Grab, Token\_Inhibit, Token\_Give, Token\_Please, Token\_Release 함수가 있다.

• 자원 관리기

공동작업 및 팀 커뮤니케이션 세션 관리기와 네트워크 다중 프로토콜과의 인터페이스를 제공한다.

• 공동작업 세션 관리기

공유 도구에 의해 공동작업을 하기 위해 필요한 개발자들의 접속과 종료 그리고 공동작업에 필요한 정보(GUI, 각종 이벤트)를 전달 및 수신할 수 있는 세션을 제어하고 관리한다.



(그림 6) 네트워크 지원 계층 구성도 (Fig. 6) Configuration of network support layer

4.6 SPACE 평가

공동작업 환경을 제공하는 관련 연구들에서의 시스템들은 구조적인 측면에서 중앙 집중식 방식이나 복제 구조 방식을 채택하고 있다. 이에 따라 각각의 시스템들은 분산 개발자들의 정보 공유를 위하여 2가지 방법 - GUI 공유 방식, 이벤트 공유 방식 -에서 하나의 방식을 사용하여 개발되었다. 하나의 구조방식과 정보 공유 방식을 지원하는데 따른 문제점들은 2장에서 기술한 것과 같다.

SPACE는 중앙 집중식 방식과 복제 구조 방식 모두를 지원하는 복합 구조로 설계되었으며, 분산 개발자들의 공동작업 지원을 위하여 GUI 공유 방식과 이

• 팀 커뮤니케이션 세션 관리기

팀 커뮤니케이션 서비스를 사용할 경우 필요한 세션을 원하는 것으로 비동기적인 서비스인 멀티미디어 메일 전송 및 수신을 위한 세션을 제어 관리하고, 실시간으로 화상회의에 필요한 개발자들의 영상과 음성 채널을 제어 관리한다. 그리고 공유 문서 설명기에 필요한 이벤트 전달 및 수신할 수 있는 세션을 관리한다.

넷 공유 방식, 2가지 모듈을 지원하는 완전 분산 제어 구조를 취하고 있다. 또한, 분산 환경에서 다수의 정보 통신 시스템 개발자가 개발 도구와 정보를 공유하여 동시에 공동작업을 지원하는 플랫폼으로써 발 언권제어, 주석, 텔리포인팅 등의 기능을 제공하고 있으며, 공동 개발자간의 소프트웨어 개발 시 원활한 의사소통을 위하여 화상회의, 비동기식 멀티미디어 정보를 주고 받을 수 있는 멀티미디어 메일 기능, 개발자간의 공유 작업 공간을 지원하는 공동작업 문서 설명 기능과 이기종간의 상호 운용성을 위하여 데이터 회의 국제 표준인 IMTC의 T.120 시리즈를 근간으로 설계하였다.

## 5. 결 론

본 논문은 소프트웨어 개발을 지원하는 CSCW 개념을 적용한 지리적으로 분산된 개발자들의 공동작업을 지원하는 플랫폼으로써 공동작업 지원, 공동작업 처리, 그룹 커뮤니케이션, 네트워크 지원 계층으로 구성된 다자간 공동작업 플랫폼인 SPACE 설계에 관한 것이다. SPACE는 Windows 95 환경 하에서 완전 분산 구조를 기반으로 하는 클라이언트/서버 모델이며 공유할 도구의 유무에 따라 서버, 클라이언트가 되기도 하는 유연성을 가지고 있다. 도구를 공유하게 하는 공유 기법으로는 GUI 공유 방식과 복제 구조를 지원하기 위하여 이벤트 공유 방식을 사용하고 있다.

이러한 개념으로 설계된 SPACE는 T.120 시리즈를 근간으로 정보 통신 소프트웨어 개발을 지원하는 하부 구조로서 지리적으로 분산된 개발자들이 공동으로 소프트웨어 개발 업무를 수행하고 협력하는 것을 지원해 주며, 계층적 구조 개념을 통해 공동 작업에 여러 레벨의 기능 모듈을 제공하여 필요한 기능이나 성능이 최적으로 활용될 수 있도록 하였다. SPACE 설계를 바탕으로 현재 프로토타입 시스템 개발 중에 있다.

앞으로의 연구 방향은 데이터 회의의 국제 표준인 IMTC 산하 T.120의 어플리케이션 공유 표준(T.Share), 다자간 회의 설정과 관리 서비스를 제공하는 T.124(GCC:Generic Conference Control)과 일반적인 다자간 접속 지향의 데이터 서비스를 지원하는 T.122/T.125(MCS:Multipoint Communication Service)를 반영한 지속적인 표준화 작업과 이에 따른 구현에 관한 것이

다. 그리고 개방 분산 환경에서 어플리케이션들 사이의 상호 운용성을 제공하고 이 기종의 다중 객체 시스템들에 대한 투명성을 제공하는 CORBA 기반 분산 객체 처리에 관한 연구도 활발히 진행 할 예정이다.

## 참 고 문 헌

- [1] A. Babadi, "COMIX:A Tool to Share X Applications", Proc. Second Workshop on Enabling Technologies:Infrastructure for Collaborative Enterprises, 1993, pp. 192-196.
- [2] A. Versey and Ajay Paul Sravana, "CASE as Collaborative Support Technologies", Communication of the ACM, Jan. 1995, pp. 83-94.
- [3] G. Toye, M. Cutkosky, L. Leifer, J. Tenenbaum, and J. Glicksman, "SHARE:A Methodology and Environment for Collaborative Product Development", Proc. Second Workshop on Enabling Technologies:Infrastructure for Collaborative Enterprises, 1993, pp. 33-47.
- [4] H. Abdel-Wahab and M. Feit, "XTV:A Framework for Sharing X windows Clients in Remote Synchronous Collaboration", Proc. IEEE Conference on Communications Software, 1991, pp. 159-167.
- [5] J. Adam and D. tenneenhouse, "The Vidboard:A Video Capture and Processing Peripheral for a Distributed Multimedia System", Proc. ACM Conference on Multimedia, 1993, pp. 13-120.
- [6] J. Lauwers, T. Joseph, K. Lantz, and A. Romanov, "Replicated Architectures for Shared Window Systems:A Critique," Proc. ACM Conference on Office Information Systems, 1990, pp. 249-260.
- [7] J. Patterson, R. Hill, S. Rohall, and M. Meeks, "Rendezvous:An Architecture for Synchronous Multi-User Applications", Proc. ACM Conference on Computer-Supported Cooperative Work, 1990, pp. 317-328.
- [8] James D. Palmer and N. Ann Fields, "Computer-Supported Cooperative Work", IEEE Computer, May 1994, pp. 15~17.

[9] K. Srinivas, R. Reddy, A. Babadi, S. Kamana, V. Kumar, and Z. Dai, "MONET: A Multi-media System for Conferencing and Application Sharing in Distributed Systems", Proc. First Workshop on Enabling Technologies for Concurrent Engineering 1, 1992, pp. 21-37.

[10] K. Watabe, S. Sakata, H. Fukoka, "Distributed Multiparty Desktop Conferencing System: MERMAID", Pro. CSCW '90, 1990, pp. 27~38.

[11] M. Altenhofen, J. Dittrich, R. Hammerschmidt, T. Kappener, "The BERKOM Multimedia Collaboration Service", Proc. ACM Conference on Multimedia, 1993, pp. 457-463.

[12] N. Guimaraes, P. Silva, J. Santos, and A. Siemaszko, "MObjectViews: A Multiuser Worksheet for a Mechanical Engineering Environment", Proc. Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1993, pp. 182-186.

[13] S. Ahuja, J. Ensor, and D. Horn, "The Rapport Multimedia Conference System", Proc. ACM Conference on Office Information System, 1988, pp. 1-8.

[14] S. Lu, K. Simth, A. Herman, D. Mattox, M. Silliman, M. Lucenti, J. Jacobs, D. Chazin, M. Lawly, and M. Case, "SWIFT: Software Workbench for Integrating and Facilitating Teams", Proc. Second Workshop on Enabling Technologies: Infrastructure for Collaborative enterprises, 1993, pp. 48-59.

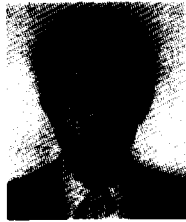
[15] T. Crowley, P. Milazzo, E. Baker, H. Forsdick, "MMConf: An Infrastructure for Building Shared Multimedia Applications", Proc. Conference on Computer-Supported Cooperative Work, F. Halase ed., ACM, 1990, pp. 329-342.

[16] T. Levergood, A. Payne, J. Gettys, G. Treese, and L. Stewart, "AudioFile: A Network Transparent System for Distributed Audio Application", Proc. Usenix Summer Conference, 1993, pp. 22-33.

[17] jboh. etc, "An Application Sharing Object and Behaviours for Collaboration Based System",

Conference on EUROMicro '97, 9, 1997.

[18] 박용진, "CSCW 연구 동향", 정보과학회지 제9권 제5호, Oct. 1991, pp. 77~81.



**조 성 빈**

1985년 울산대학교 전자계산학과 졸업(공학사)  
 1987년 동국대학교 전자계산학과 졸업(공학석사)  
 1989년~현재 한국전자통신연구소 정보공학연구실 선임연구원

연구분야: 멀티미디어, 분산 S/W, Groupware



**김 진 석**

1982년 울산대학교 전자계산학과 졸업(공학사)  
 1988년 동국대학교 전자계산학과 졸업(공학석사)  
 1982년~현재 한국전자통신연구소 정보공학연구실장(책임연구원)

연구분야: 멀티미디어 데이터베이스, 소프트웨어공학, CSCW