

## □ 기술개설 □

# CFC : CORBA를 이용한 CSCW 용 프레임워크

한국항공대학교 송동호\*

### ● 목 차 ●

- |                                 |                             |
|---------------------------------|-----------------------------|
| 1. 서 론                          | 3.2 그룹 데이터 관리               |
| 2. CORBA와 CSCW 프레임워크의 배경        | 4. CFC 프레임워크를 이용한 화상 회의 시스템 |
| 2.1 CORBA                       | 4.1 오디오, 비디오 컨퍼런스           |
| 2.2 CSCW 프레임워크                  | 5. 결 론                      |
| 3. CFC[CSCW framework on CORBA] |                             |
| 3.1 그룹 컨트롤 관리                   |                             |

## 1. 서 론

현재 수백개의 초고속 정보통신과 관련된 응용소프트웨어들이 국내 대학, 연구소, 및 기업 등에서 개발되고 있다. 대부분의 이러한 과제들은 원격교육, 원격의료, 제택근무와 같은 사용자위주의 응용프로그램 개발에 치중되어 있다. 이러한 과제들의 개발 배경은 첫째, NII에 관한 붐을 일으키는것, 둘째, 기반소프트웨어보다는 사용자 응용프로그램을 우선적으로 개발하겠다는 것으로서 기술 지배적이라기 보다는 시장 주도로 추진하자는 것이다. 이 과제들 수행에 있어서 과제 종료 시점에 초고속 선도시험망을 사용하여야 한다는것 이외 꼭 지켜져야 할 사항은 없다. 셋째, 정부는 미래 정보화 사회를 위한 도전으로서 개발자 자신의 새로운 아이디어를 실현해 보도록 격려하는데 있다. 넷째, 플랫폼이나 미들웨어에 대한 제약이 없으므로 해서 다양한 파일럿과제를 수행해 봄으로써 자연적으로 형성되는 기술 및 표준화에 기대를 하고 있는 것이다.

하지만 많은 과제들이 현재 거의 비슷한 문제에 봉착해 있다. 즉, 자신들이 응용 소프트웨어

어 개발을 시작할 플랫폼이 미비하다는 것이다. 이를 위한 대안으로서 MuX, COSMOS[1, 4,5,6,7,8]가 사용되고 있다. 하지만 이들은 다음과 같은 미흡한 점이 있다. 첫째, 개방시스템이 아니라는 것이다. COSMOS 응용프로그램은 윈도우즈, OS/2, UNIX 등 다른 프레임워크와 호환성이 없다는 것이다. 따라서 COSMOS 응용프로그램에서는 이와같은 주도적인 컴퓨터 환경에서 수행되고 응용프로그램들과 데이터 공유를 하지 못한다는 것이다. 둘째로, COSMOS 에서 멀티미디어 데이터를 처리하는 것이 간단하지 않다는 것이다. 즉, 멀티미디어 객체들간에 미세 동기를 맞추기 위하여 다양한 기능을 제공을 하는데 이들 기능들은 MCI(Media Control Interface)나 VFW(Video for Windows)와 같은 표준들을 사용하지 않는다는 것이다. 이것은 COSMOS가 VFW, AVI 등이 출시되기 이전에 설계된데 기인하기도 한다. 그래서 이러한 프레임워크는 현재 이용 가능한 O.S., 멀티미디어 처리, 분산 및 객체공유 등의 기능을 이용하여 재 설계될 필요가 있다.

프레임워크는 주어진 문제영역에서 재 사용성이 있는 연관된 클래스들의 확장 가능한 라이브러리이다. 초고속 정보통신 기반에서 필요한 모든 소프트웨어는 개념적인 프레임워크로

\*중신회원

기술될 수 있다. 응용소프트웨어 개발의 복잡성에 관한 부담을 미들웨어로 이전하기 위하여 프레임워크는 대부분의 주요기능들을 갖추고 있다. 사용자 인터페이스, 세션관리, 데이터필터링, 동적 재 연결, 동시성 제어 등을 포함한다. 이 경우 소오스노드와 목적지노드는 각각 생산자및 소비자타입을 각각 갖고 중간 노드는 릴레이 타입을 갖도록 한다.

이러한 프레임워크를 이용하여 새로운 서비스를 위한 응용소프트웨어 개발을 위한 개발 환경을 만들 수가 있다. 즉, 앞으로 응용소프트웨어의 개발에서 가장 필요한 것은 기능의 완전성과 개발기간의 단축성에 있다. 이런 요구사항을 만족시키기 위하여 프레임워크를 통한 소프트웨어 대장간의 환경조성은 필수적이다. 본 연구에서는 OurCAD(Object oriented Ultra Remote CAD) 가 개발되고 있는데 이것은 서투로직의 ASIC 주문형 반도체 설계도구인 MyCAD의 분산 컴퓨팅이라고 봐도 좋다.

이 과제는 초고속 정보통신 응용과제로서 먼저 CSCW 프레임워크를 개발하는데 초점을 두고, 이 위에서 분산 CAD를 위한 응용을 개발하는 것이다. 저자는 COSMOS의 모델링, 설계, 구현을 수행한 경험으로 이 CFC(CSCW framework on CORBA) 프레임워크를 재설계하였다. 재 모델링은 다음과 같은 새로운 요구사항을 만족시키기 위하여 수행하였다.

- 개방성 및 확장성
- 객체지향성
- CSCW 지원성
- 응용소프트웨어간 데이터 공유를 위한 통합 개발도구
- 이식성, 적응성, 순차적 확장성
- 조기 순환성(Fast-turn-around)

본 논문에서 논의할 OurCAD의 소프트웨어 구조는 그림 1과 같다.

OurCAD 분산응용소프트웨어층
CFC(CSCW 프레임워크)
CORBA
OS

그림 1 OurCAD소프트웨어 구조

모든 새로운 요구사항들은 다음 절에서 상세히 설명될 CFC 프레임워크에 포함되어 있다.

CFC는 초기의 목표인 응용소프트웨어 개발을 쉽게하기 위한것에 초점이 맞추어져 있다. 본 논문은 다음과 같은 구성을 가지고 있다. 그림 1의 CORBA에 관한 배경설명이 2절에서 설명되며 CSCW를 위한 새로운 프레임워크가 이어서 설명된다. 4절은 CFC가 OurCAD를 통하여 어떻게 응용 소프트웨어 제작에 이용되고 있는가를 설명한다. 그리고 마지막으로 5절에서 결론을 맺는다.

## 2. CORBA와 CSCW 프레임워크의 배경

### 2.1 CORBA

CORBA[11]는 고급 통신 프레임워크로서 C++언어를 이용하여 분산객체지향 응용프로그램의 개발을 쉽게 한다. CORBA는 개발자들이 이기종 컴퓨터 환경에서 견고하며, 확장성이 있는 복잡한 분산응용프로그램을 개발, 채택, 유지하도록 도와준다. CORBA와 비교될 수 있는 소프트웨어로서는 마이크로소프트사의 COM(Common Object Manager), IBM사의 SOM(System Object Manager) 등이 있다.

ORB의 요소들은 stub-skeleton, 동적 호출 인터페이스(DII), 기본객체어댑터(BOA) 등으로 구성된다. 사용자 응용프로그램들은 객체들을 위하여 IDL인터페이스를 자신의 객체들을 위하여 생성한다. 그리고 IDL 컴파일러를 통하여 스텝과 스킴리튼을 생성한다. C++ 코드는 이 스텝과 스킴리튼을 호출하여 이용한다. 객체 레퍼런스를 이용하여 클라이언트 객체에 대하여 메소드 호출을 하거나 DII를 통하여 요청을 조합할 수도 있다.

ORB Core는 객체를 찾고 필요하다면 연결을 수행하고, 요청 및 관련 데이터를 클라이언트로부터 객체구현으로 전달한다. 객체구현에는, ORB Core와 BOA는 서로 협력하여 실제 객체구현을 찾아내도록 한다. 그리고 그 요청을 스킴리튼에 의해서 메소드 호출의 형태로 전달한다. 객체구현은 클라이언트에 의해서 요청된 액션을 수행하도록 하고 ORB는 응답을

클라이언트에 되돌려 준다. 응답은 요청과 비슷한 경로를 통하여 전달된다.

OMA는 OMG표준화를 크게 3분야 : OBR, 객체 서비스, 공통기능으로 나누다, 객체서비스 구조는 기본적인 서비스 인터페이스의 카테고리를 정의한다. 공통기능들의 구조는 고급 서비스들로서 여러 응용프로그램이나 고급의 특정 영역이나 수직시장을 광범위하게 적용할 수 있는 것을 정의한다. 이것들보다 상위에는 특정한 상용제품이나 최종 사용자 시스템을 위한 응용 객체가 올라간다.

이와같은 기능들로서 지금까지 정의된 것으로서는 사용자인터페이스, 정보관리, 시스템관리, 테스트관리 등이 있다. 참고로 CORBA의 하부의 통신프로토콜로서 TCP/IP가 사용되고 있으며 10M Ethernet 및 ATM 접속이 가능하다.

## 2.2 CSCW 프레임워크

초고속 관련 대부분의 응용프로그램들은 기능이 다양해지고 코드 크기도 점점 커지고 있다. 이것은 소프트웨어 개발자가 종종 어느 정도의 운영체제가 담당해야 하는 분량을 자신의 응용프로그램에 덧붙여 왔기 때문이다. 시스템 소프트웨어 제공사들이 아무리 소프트웨어 확장 및 개선을 하더라도 대형의 상용프로그램 개발자들이 필요로 하는 특정한 시스템 소프트웨어의 요구사항을 모두 만족시키기는 어렵다. 개발자들은 반 이상의 응용코드가 자신의 응용 프로그램을 수행하는데 필요한 시스템 소프트웨어라고 이야기한다. 또한 소프트웨어 개발의 빠른 사이클이 필수적이다. 응용소프트웨어 개발자들은 응용프로그램 개발에 2~3년이 걸린다. 하지만 프로덕트 사이클들은 3~6개월이 요구되고 있다. 그래서 응용프로그램 개발자들은 수평적 혹은 수직적 공통 기능을 위한 프레임워크를 만들려고 노력한다. 하지만 CSCW를 위한 프레임워크는 아직 만들어 지지 않고 있다. 따라서 본 OurCAD(Object oriented Ultra Remote CAD) 프로젝트에서 CSCW 프레임워크를 만들기로 하였다. 본 논문은 특히 화상회의 시스템에 초점을 맞추었고, 주요 논제는 그룹컨트롤 메카니즘과 그룹데이터 및 스

트림 관리라고 할 수 있다. 그룹컨트롤 메카니즘은 그룹사용자관리 메카니즘으로 컨퍼런스 및 세션관리에 응용프로그램 접속 및 이탈이 있다. 그룹데이터와 스트림 관리는 멀티미디어 데이터 페칭, 믹싱, 인터 및 인트라스트림 동기, 공통객체를 위한 동시성 제어 등이 있다. 또한 연속미디어의 동적인 재구성성은 멀티미디어 데이터 복사 관점에서 성능을 향상시키는 핵심이다. 프레임워크의 이득은 다음과 같이 요약할 수 있다.

- 사용이 쉽고 주문형 제작이 가능 : 같은 환경의 컴퓨터내의 모든 도구들은 통합되어 있고 주어진 사용자 인터페이스의 메타포를 사용한다. 개발자들은 자신의 일에 주어진 자원을 최대한 적합하도록 수정, 적용할 수 있다.
- 개발, 소요시간의 단축 : CFC 점증 구축 시스템은 버전업되는 소프트웨어의 개발에 필요한 소요시간을 소프트웨어 재사용으로 인해서 현격하게 감소시켜 준다.

## 3. CFC[CSCW framework on CORBA]

### 3.1 그룹 컨트롤 관리

우리는 그룹 형성과 관리 메카니즘을 만들기 위하여 컨퍼런스 메타포들로부터 그룹관리를 추상화할 수 있다. 실제 컨퍼런스는 대형 컨벤션센터 내의 여러 회의실들로 구성되어 있다. 이 회의실 내부에는 마이크, 카메라, 칠판 등 회의에 필요한 장비가 갖추어져 있다. 그리고 참석자는 의장, 세션좌장, 참석자, 등록자 등이 있다. 그룹관리 서브프레임워크는 이러한 실제 회의의 사물들을 컴퓨터 도메인으로 끌어내려 은유화한 것이다.

#### 3.1.1 서비스들

그룹세션관리 서브프레임워크에 의하여 제공되는 서비스는 다음과 같다.

- 공유권한이 주어질 때 그룹사용자간에 그룹정보를 공유하는 것
- 도메인 내에서 자원들을 명명화하는 것
- 회의 참석자들에 대한 워크그룹 내에서의

공유정보 갱신

- 워크그룹에 새로운 참석자를 초청하기 위한 알림기능
- 세션 및 컨퍼런스의 등록, 관리
- 그룹참석자의 정적 및 동적인 관리
- 그룹응용프로그램의 로컬 및 리모트 관리
- 응용프로그램들간의 공유그룹객체 동시제어
- 세션모드 관리 및 발언권 제어
- 워크그룹 내에서 자원들의 액세스 능력을 향상시키기 위한 사용자 인터페이스

상기 컨퍼런스 및 세션서비스를 제공하기 위하여 그룹 서버와 그룹 클라이언트의 메카니즘을 제공한다. 그룹서버 관리는 이름서버, 발언권 제어, 세션모드 제어 등과 같은 기능을 갖는다. 사용자인터페이스성 그룹클라이언트는 상기 그룹서버에 대한 각 사용자를 위한 상대

역을 한다. 객체, 사용자, 데이터 브라우저 등이 이 그룹에 속한다. 그룹 컨트롤 관리에 관한 함수를 표 1에 나타나 있다.

3.1.2 CSCW 이름 서버

한 사용자가 유명한 컨퍼런스에 가입하고자 원한다면, 그가 어디에 어떻게 접속을 시도해야 하는가? CSCW 이름서버가 바로 이 매개 역할을 해준다.

예를들면 어떤교수가 다른 연구실들에 각각 있는 세 학생과 회의를 하고자 한다고 가정하자. 이 교수는 현재 자리에 있는 두 학생과 연결되어 회의를 시작한다. 그리고 나서 세 번째 학생이 늦게 이 회의에 들어오하고자 한다고 가정하자. 이 늦게 온 학생은 회의에 관한 정보를 웹 노운 포트인 에이전트로부터 받고 회의에 가입하고자 한다면 그가 갖고있는 소프트웨어는 구체적인 상대객체의 이름들을 알지 못한다. 이러한 경우 네임서버와 초기 접촉을 시도하고 필요한 객체들의 이름을 얻어낸다. 이제 늦게 온 학생의 서브이름서버는 교수 및 동료들의 카메라 및 윈도 등 기타 필요한 디바이스들의 객체이름을 알 수 있게 된다. 즉, 고급층과 하급층이 분리되어 있기 때문에, CSCW 이름서버는 이들을 이어준다. 이런 분리를 이어주는 역할을 이들 서버가 해주는 것이다. 따라서 모든 객체의 이름은 CSCW 이름 서버내에 있는 객체 이름 표로부터 찾을 수 있도록 한다.

표 1 그룹 제어 관련 함수들

World Information	Get_Conference_List_from_World( )	Top Working Domain
Conference Information	Create_Conference( ) Disband_Conference( ) Move_Conference( ) Get_Session_List_from_Conference( )	Conference Management
Session Information	Create_Session( ) Open_Session( ) Disband_Session( ) Move_Session( ) Get_User_List_from_Conference( ) Participant_Join( ) Participant_Leave( ) Request_Join( ) Request_Leave( ) Set_Session_Mode( ) Get_Mode_Info( ) Token_Wait( ); Token_Release( );	Basic Session managements Participants Managemnt Session Mode (Lecture, Brain Storming, Panel Discussion, Exam) Setting Floor Control
Audio, Video, Graphic Event Data Transfer	Audio_Video_Create( ) Audio_Video_Open( ) Audio_Video_Close( ) Audio_Video_Read( ) Audio_Video_Write( ) Audio_Video_Bind( ) Audio_Video_Unbind( )	

3.1.3 CORBA Bell

벨은 호출된 이를 불러내는 알림 관리자이다. 최장이 어떤 초청된 이에게 회의 가입을 요청할 때 메모를 남기게 된다. 호출된 이가 로그인하게 되면 이 사실을 알게 되고 회의에 가입할 것을 동의하면 버튼으로 의사를 표시한다. 그러면 현재 초청된자가 갖고있는 CSCW 와 관련한 객체 정보가 주이름 서버로 넘어가게 된다.

3.1.4 회의모드를 위한 동적 재구성

앞서 언급한 바와 같이 CORBA 객체는 클라이언트, 서버, 릴레이의 3가지 유형으로 나누

어진다. 클라이언트측의 CORBA응용은 사용자 인터페이스나 응용 도메인에 주로 사용되고 서버측의 객체는 대부분 시스템 자원 관리 목적으로 이용된다. CORBA는 분산 객체 관리 기이므로 클라이언트측의 객체는 위치에 무관하게 어떤 서버측 객체도 액세스할 수 있다.

발언권 제어는 CSCW의 다양한 모드를 제공하는 중요한 기능이다. 여기에는 4가지의 주요 발언권 제어모드가 있다. 브레인 스토밍, 페널토의, 시험, 강의가 주요 모드이다. 발언권 제어모드는 동적으로 변경될 수 있다. 예를 들면, 교수가 강의실에서 강의할때 강의를 끝날즈음 퀴즈를 보고자 하면 즉시 시험 모드로 변경시켜야 한다. 그래서 강의중일때 학생들은 질의 응답을 할 수가 있으나 모드가 변경된 이후에는 학생들간의 정보공유가 이루어져서는 안된다. 이러한 회의 모드를 담당하는 기능이 회의 모드 및 발언권제어 객체에 포함되어 있다.

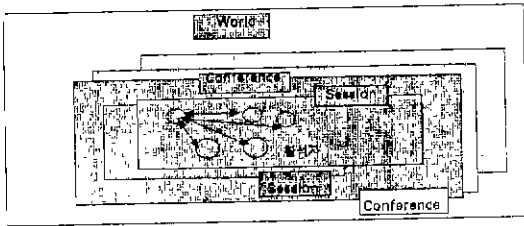


그림 2 CFC의 회의 자료 공유 도메인

### 3.1.5 이동하는 사용자들 위한 동적 네이밍

시스템 자원과 관련된 객체는 위치성을 어느 정도 갖게 되는데 예를 들면, 사용자 A의 카메라, 사용자 B의 윈도우와 같은 것이다. 사용자 A,B,C와 접속하란 명령이 떨어졌을 때 실제 A,B,C는 이동한 후이기때문에 위치정보가 변경될 수도 있다. 이름서버는 주 이름서버와 부 이름서버로 구성된다. 주 이름서버는 회의 주제자의 컴퓨터에 위치하게 되고 부 이름서버는 참가자의 호스트에 존재하게 된다. 주 이름서버는 컨퍼런스를 위한 정보를 유지하는 한편 부 이름서버는 화일이나 디바이스와 같이 자신의 정보를 유지하는데 주로 이용된다.

CFC 워킹 도메인의 모델이 그림 2에 나타나 있다. 가장 바깥쪽 월드 도메인은 이 세상의

CFC를 수행하는 네트워크로 연결이 설정된 가장 넓은 범위의 도메인이다. 호스트가 어느 정도의 데이터전송 속도로 연결되지만 한다면 지리적인 제한은 없다. 월드 도메인에서 복수의 컨퍼런스가 수행될 수 있는데 세션과 참석자는 중첩될 수도 있다.

컨퍼런스 도메인은 복수개의 세션이 들어있는 공통의 장소로서 내부 세션사이에는 객체자원을 공유할 수 있다. 대부분의 세션에 있는, 즉, 멤버쉽, 참관자 자격 등이 공유되며 높은 보안과 관련된 객체는 예외가 된다. 컨퍼런스에는 3차원 항행기법[9]을 이용한 컨퍼런스브라우저가 사용되고 있는데 이것은 대단히 많은 정보도메인내에서 복수의 시각점을 제공하고 자유로운 이동성을 제공하기 위함이다.

세션은 주어진 주제를 논의하고 정보를 교환할 수 있는 열린 공간이다. 따라서 이것은 의장과 복수의 참석자를 가지고 있다. 그래서 의장은 세션을 설정하고 관리를 할 필요가 있다. 컨퍼런스 설정과 관련한 역할로서 의장은 참석자를 선택하고 세션을 열기/닫기/등을 할 수가 있다. 그리고 오디오,비디오 세션 설정과 정지가 수행된다. 모든 컨퍼런스 관리와 관련하여 의장은 전문가를 초빙하고 참석자로 하여금 퇴장, 재초청, 발언권 제어, 회의모드 설정 등을 할 수 있도록 한다. 또한 참석자의 역할은 조인, 퇴장, 재조인, 발언권 등을 요청할 수 있다. 이와같은 역할에 관한 함수를 표 1에 상세히 나타내었다.

## 3.2 그룹 데이터 관리

### 3.2.1 서버 프레임워크 서비스

그룹 멀티미디어 데이터 관리 프레임워크에서 제공되는 서비스로서는 다음과 같은 것이 있다.

- 연속 미디어 관리를 위한 실시간 스트림 및 버퍼링
- 데이터 타입 변환 등, 압축, 복원
- 디바이스 종속 데이터 타입 변환

### 3.2.2 분산스트림을 위한 객체

전술한 바와같이 CSCW 워킹도메인 내에서는 네트워크들이 자주 사용된다. 즉 고급프로

그림에서 파일, 디바이스 위치정보에 상관없이 논리적 이름만을 가지고 자원을 액세스하는 것을 말하는데 이 경우 많은 하부 시스템 리소스는 실제 원격지에 위치하게 된다. 고급 네이밍과 저급의 자원 위치계의 차이를 배우기 위하여 분산자원 관리가 필요하다.

실례로 CFC에서의 네이밍은 <자원 이름>으로 간단하지만 기존 운영체제에서의 분산 네이밍은 <자원이름, 호스트이름>방식을 사용한다. 프로그래머는 하부 운영체제에서 지원하는 분산화일 서비스를 이용한다고 가정하더라도 네트워크상에 있는 원격화일 및 자원을 액세스하는 것은 어렵다. CORBA는 이런측면에서 RPC 보다도 더 쉽다.

클라이언트측의 파일들은 사용자 인터페이스와 응용도메인 내에서 수정되고 서버측의 파일 및 디바이스 객체들은 주로 제공자 측이 된다. CFC를 이용하면 클라이언트측의 어떤 객체에 대한 오퍼레이션 요구도 위치투명하게 액세스할 수 있다는 것이다. 클라이언트측의 API로서 open( ), close( ), read( ), write( ), create( ), ioctl( ) 등이 제공 된다. 파일이나 디바이스가 원격 클라이언트에게 서비스를 제공하고자 하면 CORBA에 등록하여 원격지에서 보일 수 있게 하는 오퍼레이션이 필요하다.

이와같은 원격화일 및 디바이스 처리가 된 분산 스트림 관리는 클라이언트 측에서 제공되는 논리적 시계에 의해 pull 모델로 이루어진다. 즉 CORBA 서버는 항상 어떤 임의의 클라이언트에서 연속 미디어 혹은 단위 미디어에 관한 액세스 요청이 있을시 균일한 데이터 흐름에 관한 책임은 지지 않는다. 그 이유는 이러한 균일성은 클라이언트 객체측의 논리적 시계에 의한 pull에 의하여 결정되기 때문이다. 단, 서버스케줄러에서 처리 용량을 넘어서는 요청을 피하는 오퍼레이션은 물론 서버에서 처리된다.

CFC에서 스트림 처리는 기존 마이크로소프트에서 제공하는 VFW, AVI포맷을 사용한다. 따라서 미디어간 동기화 관해서도 아직은 독자적인 스트림간의 인터스트림 동기는 제공하고 있지 않다.

### 3.2.3 DII와 BOA를 이용한 Buffer객체

버퍼는 대부분의 소프트웨어에서 기본 소프트웨어 모듈이다. 컴퓨터간 경계가 없는 버퍼 오퍼레이션은 분산 생산-소비자 형태의 컴퓨터 모델의 전형적인 형태이다. 즉, 네트워크로 시스템들간의 현상이 숨겨질 필요가 있다. 그 이유는 고급 프로그래밍은 분산문제에 대하여 하부 시스템 프로그램에서 담당해 주어 자신에게는 투명성을 보장해 주도록 요구하고 있다. 예를들면 카메라 이미지 캡취를 위한 소프트웨어는 원격카메라로 부터도 로컬카메라처럼 이미지를 캡취할 수 있어야 한다. 즉, 네트워크상의 모든 카메라들이 각 호스트에 있는 캡취 소프트웨어에 똑같이 보인다면 분산 자원관리가 되는 것이다. 따라서 소비자는 생산자의 위치에 무관하게 이미지를 캡취할 수 있는 것이다.

이런 응용에서 생길수 있는 문제점이 버퍼라고 할 수 있다. 생산자와 소비자간의 데이터 흐름은 적당한 버퍼링으로 균일하게 흐르지는 않을 것이다. 이를 해결하기 위하여 프로듀서는 버퍼에 데이터를 채우고 소비자는 버퍼로부터 데이터를 가져가는 것이다. 이런 버퍼문제는 응용프로그램들이 각각 갖기보다는 시스템 소프트웨어에서 하나만 갖고 응용프로그램에서 필요로 할 때 사용할 수 있도록 하는 것이 일반적이다. 따라서 버퍼는 시스템 객체로서 우선순위 측면에서도 우위를 차지할 수 있다.

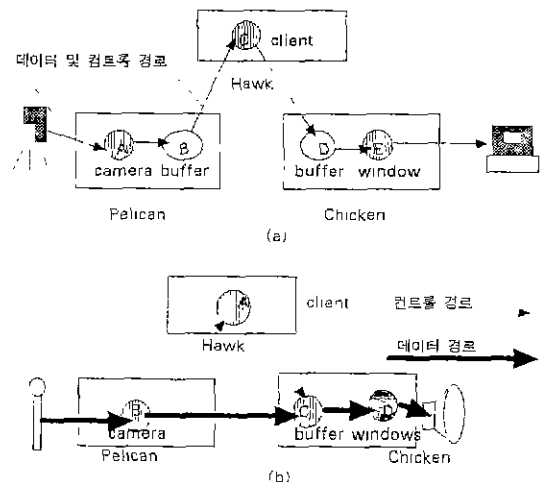


그림 3

### 3.2.4 서버 바인딩

멀티미디어 데이터 처리에서 한 주소공간에 있는 데이터를 다른 주소 공간으로 복사하는 것은 최소화할 필요가 있다. 그 이유는 멀티미디어 데이터의 크기는 시스템 전반에 영향을 미칠만큼 방대한 양이기 때문이다. 그림 3에 전형적인 데이터 복사 모델인 생산자-소비자 모델을 나타내었다. 생산자는 카메라로부터 데이터를 생성하고 이를 버퍼클라이언트 주소공간으로 보낸다. 그러면 버퍼클라이언트는 동시에 데이터 생산자가 되면서 원격지에 있는 윈도우 소비자에게 데이터를 포워드 한다. 이 모델에서 개입된 세 개의 객체는 주소공간이 모두 독립적이며 이들간의 각 프레임에 대한 데이터 복사비용은 매우 비싸다. 비슷한 현상이 클라이언트로 부터 중간 버퍼를 가진 원격 윈도우 사이에서 발생하게 되는데 이는 그림 (a)에서 보는 바와 같이 5번의 복사가 이루어진다. 복사 비용을 줄이기 위하여 그림 (b)와 같이 단순화된 모델을 제시한다. 즉, 한 클라이언트는 관련된 모든 서버들과 제어를 위한 경로를 선정하고 서버들간의 데이터는 직접 목적주소 공간으로 전송되도록 한다. 그림 (b)에서 나타낸바와 같이 점선은 제어경로를, 실선은 데이터 경로를 나타낸다. 이와같이 서버간 직접 데이터 전송을 시도하였을때 같은 응용을 3번으로 줄일수 있는 잇점이 있다. 동적 링킹과 관련된 상세한 사항은 다음절에서 설명하도록 하겠다.

이렇듯 직접 전송을 가능케 하는 객체 구현은 개념적 모델은 그림 4와 같다. Skeleton은 객체에 서버로서의 특성을, Stub는 클라이언트

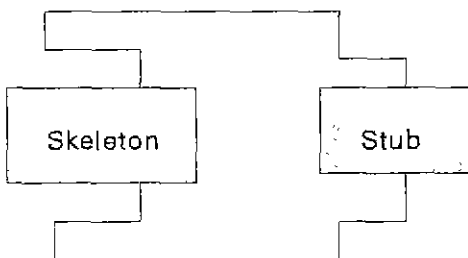


그림 4 서버간 직접 데이터 전송을 위한 릴레이 객체 모델

로서의 특성을 각각 부여한다. 이러한 모델을 가진 객체구현은 직접전송의 장점외에 런타임시에 시스템상에서 결합 가능한 다른 객체 구현들과 동적으로 결합될 수 있다는 장점을 갖는다. 그러나 앞의 객체구현들이 모든 어플리케이션에서 범용으로 수행될 수는 없다는 것을 밝혀둔다. 따라서 객체구현을 설계함에 있어 응용프로그램의 도메인을 제한하는 작업이 선행되어야 한다.

버퍼서버는 실시간 처리 시스템에 있어 데이터의 생성자측과 소비자측이 동시에 공유할 수 있는 방법론을 제공함으로써 두 객체구현 사이의 전송속도차를 해결 한다는 기본 목적과 함께, 중요한 두가지 특성을 갖는다. 첫째, 객체구현화 되어 있는 버퍼 서버는 시스템에 비의존적으로 동일한 기능을 수행함으로써 분산시스템의 요구에 적합한 형태를 갖는다. 둘째로, 하나의 객체구현 안에 서버의 특성과 클라이언트의 특성을 포함함으로써 실행시간에 데이터 서버들과 동적으로 결합한다.

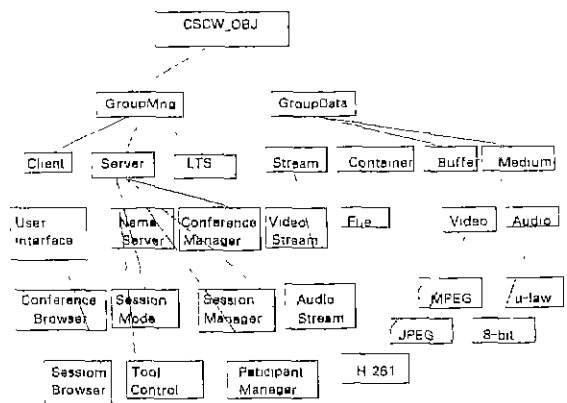


그림 5 그룹웨어용 프레임워크

컨퍼런스 제어 관리 및 데이터 관리인 두 개의 주요 기능을 위한 프레임워크를 만들기 위하여 많은 클래스들이 하나의 전체적인 체계인 프레임워크로 통합 구성되어 있다. 이 클래스 계층은 C++ 언어 및 IDL의 상속성을 이용하였다. 하지만 현재까지 IDL에서 폴리모피즘은 지원되지 않고 없어서 애로가 있었다. 이 프레임워크를 구성하는 활용하여 화상회의를 구축하였을때 장점은 다음과 같다. 첫째, 응용소프트

트웨어를 구성하는 모듈들이 유니폼한 객체구조를 갖고 있어서 소프트웨어 IC개념을 충실히 따른다. 따라서 전체적인 소프트웨어 구축이 간단 명료 하다. 둘째, 객체간의 제어제통과 데이터 전달제통이 분리되어 있어서 최적화 하였다는 점이다. 셋째, 추후 확장성에 있어서 선형적인 기능 증가가 쉽도록 되어 있다는 것이다.

이 연구를 통하여 확인된 사실들은 다음과 같다.

- 복잡한 시스템의 설계는 하나의 프레임워크를 나타내고 내부에 서브시스템을 위한 서브프레임워크를 갖는 형태가 가장 적합하다는 것이다.
- CORBA를 이용한 분산객체처리 방식에 RPC, DCE 등에 비하여 코드나 설계 재사용성을 최대로 할수 있다는 것이다.
- CORBA를 사용하는 것이 Socket( )를 직접 사용하는 것에 비하여 결코 성능의 저하를 가져오지 않는다는 것이다.

#### 4. CFC 프레임워크를 이용한 화상회의 시스템

전술한 CFC 프레임워크를 이용하여 탁상용 회의시스템을 만들 수가 있다. 본 절에서는 예로서 가장 단순한 화상회의 시스템의 서브시스템들을 설명하기로 한다.

**컨퍼런스 관리자 :** 컨퍼런스 관리자는 그룹 서버 서브프레임워크로 부터 상속 받는다. 한 컨퍼런스 관리자는 하부의 세션과 관련 객체들에 관한 정보를 관리 한다. 새로운 세션이 생성될 때 한 컨퍼런스 도메인 내에서 이루어 진다.

**컨퍼런스 인터페이스 :** 한 컨퍼런스 인터페이스는 사용자 인터페이스로 부터 상속 받아서 컨퍼런스 브라우저와 세션 브라우저를 생성한다. 한 사용자는 그룹사용자와 세션에 관한 정보를 액세스하기 위하여 컨퍼런스 인터페이스를 연다. 사용자는 컨퍼런스 인터페이스를 통하여 세션을 시작하고 가입을 할 수도 있다. 컨퍼런스 인터페이스는 3차원 항행 기법을 사용하고 있다. 여타 관리자에 관한 설명은 설명은 전술한 것으로 대신하기로 하자.[9, 10]

#### 4.1 오디오, 비디오 컨퍼런스

사용자가 비디오 컨퍼런스를 시작하면 우선 3D 사용자 인터페이스를 통하여 좌장이 참석자를 선택한다. 초청자가 선택되면 회의 모드를 결정하고 이 모드에 따라 참석자에 관한 정보가 이름 서버에 저장이 된다. 그후 who서버를 통하여 초청자가 로그인 되어 있는 컴퓨터를 원격 who객체 구현들을 통하여 검색하여 찾아낸다. 이 정보는 주 이름 서버에 등록이 된다. 원하는 초청자가 확인되면 벨 객체를 통하여 초청 메시지를 보낸다. 이 메시지도 원격 벨 객체들에게 필요한 오디오 정보만 제공하고 마치 마스터 벨은 자신의 호스트 내에 있는 벨을 구동하듯 이루어진다. 그후 초청자가 회의에 응하면 초청자가 로그인된 호스트의 CSCW관련 객체 정보가 주 이름 서버를 통하여 주 이름 서버로 모이게 된다. 이렇게 결집된 관련 객체 구현의 정보를 이용하여 클라이언트는 초청자에 대한 객체 구현들에 바인딩을 하게 된다. 이때 필요한 릴레이 객체가 생성객체와 소비객체 사이에 접속되어 원활한 데이터 흐름을 위한 완충 작용을 하게 된다.

**비디오 컨퍼런스는 2개의 객체 :** 비디오 스트림을 압축하는 객체 및 전송되어져 온 비디오 스트림을 복원하는 객체로 구성되어 있다. 각 객체는 독립적이며 독자적으로 오퍼레이션된다.

**압축객체 :** 카메라에서 입력된 화상 데이터 스트림을 압축해서 버퍼에 채워준다. 즉, 카메라 와 압축 기능은 하드웨어적으로 통합이 되어 있는데 이것은 객체 구현에 해당된다.

**복원객체 :** 다른 호스트로부터 전송 복원 되어진 화상 데이터 스트림을 복원하여 윈도우를 통해 화면에 출력하는 기능을 하는 클라이언트 객체이다. 이 객체는 LTS의 도움으로 원격지의 압축객체로부터 데이터 프레임을 pulling하게 된다.

압축과 복원은 H.261 표준에 의해 압축과 복원을 한다. 압축과 복원을 하는데 필요한 버퍼링은 압축된 데이터를 위한 버퍼와 네트워크를 통해 다른 호스트로부터 전송되어온 데이터를



임시로 저장한 버퍼가 있다. 압축을 위한 버퍼는 512k사이즈 48개를 가지고 카메라로부터 전송되어온 화상 데이터 스트림을 압축을 해서 버퍼에 채운다. CORBA를 이용한 전송 모듈은 복원버퍼를 채우기 위하여 원격 압축버퍼에서 데이터 스트림을 꺼내서 가져온다. 다른 호스트로 전송한다. 다른 호스트로부터 전송받은 화상 데이터 스트림을 저장할 복원버퍼는 512k 48개로 구성되어 있다. 복원객체에서는 복원버퍼에 채워진 데이터 스트림을 복원하여 디스플레이로 출력한다.

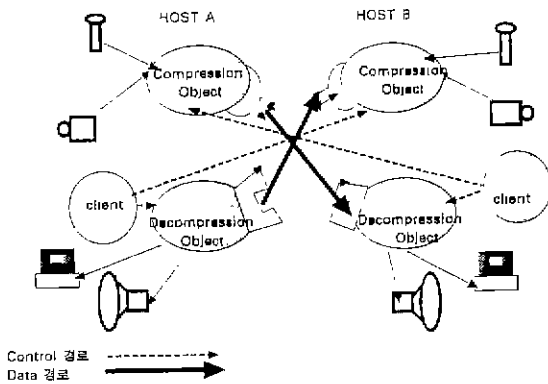


그림 6 비디오 컨퍼런스 객체 구성도

## 5. 결 론

초고속 정보 통신망을 위한 멀티미디어 그룹웨어응용들을 위한 서비스를 제공하기 위하여 CORBA 를 이용한 CFC 그룹웨어 프레임워크를 제시 했다. 이 프레임워크는 그룹 제어와 그룹데이터 관리를 위한 서브 프레임워크로 나뉜다.

이러한 메카니즘은 그룹사용자와 응용들을 위한 동적 재구성과 적응성이 있다. 그리고 객체지향적인 특성으로 인해 구현과 관련한 상세한 것들은 모두 응용프로그램 개발자들로부터 숨겨져 있음을 알 수 있다. 그래서 응용프로그램 개발자는 시스템소프트웨어 및 미들웨어 구현에 크게 부담을 갖지않고 응용프로그램 그 자체에 더 많은 개발 노력을 보낼 수 있다. 이러한 개발 환경이 바로 서비스 개발환경 구축인데 본 과제에서 추구하는 목적이기도 하다.

본 논문에서 논의된 대부분의 프레임워크는 이미 WINDOWS NT, WINDOWS 95 에서 구현되고 실행되고 있다.

현재 개발된 프레임워크는 더욱 확장하여 VOD 와 같은 대규모 상용 서비스를 위한 초고속 응용소프트웨어 제작을 위한 미들웨어로 제시하고자 하며 향후 OMG의 Common Facility한 기능으로 기고하고자 한다.

## 참고문헌

- [1] Dongho Song, et.al., "COSMOS : An Extended Operating System for Multimedia Group Presentation," Collected Abstractions for 4th International Workshop on Network and Operating System Support for Digital Audio and Video Systems," pp. 75-80, ACM and IEEE, Univ. of Lancaster, U.K. Nov. 2-5, '93
- [2] Brinck T., "Building Shared Graphical Editors Using the Abstraction-Link-View Architecture," Proceedings of ECSCW'93, PP.311-324, Milan, Italy, May 1993.
- [3] Dongho Song, "A Framework for CSCW using CORBA," Proceedings on 3rd International Conference "IMAGE COM96", Bordeaux, France, May 20-22, 1996
- [4] Dongho Song, et.al., "COSMOS, A Framework for Extended Operating Systems for Multimedia Group Presentation," Proceedings on 1995 Pacific Workshop on Distributed Multimedia Systems, pp125~129, University of Hawaii at Manoa, 1995.3.31.
- [5] Dongho Song, et.al., "COSMOS : An Extended Operating System for Multimedia Group Presentation," Proceedings for 4th International Workshop on Network and Operating System Support for Digital Audio and Video Systems, pp.87~92, ACM and IEEE sponsored, Univ. of Lancaster, U.K., Nov. 2~5, '93.
- [6] Dongho Song, et.al., "An Abstract Data Model for Multimedia Collaborative Works," Proceedings for 1994 Pacific

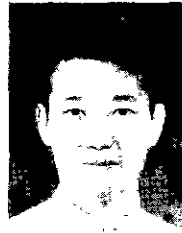
Workshop on Distributed Multimedia Systems, pp.97~108, Taipei, Feb. 1994.

- [7] Dongho Song, et.al., "COSMOS : An Object-Oriented Framework for Multimedia Group Presentation," Proceedings on ICOIN-9, The 9-th International Conference on Information Networking, Dec. 1994.
- [8] Dongho Song, et.al., "COSMOS : Framework for Real-time Multimedia Group Presentation," SPIE 95, High-Speed Networking and Multimedia Computing 1995, Bellingham, WA, U.S.A., Feb. 6-8, 1995.
- [9] 송동호, 이남경, "3D 공간상의 항행기법을 이용한 그룹웨어 GUI," HCI'96 학술대회 발표논문집, pp.244-250, 한국정보과학회 인간과 컴퓨터 상호작용 연구회 회보 5권 3호 96. 2.8-9
- [10] 인연진, 한정규, 송동호, "CORBA를 이용한 멀티미디어 데이터처리 및 그룹웨어용 소프트웨어 레고의 구현," pp.1391-1394 '95 추계

학술발표회 한국정보과학회 제22권 2호

- [11] IONA Technologies Ltd. "Orbix 2 Programming guide", Release 2.0 Nov. 1995.

**송 동 호**



- 1984 경북대학교 공과대학 전자공학과(학사)
- 1986 한국과학기술원 전기및전자공학과(석사)
- 1991 영국 Univ. of Newcastle-upon-Tyne 전산학과(박사)
- 1993 미국 Stanford Research Institute 객원연구원
- 1986~1995 한국전자통신연구소 인공지능 연구실장
- 1991~1994 멀티미디어 워크스테이션 Combistation 미들웨어 과제책임자
- 1994~1995 초고속정보통신 이용기술 GIaNT 기본소프트웨어 과제책임자
- 1995~현재 한국항공대학교 컴퓨터공학과 조교수
- 관심분야 : 운영체제, 분산시스템, CORBA, 초고속정보통신을위한미들웨어

**● APSEC '96 논문모집 ●**

- 논문마감 : 1996년 5월 31일
- 일 시 : 1996년 12월 4~7일
- 장 소 : 교육문화회관
- 주 최 : 소프트웨어공학연구회
- 문 의 처 : 포항공과대학교 강교철 교수  
T. 0562-279-2258  
F. 0562-279-2299  
E-mail: kck@wision.postech.ac.kr