

CORBA 기반 멀티미디어 응용을 위한 공동작업 서비스 설계

김 남 용[†] · 이 승 근^{††} · 왕 창 종^{†††}

요 약

초고속 통신망의 등장과 컴퓨팅 능력의 향상으로 분산 멀티미디어 응용에 대한 요구가 증대되고 있으나, 분산 응용의 개발과 확장, 유지 보수의 어려움으로 인해서 응용 개발 시 많은 시간과 노력이 필요하다. 이를 해결하기 위해서 분산 응용을 위한 개발 표준인 CORBA가 널리 이용되고 있다. 그러나, CORBA 환경에서 보다 쉽게 분산 멀티미디어 응용을 개발하기 위해서는 멀티미디어 응용을 위한 서비스들이 요구된다. 특히, 다중 사용자를 대상으로 하기 때문에, 사용자들의 정보와 그룹 정보를 관리하고, 사용자간의 상호 작용을 지원하기 위한 공동 작업 관리 기능이 필수적이다.

본 연구에서는 CORBA환경을 기반으로 한 분산 멀티미디어 응용 개발에서 필요한 공동 작업 관리 기능 제공을 위한 공동작업 서비스를 설계한다. 공동작업 서비스는 여러 사용자들로 구성된 그룹 관리와 사용자들간의 공동 작업을 원활하게 하기 위한 다양한 기능들을 제공한다. 응용 개발자는 응용 개발 시 공동 작업 기능을 구현하지 않고, 본 연구에서 설계한 서비스의 기능을 제공받음으로써 응용 개발을 보다 용이하게 할 수 있다

Design of Collaboration Service for CORBA Based Multimedia Applications

Nam-Yong Kim[†] · Seung-Geun Lee^{††} · Chang-Jong Wang^{†††}

ABSTRACT

Information superhighway and improvement of computing power result in the increase of the requirements for distributed multimedia applications. But, developer of applications needs much time and efforts because of the difficulties of development, enhancement and maintenance of distributed applications. To solve this problem, CORBA is used widely. But, the definitions of CORBA services are required for the brief developments of distributed multimedia applications. Especially, collaboration service is required because multiple users use distributed multimedia service. Collaboration service manages user and group information of applications and makes the interoperability between of multi users.

In this paper, we designed collaboration service for CORBA based distributed multimedia applications. Collaboration service is able to maintain group information and to present the ability for collaborative works between of multi users. Application developers can develop the applications easily because the ability of collaborative works is offered by collaboration service.

※ 이 논문은 1997년도 정보통신연구관리단의 대학기초연구지원사업 연구비에 의하여 연구되었음.

† 정 회 원 : 신홍대학 전산정보처리과 교수

†† 정 회 원 : 인하대학교 대학원 전자계산공학과

††† 정 회 원 : 인하대학교 전자계산공학과 교수

논문접수 : 1998년 1월 24일, 심사완료 : 1998년 11월 20일

1. 서 론

네트워크의 발달과 컴퓨팅 환경의 개선으로 원격 화상회의, 컴퓨터 지원 공동 작업(CSCW: Computer Supported Cooperative Working), 원격 교육 시스템 등의 분산 멀티미디어 응용에 대한 요구가 증대되고 있다. 그러나, 분산 환경에서의 멀티미디어 응용은 기존의 독립형 시스템(stand-alone) 개발과는 달리 개발 방법, 확장 및 유지 보수 작업이 매우 복잡하기 때문에 개발자에게 많은 시간과 노력을 요구한다. 따라서, 분산 응용 개발을 보다 용이하게 하는 개발 방식이 요구된다[1].

OMG(Object Management Group)에서 제안한 CORBA(Common Object Request Broker Architecture)는 분산 환경에서의 객체 기술을 이용한 응용 개발 표준으로, 이기종 환경에서의 손쉬운 응용 개발 방법을 제공한다[1]. CORBA 환경은 사용자에게 시스템의 분산과 이질성에 대한 투명성을 보장하며, 응용 개발에 필요한 여러 기능들을 서비스로 정의하고 있다[1]. 특히, OMG에서 1998년 6월에 표준으로 정의한 CORBA Telecom은 멀티미디어 데이터의 상호 교환 방식을 제안하고 있으며, 이는 CORBA 환경에서 분산 멀티미디어 응용 개발을 보다 용이하게 한다[2, 3].

그러나, 다수의 사용자를 대상으로 하는 분산 멀티미디어 응용의 보다 손쉬운 개발을 위해서는 그룹 통신 기능을 포함한 공동 작업 관리 기능을 필수적으로 요구한다[4, 5, 6, 7]. 공동 작업 관리 기능은 응용을 이용하는 사용자들의 정보와 그룹 정보를 관리하고, 사용자간의 원활한 상호 작용을 가능하게 한다. 특히, 대부분의 분산 멀티미디어 응용은 비슷한 형태의 공동 작업 관리 기능을 요구한다[4, 6, 7, 8]. 따라서 공동 작업 기능을 하나의 서비스 형태로 개발된다면 다수의 응용들이 보다 쉽게 이용할 수 있을 것이다.

이러한 공동 작업 관리 기능을 서비스 형태로 제공하는 플랫폼에 대한 연구는 Bellcore사의 Touring Machine과 Eurocom의 BELTEUS가 있다[9, 10]. 그러나 이러한 연구는 표준 환경이 아닌 특정 환경에서 설계되었기 때문에 범용성 문제가 있으며 기본적인 그룹 통신 기능만을 제공함으로써, 실제 분산 멀티미디어 응용 개발에 적용하기에는 미흡한 부분이 있다.

따라서 본 연구에서는 분산 객체 환경의 국제 표준인 CORBA를 기반의 분산 멀티미디어 응용을 위한 공동작업 서비스를 설계한다. 설계되는 서비스는 분산

멀티미디어 응용 개발을 위한 공동 작업 기능을 제공한다. 공동작업 서비스는 여러 개의 CORBA 객체들로 구성되며, CORBA 인터페이스를 통해서 사용자와의 상호 작용과 객체간의 통신을 처리한다. 공동작업 서비스의 특징은 여러 사용자들로 구성된 그룹 관리와 사용자들간의 공동 작업을 지원하기 위한 다양한 기능들을 제공하며, 전체 사용자와 그룹에 대한 관리를 보다 원활하게 하기 위한 집중 제어 방식으로 설계한다. 공동작업 서비스는 응용 개발자에게 응용 개발 시 공동 작업 관리 기능을 직접 구현하지 않고, 서비스로부터 기능들을 이용함으로써 응용 개발을 보다 용이하게 한다. 본 연구는 CORBA 환경에서 개발자들이 보다 손쉽게 분산 멀티미디어 응용을 개발 할 수 있게 하는 데 의미가 있다.

2. 관련 연구

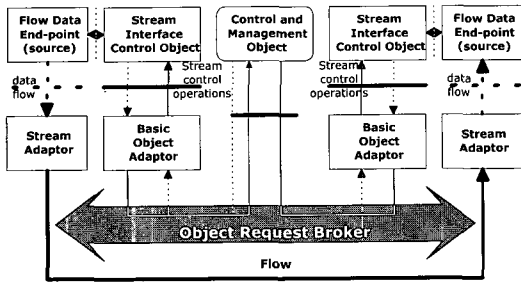
본 장에서는 CORBA에서 제공하는 멀티미디어 개발 환경과 기존의 분산 멀티미디어에서의 공동 작업 관리 기능에 대해서 고찰한다.

2.1 CORBA에서의 멀티미디어 개발 환경

OMG에서 제안한 CORBA는 분산 객체 기술을 이용한 응용 개발 표준으로써, 현재 버전 2.3이 나와있는 상태이다[1]. 크게 객체 통신을 담당하는 ORB(Object Request Broker)와 응용 개발에 필요한 여러 기능들을 서비스 형태로 정의한 COSS(Common Object Service Specification)로 구성된다. 분산 응용 개발에 있어서도 응용 개발자는 COSS를 사용함으로써 시스템의 상호 작용성을 높이고 시스템 구현을 단순화시킬 수 있는 장점을 얻는다.

특히, 1998년 7월 OMG에서 새롭게 정의한 CORBA Telecom은 CORBA 환경에서의 분산 멀티미디어 응용들을 개발하기 위한 환경을 제공한다. (그림 1)은 OMG에서 제안한 CORBA Telecom의 기본 구조이다[2].

CORBA Telecom은 연속된 프레임들을 정의한 흐름(Flow) 객체와 객체간의 흐름의 집합인 스트림(Stream) 객체를 정의하고 있다. 오디오/비디오 데이터의 제어는 기존의 ORB를 통해서 스트림 인터페이스 조정 객체에 의해서 이루어지며, 실제 오디오/비디오의 흐름은 스트림 어댑터를 통해서 ORB 밖에서 이루어지는 형태로 정의되어 있다.



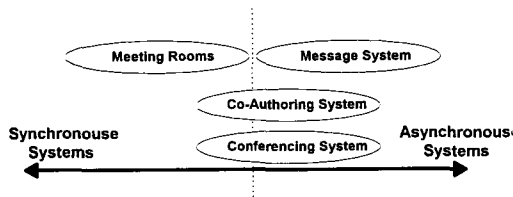
(그림 1) CORBA Telecom 구조
(Fig. 1) CORBA Telecom Architecture

CORBA Telecom의 정의는 멀티미디어 개발에 부적합한 CORBA 환경을 확장함으로써 기본 CORBA가 제공하는 분산 응용 개발의 잇점을 분산 멀티미디어 응용에서 이용할 수 있게 되었다. 따라서, CORBA 환경에서의 분산 멀티미디어 응용 개발이 보다 활발하게 이루어 질 것으로 예상된다[3].

2.2 공동 작업 형태의 분류

2.2.1 협력 형태에 따른 분류

분산 멀티미디어 응용에서 사용자들 사이에서 이루어지는 공동작업의 협력 형태에 따라 동기형 협력(synchronous cooperation)과 비동기형 협력(asynchronous cooperation)으로 나눌 수 있다[Rod92]. 동기형 협력은 사용자들이 공동작업에 모두 참여를 통한 즉각적인 상호작용을 요구하는데 비해 비동기형 협력은 모든 사용자들이 즉각적인 상호작용을 요구하지 않고, 오랜 시간에 걸쳐 이루어지는 협력 방식을 사용한다. (그림 2)는 일반적인 분산 멀티미디어 응용들을 협력 형태에 따라서 분류한 것이다.



(그림 2) 분산 멀티미디어 응용의 협력 형태
(Fig. 2) Cooperation Types of Distributed Multimedia Application

완전 동기화 시스템은 모든 사용자들이 공동작업시

동시에 존재할 것을 요구한다. 이러한 시스템들은 일반적으로 채팅(chatting)과 같은 미팅 룸(meeting room) 응용에서 사용된다. 완전 비동기 시스템은 모든 그룹 구성원들이 동시에 존재하지 않고 공동 작업을 가능하게 하는 시스템이다. 메시지 시스템(message system)은 완전 비동기 시스템이 대표적인 예로써, 사용자들은 메시지의 전송과 수신 독립적인 역할들을 수행한다. 혼합 시스템(mixed system)들은 동기적 협력과 비동기적 협력을 모두 지원하는 요소를 포함한다. 이러한 시스템들은 실시간 동기적 협력과 시간에 독립적인 비동기적 작업을 수행할 수 있다. 컴퓨터 화상회의 시스템과 공동 저작 시스템 등이 이에 속한다. 컴퓨터 화상회의 시스템은 동기적 측면이 강하며, 공동 저작 시스템은 비동기적 측면이 강하나, 다른 측면의 협력 작업에 대한 고려가 필요하다. 본 연구에서는 실제로 사용자들간의 상호작용이 빈번하게 발생하며 이를 위한 처리가 복잡한 동기형 협력을 지원하는 서비스의 설계를 목적으로 한다. 즉, 설계되는 서비스는 사용자의 그룹 참여와 탈퇴가 자유롭게 일어날 수 있으며, 이렇게 동적으로 변동되는 그룹 관리를 가능하게 한다.

2.2.2 메시지 전송 유형에 따른 분류

일반적으로 분산 멀티미디어 응용에서 사용되는 메시지 전송 유형은 특정 사용자에게 메시지 전달의 일대일 통신을 실현하는 유니캐스트(unicast), 공동작업에 참여한 모든 사용자에게 메시지를 전송하는 브로드캐스트(broadcast), 특정 목적지 및 그룹의 통신을 가능케 하는 멀티캐스트(multicast)로 구분할 수 있다[5, 11].

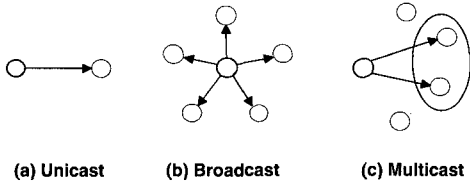
유니캐스트 전송은 한 사용자에서 다른 사용자로의 일대일 전송을 하는 방식으로 다중 전송이 필요한 환경에서는 전송하고자 하는 메시지를 일대일 전송방식으로 N번을 전송해야 하기 때문에 매우 큰 전송 지연 시간을 요구한다.

브로드캐스트 전송 구조에서는 공동작업에 참여하는 모든 사용자가 특정 메시지를 수신하고 처리하기 때문에 불필요한 지연 시간을 요구한다.

멀티캐스트 전송은 특정 사용자에게 메시지를 전송하는 방식이다. 멀티캐스트 전송 방식은 유니캐스트, 브로드캐스트 전송을 모두 지원한다. 즉, 유니캐스트는 하나의 목적지로 이루어진 그룹으로, 브로드캐스트는 전체 목적지를 하나의 그룹으로 간주한 전송과 같다.

본 연구에서 서비스의 메시지 전송 유형은 멀티캐

스트 방식을 이용한다. 따라서, 사용자간의 일대일 형태의 메시지 전송과 특정 그룹에게로의 메시지 전송이 가능하도록 한다.



(그림 3) 메시지 전송 유형
(Fig. 3) Type of Message sending

3. 공동 작업 서비스의 설계

본 장에서는 CORBA 환경의 분산 멀티미디어 응용 개발에 필요한 공동 작업 관리 기능을 제공할 수 있는 공동 작업 서비스의 기능을 정의하고, 해당 기능을 수행할 수 있는 객체들을 설계한다. 응용 개발자는 서비스의 인터페이스를 이용해서 서비스의 기능을 이용한다.

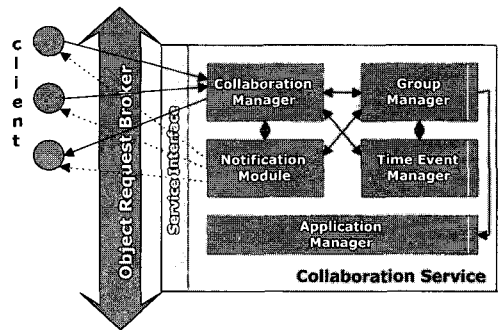
3.1 공동 작업 서비스의 구성

본 연구에서는 기존의 분산 멀티미디어 응용에서 요구되는 공동 작업 관리 기능들을 바탕으로 시간 종속적인 그룹 활동 지원이 가능한 공동작업 서비스를 설계한다. <표 1>은 공동 작업 서비스에서 제공하는 기능들이다.

<표 1> 공동작업 서비스 기능
<Table 1> Functions of Collaboration Service

연산의 종류	연산의 기능
Basic Operations	Create Group
	Destroy Group
Group Management Operations	Join Group
	Leave Group
	Manage Applications
	Manage Time Event
	Control Token
	Manage Group Context
Data Transmission Operation	Support Unicast/Multicast
Information Management Operation	Manage and Retrieve User/Group Information

공동작업 서비스는 공동작업 관리자(Collaboration Manager), 그룹 관리자(Group Manager), 응용 관리자(Application Manager), 통지 모듈(Notification Module), 시간 이벤트 관리자(Timing Event Manager) 등의 CORBA 객체들로 구성된다. (그림 4)는 본 연구에서 설계한 CORBA 기반 공동작업 서비스이다.



(그림 4) CORBA 기반 공동작업 서비스
(Fig. 4) CORBA based Collaboration Service

공동작업 관리자는 전체 서비스와 그룹 관리자를 관리하며, 사용자의 그룹 생성 요청에 대해서 새로운 그룹 관리자와 응용 관리자를 생성한다. 그룹 관리자는 그룹 정보와 그룹 내에서의 사용자간의 공동 작업을 지원하며, 응용 관리자를 관리한다. 응용 관리자는 각 그룹에서 사용되고 있는 응용들에 대한 정보를 관리한다. 통지 모듈은 공동작업 관리자와 그룹 관리자에서 발생하는 이벤트를 해당 사용자에게 전달하는 역할을 수행한다. 시간 이벤트 관리자는 시간에 종속적인 그룹 관리를 가능하게 한다. 이와 함께, 사용자 정보와 공동작업 정보, 그룹 정보, 응용 정보의 관리를 위한 데이터베이스를 포함한다.

3.2 관리자 설계

3.2.1 공동작업 관리자

공동작업 관리자는 사용자를 인증할 수 있는 기능과 인증된 사용자의 새로운 그룹 생성 요구에 대하여 그룹 관리자와 응용 관리자를 생성한다. 또한, 공동작업 관리자는 서비스에서의 사용자 정보와 공동작업에 관계되는 정보를 관리하며, 사용자의 정보 요구에 대해서 해당 정보를 사용자에게 제공한다. 공동작업 관리자를 통해 서비스를 이용하는 사용자는 전체 서비스에 대한 운영을 담당하는 서비스 운영자와 일반 서비

스 이용자들이 구분된다. 공동작업 서비스를 이용할 수 있는 사용자는 서비스 운영자에 의해서 사전에 등록되며, 연산의 인자(parameter) 값으로 전달되는 사용자 ID(ClientID)를 통해서 정당한 사용자인지를 결정한다.

서비스 운영자는 공동작업 서비스를 이용할 수 있는 사용자들에 대한 정보 변경과 활동중인 그룹에 대한 정보의 변경이 가능하며, 그룹의 등록과 해제 등의 작업을 수행할 수 있다. (그림 5)는 서비스 운영자에게 제공되는 인터페이스이다. 서비스 사용자는 서비스를 이용할 수 있는 그룹과 사용자의 등록과 해제, 등록 정보 변경 등의 중요한 임무를 수행할 수 있기 때문에, 별도의 암호를 이용한 인증과정을 거친다.

```
boolean RegisterUser( in string passwd, in string ClientID, in string LocationInfo );
boolean UnregisterUser( in string passwd, in string ClientID );
boolean ChangeUserInfo( in string passwd, in string ClientID, in string LocationInfo );
UserInformation QueryUserInfo( in string passwd, in string ClientID );
UserInformationlist QueryAllUserInfo( in string passwd );
boolean RegisterGroup( in string passwd, in string GroupID, in string topic, in unsigned short MAX, in TimeT term );
boolean UnregisterGroup( in string passwd, in string GroupID );
boolean ChangeGroupInfo( in string passwd, in string GroupID, in string topic, in unsigned short MAX, in TimeT term );
GroupInformation QueryGroupInfo( in string passwd, in string GroupID );
GroupInformationlist QueryAllGroupInfo( in string passwd );
```

(그림 5) 서비스 운영자를 위한 인터페이스
(Fig. 5) Interface for Service Administrator

사용자 정보(UserInformation)는 사용자 ID와 위치 정보, 그리고 사용자가 참여하고 있는 그룹 ID의 리스트로 구성된다. 그룹 정보(GroupInformation)는 그룹 ID와 그룹 주제, 지속 시간(duration) 정보, 그룹에 참여하고 있는 사용자들의 ID의 리스트로 구성되는데, 이중에서 지속 시간 정보는 그룹 생성자가 생성시 부여하는 값으로, 지정된 시간 이후 그룹의 자동 소멸을 위해 사용된다. 또한, 공동작업 관리자는 서비스에 등록된 사용자가 서비스를 이용할 수 있는 인터페이스를 제공한다. 인터페이스를 통해서 제공 받을 수 있는 기능은 그룹의 생성, 소멸, 참여, 탈퇴 등의 기본적인 기능과 사용자와 그룹 정보에 대한 변경 및 질의, 그리고 다른 사용자에게 유니캐스트와 멀티캐스트 형태의 메시지 전송 기능들이다. (그림 6)은 공동작업 관리자의 인터페이스이다.

```
typedef sequence<GroupInformation> GroupInformationlist;
typedef unsigned long TimeT;
enum SendType {unicast, multicast}
interface CollaborationManager {
    unsigned short GroupCreate(in string ClientID, in string passwd, in unsigned short max, in string topic, in TimeT duration);
    boolean GroupDestroy(in string ClientID, in string passwd, in short GroupID);
    boolean GroupJoin(in string ClientID, in string GroupID);
    boolean GroupLeave(in string ClientID, in string GroupID);
    GroupInformation GroupSearch(in string GroupID);
    GroupInformationlist GetAllGroupInfo(in string ClientID);
    UserInformationlist GetAllUserInfo(in string ClientID);
    UserInformation GetUserInfo(in string ClientID);
    string GetGroupID(in string ClientID);
    Boolean SendMessage(in string ClientID, in any Message, in SendType type, in string TargerID);
};
```

(그림 6) 공동작업 관리자 인터페이스
(Fig. 6) Collaboration Manager's Interface

사용자의 요구가 새로운 그룹의 생성인 경우에는 새로운 그룹 관리자를 생성하고, 그룹 정보(group context)를 관리할 수 있는 자료 구조를 생성한다. 기존의 그룹에 참여를 원하는 사용자 요구에 대해서는 사용자 정보를 해당 그룹 관리자에게 전달한다. 사용자 정보는 사용자 ID와 사용자의 위치 정보, 사용자가 참여하고 있는 그룹의 리스트로 구성된다. 공동작업 정보는 전체 그룹 정보들을 말한다. 그룹 정보는 그룹의 ID, 그룹 참여자들의 ID 리스트, 그룹 주제, 그룹의 수용 인원수 정보로 구성된다. 공동작업 관리자는 인터페이스를 통해서 전달되는 사용자의 요구에 따라서 사용자 정보와 공동작업 정보를 적절히 변경한다.

[알고리즘 1] 공동작업 관리자의 사용자 요구 처리

```
begin Method
    Check whether User is valid or invalid;
    If ValidUser then
        Case UserRequest
            GroupCreate :
                Create GroupManger, ApplicationManager;
                Update User/Service Infomations;
                Send Event to EventDamon;
            GroupDestroy :
                Delete GroupManager;
                Update User/Service Infomations;
                Send Event to Notification Module;
            GroupSearch, GetAllGroupInfo, GetAllUserInfo, GetUserInfo, GetGroupID :
                Send Informations to RequestUser;
            SendMessage :
                If SendType is unicast then
```

```

    Send Message to TargetUser through Notification
    Module;
    Else if SendType is multicast then
        Send Message to GroupManager;
    GroupJoin, GroupLeave :
        Send Information to TargetGroupManager;
    End Case;
    EndIf;
End Method;

```

3.2.2 그룹 관리자

그룹 관리자는 그룹 생성시 공동작업 관리자에 의해서 생성되며, 그룹의 상태에 대한 문맥(context)을 관리하고, 그룹 구성원들간의 상호 작용에 필요한 기능들을 수행한다. 또한, 공동작업 관리자를 통해 전달되는 사용자의 요구를 처리한다.

그룹에 대한 정보는 그룹 문맥을 이용해서 유지된다. 그룹 문맥에는 현재 그룹에 참여하고 있는 참여자들의 정보와 현재 의장의 정보, 발언권을 가진 사용자의 위치 등을 갖는다. 그룹 관리자는 최근에 처리한 사건에 해당하는 그룹 문맥을 유지하는데, 만약 새로운 사용자의 참여나 사용자 탈퇴, 발언권 이동등의 경우 해당하는 문맥으로 문맥전환이 이루어진다. 문맥전환이 이루어지면 해당 사실이 그룹 참여자들에게 전달된다.

그룹 참여자는 그룹 의장, 그룹 멤버, 방청객의 세 가지 역할 중에 하나를 수행하게 된다. 역할은 그룹 의장으로부터 그룹 참여시 부여되는데, 그룹 의장은 그룹 생성자가 되고, 다른 사람에게 의장 역할을 부여할 수 있다. 그룹 의장은 그룹에 참여하는 사용자들의 역할을 부여할 수 있는 권한 외에 그룹의 강제 종료와 그룹에 대한 정보를 변경할 수 있는 권한을 가지며, 그룹내의 발언권을 제어할 수 있는 권한을 가진다. (그림 7)은 그룹의 상태 관리를 위한 그룹 문맥의 형태이다.

```

enum Role (GroupChairman, GroupMember, Inspector)
struct Participant {
    string ClientID;
    string Alias;
    Role role;
};
typedef sequence<Participant> Participantlist;
struct GroupContext {
    unsigned short GroupID;
    ParticipantList List;
    Participant TokenOwner;
    Participant GroupChairman;
    string topic;
    TimeT duration;
    unsigned short max;
};

```

(그림 7) 그룹 문맥 형태
(Fig. 7) Group Context

사용자는 발언권 토큰을 가지고 있어야만 다른 사용자에게 데이터를 전송할 수 있다. 만약, 그룹 의장(GroupChairman)이 토큰을 요구(getToken)을 하는 경우에는 그룹 관리자는 즉각 토큰을 그룹 생성자에게 반환한다. 만약, 그룹 멤버가 토큰을 요구하는 경우에는 그룹 관리자는 그 사실을 의장에게 전달해서, 의장이 적절한 조치를 취할 수 있게 한다. 토큰 소지자의 변화나 새로운 멤버의 추가 혹은 탈퇴 등의 문맥 전환시 그룹 관리자는 해당 사건을 통지 모듈에게 전송한다. (그림 8)은 그룹 관리자의 인터페이스이다.

```

interface GroupManager {
    boolean giveRole( in string passwd, in string ClientID, in Role role );
    boolean requestToken( in string ClientID );
    boolean releaseToken( in string ClientID );
    boolean giveToken( in string ClientID );
    boolean fetchToken( in string passwd, in string ClientID );
    Participant getParticipantfomation(in string ClientID);
    boolean TokenOwner( in string GroupID );
    ParticipantList getParticipantList(in string ClientID);
    boolean SendMessageToAll(in string ClientID, in any Message);
    Participant GetChairmanInfo(in string ClientID);
    Participant TokenOwnerInfo(in string ClientID);
    Participant GetGroupmemberInfo(in string ClientID);
    Participant GetInspectorInfo(in string ClientID);
}

```

(그림 8) 그룹 관리자의 인터페이스
(Fig. 8) Group Manager's Interface

그룹 관리자의 인터페이스에서 제공하는 연산 중에서 SendMessageToAll은 사용자가 그룹 전체의 사용자들에게 메시지를 전달하는 경우, 공동작업 관리자에 의해서 호출된다. 그룹 관리자에서는 메시지를 보낸 사용자ID와 메시지 내용을 대상 사용자들의 정보와 함께 통지 모듈을 통해서 전달한다.

[알고리즘 2] 그룹 관리자의 사용자 요구 처리

```

begin Method
Case UserRequest
GroupJoin :
    User Role is determined by GroupChairman;
    Update GroupContext->List;
    Send signal to EventDaemon;
GroupLeave :
    Update GroupContext->List;
    Send signal to EventDaemon;
RequestToken, FetchToken :
    If Requestor is GroupChairman then
        Update GroupContext->hasToken;
        Send signal to all group members through Notification Module;

```

```

EndIf
If Requestor is GroupMember then
    Send Message to GroupChairman and the others
    memebers through Ntification Module;
If Requestor is Inspector then
    Send Deny_Message to Requestor;
GiveToken, ReleaseToken :
Update GroupContext->hasToken;
Send signal to all group members through Notifi-
cation Module;
GetChairmanInfo,TokenOwnerInfo,GetGroupmemberInfo,GetIn-
spectorInfo :
    Send informations to Requestor;
SendMessageToAll :
    Send Message and Targer Userlist to Notification Module;
End Case;
End Method;
    
```

3.2.3 응용 관리자와 시간 이벤트 관리자

그룹 구성원들은 여러 응용들을 이용해서 상호 작용한다. 이를 위해서 그룹 생성시 공동작업 관리자는 응용들의 정보를 관리하기 위해서 응용 관리자를 생성한다. 응용 관리자는 그룹에서 사용되고 있는 응용들의 추가와 삭제 등의 기능을 제공하며, 하나의 응용마다 생성하는 응용(Application) 객체를 관리한다. 응용 객체에는 모든 그룹 구성원들의 응용 포트(port) 정보를 관리한다. 이 정보는 실제 구성원들간의 멀티미디어 데이터 전송을 위해서 사용된다.

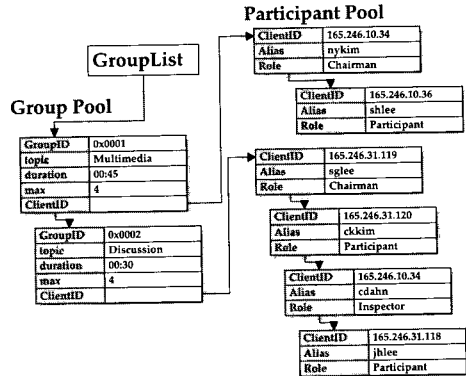
시간 이벤트 관리자는 시간 중속적인 그룹 활동을 가능하게 하기 위한 기능을 수행한다. 그룹 생성자는 그룹의 활동 시간을 미리 정해 둘 수 있으며, 지정된 시간에 그룹의 자동 소멸이 가능하다. 시간 이벤트 관리자는 시간 이벤트의 관리가 필요한 그룹별로 정보를 관리한다. 시간 이벤트 서비스는 사용자에게 의해 지정된 시간 이벤트를 처리한다. 해당 이벤트가 발생하면 그룹 관리자에게 전달한다.

4. 실험 및 평가

본 장에서는 3장에서 설계한 공동작업 서비스의 타당성을 보이기 위해 원격 교육 시스템을 구현에 필요한 기능들을 가지고 실험하였다. 실험 환경은 운영 체제로 윈도우 95, 윈도우 NT 4.0를 사용하였고, IONA사의 Orbix2.2(윈도우 95/NT 버전)을 미들웨어로 하여, 개발 언어는 Visual C++ 5.0을 사용하였다.

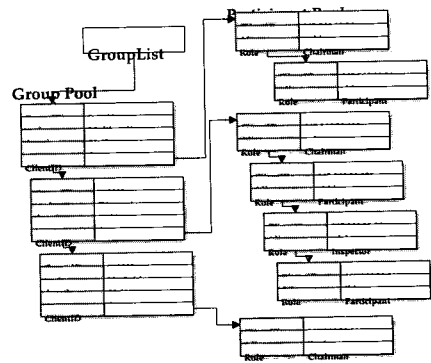
본 실험은 화상 교육 시스템에서 사용자 관리와 그룹 관리의 기능에 대한 것들만을 실험한다. (그림 9)는

서비스에서 관리하는 그룹과 그룹 사용자 정보의 형태를 보여준다.



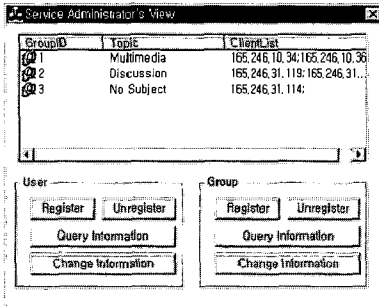
(그림 9) 그룹과 그룹 사용자 정보 (Fig. 9) The information about Group and Group-user

현재, 서비스에는 Multimedia와 Discussion의 제목을 가진 두개의 그룹이 활동중이며, 그림과 같은 사용자들의 정보 형태를 가지고 있다. 사용자의 새로운 그룹 생성 요구에 의해서 공동작업 관리자는 그룹 관리자와 응용 관리자를 생성하고, 통지 모듈을 통해서 다른 사용자들에게 그룹 생성 사건을 알린다. 그룹 생성 후 (그림 9)의 그룹과 사용자 정보가 (그림 10)과 같이 변경된다. (그림 10)은 No Subject 주제의 그룹에 대한 정보를 그룹 리스트에 추가하고, 그룹 생성자를 그룹의장으로 하는 사용자 정보를 추가한다.



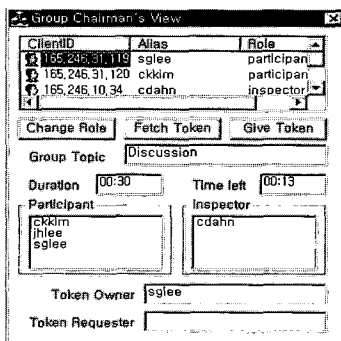
(그림 10) 변경된 그룹과 사용자 정보 (Fig. 10) The changed information about Group and Group-user

(그림 11)은 서비스 운영자를 위한 윈도우이다. 운영자는 서비스의 사용자와 그룹에 대한 등록, 해제, 정보 질의와 변경의 기능을 갖는다. 현재 서비스에는 세 개의 그룹이 등록되어 있다.



(그림 11) 서비스 운영자 윈도우
(Fig. 11) Service Administrator's window

서비스에 등록된 사용자는 그룹의 생성 또는 기존의 그룹에 참여할 수 있다. 사용자가 그룹 생성을 요구하는 경우, 그룹 생성자는 그룹 의장의 역할을 부여 받는다. 그룹 의장의 주된 역할은 그룹 전체의 발언권 제어와 참여자들의 역할 변경이다.



(그림 12) 그룹 의장을 위한 윈도우
(Fig. 12) Group Chairman's window

(그림 12)는 그룹 의장에게 제공되는 윈도우로, 그룹 전체 구성원들의 리스트와 발언권 제어 버튼, 그룹 밖의 그룹 정보를 출력할 수 있는 윈도우로 구성된다. 현재, 그룹 멤버의 역할로 ckkim, jhlee, sglee, 방청객 역할로 cdahn이 그룹에 참여 중이며, 발언권은 sglee가 가지고 있다.

본 실험은 현재 CORBA Telecom이 구현되지 않은

관계로 실제 멀티미디어 데이터를 이용하지는 못했다. 그러나, 제공하는 기능들을 원격 화상 교육시스템에서 적절히 이용할 수 있었으며, 실제로 응용 개발에 필요한 기능들을 올바르게 제공하였다.

5. 결 론

본 연구에서는 CORBA 환경에서 분산 멀티미디어 응용 개발 시 요구되는 공동작업 관리 기능을 제공할 수 있는 공동작업 서비스를 설계하였다.

설계한 공동작업 서비스는 응용 개발자가 CORBA 환경을 이용한 분산 멀티미디어 응용 개발 시 대부분의 멀티미디어 응용에서 필수적으로 요구하는 공동 작업 기능을 제공함으로써, 보다 쉽게 응용을 개발할 수 있도록 하였다. 따라서, 응용 개발자는 설계된 서비스로부터 공동작업 기능을 제공받고, 실제 응용 부분에 대한 기능만을 설계하게 함으로써, 전체 응용 개발에 소요되는 시간과 경비를 감소시킬 수 있다.

또한, 설계된 서비스는 다음과 같은 장점을 갖는다. 기능면에서는 기존의 공동작업 서비스에 비해서 보다 다양한 기능들을 제공함으로써, 다양한 응용 개발에 도움이 되도록 하였으며, 집중 제어 방식으로 설계됨으로써 보다 쉽게 정보를 관리하고, 사용자간의 상호 작용을 제어가 가능하다. 또한, 시간 이벤트 관리자의 도입으로 시간에 종속적인 그룹 활동이 가능하며, 이러한 기능은 주문형 비디오 시스템이나 원격 교육 시스템에서 유용하게 사용될 수 있다.

향후 연구 방향으로는 공동 저작 시스템과 같은 비동기화 협력 시스템들에 대한 기능을 추가하는 것과 현재, CORBA Telecom 스펙과의 연동 부분이 연구중이다.

참 고 문 헌

- [1] OMG, The Common Object Request Broker: Architecture and Specification Revision 2.0, OMG Document, 1997.
- [2] OMG, CORBA Telecom Specification, OMG Document, 1998.
- [3] IONA, "OrbixMX: Framework for Distributed Multimedia Exchange," <http://www.iona.com/products/orbix/orbixmx/index.html>, 1998.

[4] J. Dittrich, K. Eckert and P. Schoo, Design of a multimedia collaboration service for an environment of distributed procession platforms, International Workshop on Distributed Object Oriented Computing, Frankfurt/M, Germany, October 11, 1995.

[5] A. Manthe and S. Namuye, From Requirements to Services : Group Communication Support for Distributed Multimedia Systems, http://www.comp.lancs.ac.uk/computing/users/nigel/new_mpg/publications/1996_abstracts.htm, 1996.1

[6] T. H. Yun, J. Y. Kong and J. W. Hong, Object-Oriented Modeling of Distributed Multimedia Services, Proceedings of the IEEE International Conference on Communications (ICC), Montreal, Canada, June 1997.

[7] G. Coulson and D. Waddington, A CORBA Compliant Real-Time Multimedia Platform for Broadband Networks, Proceedings of the International Workshop on Trends in Distributed Systems (TreDS), Lecture Notes in Computer Science Vol.1161, Aachen, Germany, October 1996.

[8] S. Namuye, S. Simpson and D. Hutchison, An Architecture for Flexible Group Management Services, Proceedings of the 3rd Communications Networks Symposium, Manchester, U.K., 8-9 July 1996.

[9] M. Arango, P. Bates and R. Fish, Touring Machine : A Software Platform for Distributed Multimedia Applications, IFIP International Conference on Upper Layer Protocols, Architectures and Applications, 1992.

[10] C. Blum, P. Dubois and R. Molva, A Semi-Distributed Platform for the Support of CSCW Applications, Proceedings of the 1st International Distributed Conference IDC95, 1995.

[11] J. Rezende, A. Mauthe and D. Hutchison, M-Connection Service : A Multicast Service for Distributed Multimedia Applications, Proceedings

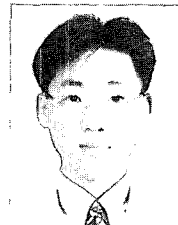
of the 2nd COST 237 Workshop on Teleservices and Multimedia Communications, Copenhagen, Denmark, 1995.



김 남 용

e-mail : nykim@shinheung-c.ac.kr
 1985년 인하대학교 전자계산학과 (학사)
 1989년 인하대학교 대학원 전자계산학과(이학석사)
 1995년 인하대학교 대학원 전자계산학과 박사과정 수료

1990년~현재 신홍대학 전산정보처리과 조교수
 관심분야 : Software Engineering, 분산 객체 컴퓨팅, 원격 교육



이 승 근

e-mail : sglee@selab.inha.ac.kr
 1996년 인하대학교 전자계산공학과(학사)
 1998년 인하대학교 전자계산공학과(공학석사)
 1998년~현재 인하대학교 전자계산공학과 박사과정

관심분야 : 분산 멀티미디어, 분산 객체 컴퓨팅, CSCW, 원격 교육

왕 창 종

e-mail : cjwangse@dragon.inha.ac.kr
 1964년 고려대학교 물리학과(학사)
 1975년 성균관대학교(경영학 석사)
 1981년~1990년 인하대학교 전자계산소장
 1992년~1993년 정보과학회 부회장, 전산교육연구회 위원장
 1979년~현재 인하대학교 전자계산공학과 교수
 관심분야 : Software Engineering, 멀티미디어, 분산 객체 컴퓨팅, 전산교육