

□ 기술개설 □

객체지향 검색형 멀티미디어 서비스 개발 환경

한국전자통신연구소 이범식* · 홍철의* · 조기환* · 임기욱**

● 목 차 ●	
1. 서 론	3.1 개요
2. OMA/CORBA	3.2 DMCF의 특성
2.1 개요	3.3 검색형 지원을 위한 DMCF의 기능
2.2 CORBA	3.4 DMCF의 구조
2.3 객체 서비스	3.5 응용 프로그래밍 인터페이스(API)
2.4 공통 지원 기능	4. 응용 프로그램 개발 예
3. DMCF	5. 결 론

1. 서 론

객체지향 기술은 Smalltalk, C++ 등의 객체지향 언어에 의해 확산되기 시작하여 여러 분야에 빠르게 적용되고 있다. 그 이유로 기존의 구조적 설계 개념은 복잡하고 다양한 사용자의 요구를 충족시키지 못할 뿐 아니라 멀티미디어, 그래픽 사용자 인터페이스, 클라이언트/서버 등 새로운 환경에 적절하지 못하기 때문이다[1].

객체지향 개념은 분산 시스템의 문제중의 하나인 응용 프로그램 개발이 어렵다는 단점을 해결하기 위해 적용되었는데, 그 결과로 OMG(Object Management Group)에서 분산 객체 컴퓨팅 표준 구조로 OMA(Object Management Architecture)를 제안하였다[2][3][4].

OMA가 분산 객체 실행 환경을 지원함으로써, 응용 프로그램은 운영체제와 프로그래밍 언어에 독립적이며 표준화된 객체 인터페이스를 통하여 네트워크의 다른 곳에서 실행되는

응용 프로그램과 통신할 수 있어 분산 프로그램 개발을 쉽게 해준다.

그런데 검색형 정보 처리 서비스는 WWW(World Wide Web)[5]의 급격한 확산에서 보듯 기본적으로 분산 환경을 가정하고 있으며, 정보 자체도 단순한 텍스트가 아닌 그래픽, 오디오, 텍스트 등이 혼합된 멀티미디어 객체로 구성되어 있다. 이러한 분산 멀티미디어 환경에서 응용 서비스를 개발하기란 쉬운 일이 아니다. 따라서 분산된 환경에서 멀티미디어 서비스를 개발하기 위해 OMA의 분산 객체 프레임워크인 CORBA(Common Object Request Broker Architecture)[2][3][4] 환경을 이용하는 것은 하나의 자연스러운 흐름이다.

현재 멀티미디어 정보 검색 시스템의 한 예로 WWW를 들 수 있는데, 사용하기가 쉬워 매우 빠른 속도로 확산되어 거의 산업체 표준이 되었다. 그러나 WWW 환경은 문서 검색을 기본으로하여 설계되었기 때문에 연속적인 스트림을 처리하기에는 부적합하다[6].

그리고 DAVIC(Digital Audio-Visual Council)[7]에서는 검색형 멀티미디어 서비스의 구축을 위한 개방적 구조의 사양 정의 작업

*정 회원

**종신회원

을 수행하고 있다. 최근에 발표된 DAVIC 1.0에서는 서비스를 위한 구조로 서비스 제공자, 전달 시스템, 서비스 사용자, 정보 제공자의 네 요소로 분할하여 각 요소에 대한 인터페이스를 정의하고 있다. 현재 Sony, Digital 등에서 프로토타입을 구현하고 있다.

본 논문에서는 CORBA를 기반으로 하여 이 질적인 분산 환경에서 검색형 멀티미디어 서비스를 쉽게 개발할 수 있는 개발 환경으로 DMCF(Distributed Multimedia Common Facility)를 제안한다. 이 DMCF는 CORBA의 수직 공통 지원 기능(Vertical Common Facility)으로 개발자에게 제공된다.

DMCF는 검색형 서비스의 효율적인 지원을 목표로 하고, 연속적인 미디어로 MPEG-2 스트림을 효율적으로 전달하기 위해 제안된 ISO/IEC 13818-6(DSM-CC : Digital Storage Media Command & Control)[8] 규격을 기반으로 DAVIC 모델을 참조하여 설계되었다. 검색을 효율적으로 지원하기 위해서, DMCF는 메타 데이터를 관리하는 데이터베이스 관리 기능과 연속적인 스트림 데이터를 위한 저장 시스템 그리고 서비스의 질을 보장하기 위한 QoS(Quality of Service)[9] 관리 기능을 제공한다. 또한 DMCF는 WWW 환경과도 연동하여 동작하도록 설계되었다. 즉, WWW 인터페이스를 이용하여 정보를 검색할 때, 데이터베이스 시스템의 위치나 종류는 사용자에게 투명하여 일관된 사용자 인터페이스를 제공한다. 그리고 WWW가 검색한 문서가 연속적인 스트림을 포함한 경우 DMCF는 스트림의 전송과 해당 스트림에 대한 제어를 지원한다.

여기에서 언급하는 검색형 멀티미디어 서비스는 초고속 통신망 환경에서 운용되는 Movie On-Demand, Near Video On-Demand, 게임, Karaoke On-Demand, News On-Demand, Teleshopping 등과 같은 Video On-Demand 유형의 DAVIC에서 정의한 응용 시스템이다.

본 논문의 구성은 다음과 같다. 2절에서는 분산 시스템에서 객체 지향을 지원하는 OMA/CORBA에 관하여 소개하며, 3절에서는 현재 병렬프로그래밍 연구실에서 수행중인 멀티미디어 검색형 서비스를 위한 프레임워크(Frame-

work)[10][11]인 DMCF에 관하여 소개한다. 그리고 4절에서는 VOD(Video On-Demand) 서비스를 제안한 DMCF를 이용하여 설계한 시스템을 설명하고 마지막으로 5절에서 결론을 맺는다.

2. OMA/CORBA

2.1 개요

분산 시스템이 가진 문제를 해결하기 위해 OMG에서는 분산 객체 컴퓨팅 표준 구조로 OMA 및 CORBA를 제안하였다.

OMA는 OOP 언어, 데이터베이스 시스템, 객체 라이브러리에 대한 용어를 명확히 규정하였으며, 객체지향 시스템에 대한 추상 수준의 구조 및 목적, 분산 객체 시스템에서의 참조 모델을 정의하였다. 이 OMA(그림 1)는 응용 객체(Application Objects), 객체 요청 중계자(ORB, Object Request Broker), 객체 서비스(Object Services), 공통 지원 기능(Common Facilities)의 4 가지 객체로 구성되어 있다.

응용 객체란 OMG에서 정의한 규격대로 작성된 응용 프로그램으로 서로간에는 객체 요청 중계자를 통하여 통신한다. 예를 들면, 워드프로세서, 스프레드시트 그리고 VOD 등이 있다. 객체 요청 중계자는 OMA 구조에서 가장 중요한 부분으로 CORBA로 표준화가 되었으며 서로 다른 곳에 존재하는 객체간의 메시지(요청/응답)를 전달하고 표준화하는 층이다. 객체 서비스는 모든 객체를 구현할 때 유용하게 사용할 수 있는 OMG에서 정의한 공통되는 서비스

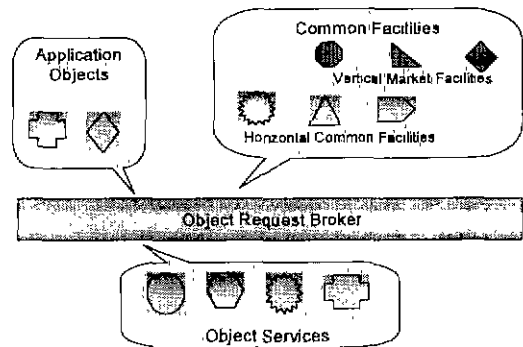


그림 1 OMA 구조

들이다. 예를 들면, 객체 이름 관리(naming) 서비스, 객체 사이의 사건을 교환하기 위한 사건(event) 서비스 등이 있다. 공통 지원 기능은 응용 프로그램간에 객체를 공유하거나 OMA를 기반으로 하는 응용 프로그램을 쉽게 생성하기 위한 프레임워크이다. 예를 들면, 복합문서 지원과 같은 데스크탑 인터페이스를 제공하여 응용 프로그램 개발자가 쉽게 복합문서를 처리하는 프로그램을 개발 할 수 있게 한다.

2.2 CORBA

객체 요청 중개자는 OMA 구조에서 가장 핵심이 되는 부분으로 1991년에 OMG는 참조 모델의 객체 요청 중개자 부분에 대한 인터페이스 기법을 표준화하는 CORBA를 규정하였다. CORBA 규정은 각 객체가 메시지를 전달할 때, 객체사이에 위치와 구현에 관한 투명성을 보장한다. 이러한 방법으로 응용 프로그램과 객체 시스템은 이 기종 분산 환경 하에서 서로 다른 시스템에서 실행되면서도 정보를 용이하게 교환한다. 그림 2는 CORBA 구조를 보여주고 있다.

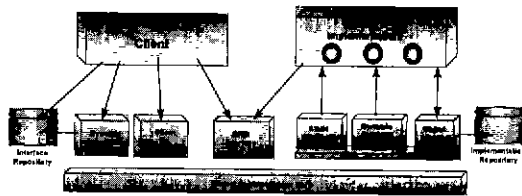


그림 2 CORBA 구조

2.2.1 CORBA의 동작 특성

객체 요청 중개자는 이 기종 객체간의 통신에서 중심 역할을 한다. 각 객체는 객체 요청 중개자에 등록된 후 서로 다른 언어나 환경에서도 객체 인터페이스를 통하여 통신할 수 있다. 즉, 객체 요청 중개자는 응용 객체와 객체 요청 중개자 사이에 있는 어댑터의 일종인 proxies를 사용하여 클라이언트로부터 전달된 메시지를 등록된 객체에 전송한다. 그리고 OMG는 이 기종간의 호환성을 위해 CORBA 2.0을 발표하였다. CORBA 2.0은 서로 다른 구현 사이에서의 상호 호환성에 중점을 두어, 응용 프로그램과 데이터 형식에 영향을 받지

않도록 규정을 마련하였다. 그 특징은 다음과 같다.

- ◆ 이진 상호 운용성 보장
- ◆ 멀티미디어 객체 지원
- ◆ 복제(Replication) 지원
- ◆ 비동기성 메시지 전달 지원
- ◆ 동시성(Concurrency) 지원
- ◆ TCP/IP 외 특정 프로토콜(DCE) 지원
- ◆ Cross 영역에 대한 지원 (브릿징)
- ◆ 객체 참조 포맷 및 global identifier 정의

2.2.2 구성 요소

- ◆ ORB core - 객체간의 상호 연결 버스로 객체간 통신을 위한 투명성을 보장하고 클라이언트/서버 객체간에 상호 연결을 가능하게 해준다.
- ◆ IDL(Interface Definition Language) - 객체의 인터페이스와 데이터 구조를 정의하기 위한 언어로서 프로그래밍 언어는 아니다.
- ◆ Client IDL stub - 클라이언트 쪽의 객체에 대한 정적인 인터페이스를 제공한다.
- ◆ Dynamic Invocation Interface - 실행 시간에 동적으로 객체를 호출할 수 있게 하는 인터페이스를 제공한다.
- ◆ Interface Repository - IDL에서 정의한 인터페이스, 메소드, 상수, 타입 선언 등을 저장하여 객체를 호출할 때 이용 가능하게 해준다.
- ◆ ORB Interface - ORB core가 제공하는 서비스를 호출하는 인터페이스를 제공한다.
- ◆ IDL skeleton - 서버 쪽에 구현된 특정 객체를 정적으로 호출하기 위한 기능을 한다.
- ◆ Dynamic Skeleton Interface : 서버 쪽에 구현된 특정 객체를 동적으로 호출하기 위한 기능을 한다
- ◆ Object Adapter - ORB core가 객체 구현 부에 접근할 수 있게 해주는 기본 기능이다.
- ◆ Implementation Repository - 실행 시 서버 쪽에 구현된 객체에 관한 정보를 저

장하여 객체 호출 시 이용 가능하게 해준다.

2.3 객체 서비스

객체 서비스란 분산 환경에서 객체 공유를 위해 기본적으로 필요한 서비스 집합으로 IDL로 기술된 인터페이스와 함께 컴포넌트 형태로 패키징화 된 서비스들의 집합이다. 이 서비스는 객체 요청 중계자의 기능을 보완해준다. 다음은 OMG에서 정의한 서비스들이다.

- ◆ naming service - 이름으로 객체를 찾을 수 있게 하는 인터페이스
- ◆ event service - 사건을 동적으로 등록, 삭제하고, 객체들 사이에 분배하는 인터페이스
- ◆ persistence service - 객체를 영구 저장 장치에 저장하는 인터페이스
- ◆ life-cycle service - 생성, 복사, 삭제 등에 관한 인터페이스
- ◆ transaction service - 객체들 사이에서 2 단계 commit 협동을 위한 인터페이스
- ◆ concurrency control service - 트랜잭션이나 쓰레드를 위한 잠금 기능에 관한 인터페이스
- ◆ relationship service - 서로 모르는 객체 간에 동적인 연관성을 부여하는 인터페이스
- ◆ externalization service - 데이터를 객체로 읽어들이거나 다른 객체로 전송하는 인터페이스
- ◆ query service - 객체에 대한 조회 인터페이스
- ◆ licensing service - 객체의 사용 상태에 관한 정보를 측정하는 인터페이스
- ◆ property service - 이름과 값은 쌍인 속성을 객체와 연관하는 인터페이스
- ◆ security service - 객체 시스템의 보안을 책임지는 인터페이스
- ◆ time service - 분산된 시스템에서 공통의 클럭(Clock)을 유지하는 인터페이스
- ◆ trader service - 클라이언트가 등록 받은 서비스와 요청을 연관시키는 인터페이스

- ◆ collections service - 객체를 묶음으로 조작하는 인터페이스(Stack, Queue, Tree, ...)
- ◆ change management service - 객체에 대한 진화의 기록과 버전을 관리하는 인터페이스

2.4 공통 지원 기능

OMA의 구성 요소 가운데 하나인 공통 지원 기능은 광범위한 응용 분야에 공통적으로 적용될 수 있는 고 수준의 서비스, 또는 특정 응용 분야를 위한 고급 기능들의 집합을 의미한다. 전자의 경우는 대부분의 응용 분야에 공통적으로 제공되는 서비스를 대표함으로써 수평 공통 지원 기능(horizontal common facilities)이라고 하고, 후자의 경우는 어느 정도 독립적인 응용 분야에 한정되기 때문에 수직 공통 지원 기능(vertical common facilities)이라 부른다. 이들은 CORBA 기반 객체 인터페이스를 통해서 참조될 수 있다. 이 공통 지원 기능은 그 자체로 프레임워크를 구성하며, 각각의 프레임워크는 협동하여 응용 프로그램의 기반을 제공한다.

객체 서비스가 모든 시스템에서 지원되는 것과 달리 공통 지원 기능은 선택적으로 지원된다. 응용 프로그램 개발자는 공통 지원 기능을 활용하여 보다 손쉽고 효과적으로 응용 프로그램을 개발할 수 있다.

2.4.1 수평 공통 지원 기능

- ◆ 사용자 인터페이스(user interface) - 그래픽 사용자 인터페이스(Graphic User Interface)와 사용자 데스크 탑 환경 관리에 관한 인터페이스
- ◆ 정보 관리(information management) - 정보 전달, 데이터 압축 및 해제, 정보 저장 및 복원에 관한 인터페이스
- ◆ 시스템 관리(system management) - 시스템 자원 관리, 틀 통합에 관한 인터페이스
- ◆ 태스크 관리(task management) - Workflow 관리, 에이전트, 작업 자동화에 관한 인터페이스

2.4.2 수직 공통 지원 기능

수직 공통 지원 기능은 객체 서비스와 수평 공통 지원 기능을 이용하여 매우 다양하게 개발될 수 있는데, 바로 특정 분야에서의 프레임워크를 나타낸다. 따라서, 응용 프로그램 개발자는 프레임워크를 이용함으로써 기본 동작을 상속받거나 새로운 동작으로 덧쓰기 함으로써 기능을 변경하는 것이 가능하다. 뿐만 아니라 한 프레임워크에서 다른 프레임워크를 CORBA의 ORB를 통해 호출할 수 있어 응용 프로그램 개발을 매우 쉽게 한다.

수직 공통 지원 기능은 그 특성상 한 분야에서 여러 형태로 구현될 수 있고, 각각의 분야마다 구현될 수도 있다. 그 중 몇 가지를 나열하면 다음과 같다.

- ◆ imagery
- ◆ information superhighway
- ◆ computer integrated manufacturing
- ◆ distributed simulation
- ◆ accounting
- ◆ multimedia

3. DMCF (Distributed Multimedia Common Facility)

3.1 개요

DMCF는 한국전자통신연구소 병렬프로그래밍 연구실에서 수행중인 RODEO(Research on Object-oriented multimedia application Development Environment on Object man-

agement architecture : 그림 3) 시스템의 한 서브시스템인 검색을 위한 프레임워크이다. RODEO 시스템은 DMCF, 재사용 시스템(RESMO), 시각 프로그래밍 도구 서브시스템(VIOLA)으로 구성되어 있다. 이중에서 재사용 시스템 서브시스템은 한국전자통신연구소 정보기술개발단, 시각 프로그래밍 도구 서브시스템은 시스템공학연구소에서 개발하고 있다.

DMCF의 목표는 이질적인 분산 환경에서 일반 미디어뿐만 아니라 시간 의존적인 미디어도 처리할 수 있는 검색형 멀티미디어 서비스를 지원하는 프레임워크를 제공하는데 있다. DMCF는 분산된 환경에서 서로 통신하기 위해 OMG의 CORBA를 이용하고 MPEG-2 스트림을 지원한다.

DMCF가 운용되는 환경은 MPEG-2 스트림을 지원하기 위해 ATM 망을 이용하고, ATM 망 위에서 TCP/IP 프로토콜을 지원하는 에뮬레이션 소프트웨어를 가진 UNIX와 Windows95 시스템이다.

검색의 결과로 사용자에게 되돌려지는 객체는 응용 프로그램의 요구 사항에 맞추어 프로그래밍이 가능하다. 결과 객체의 한 예는 WWW 환경에서 사용하는 HTML(HyperText Markup Language[12]) 문서의 형태를 생각할 수 있다.

DMCF는 MPEG-2 스트림을 제어하기 위해 DSM-CC에서 정의한 프로토콜과 인터페이스를 사용한다. 그리고 검색을 효율적으로 지원하고 데이터베이스에 대한 투명성을 제공하기 위한 기능을 고려하여 설계하였다. 또 다른 고려 사항으로, MPEG-2 스트림은 막대한 저장 용량을 필요로 할뿐만 아니라 전송에 필요한 대역폭도 매우 크기 때문에 저장 시스템을 효율적으로 구현하지 않으면 안된다. DMCF에서는 계층적인 저장 구조를 사용하여 이러한 문제를 해결할 수 있게 하였다. 마지막으로 서비스의 질을 보장하기 위한 기능으로 QoS 관리 기능을 추가하였다.

DMCF의 대부분의 기능은 서버 쪽에 구현이 되며 클라이언트 쪽은 MPEG-2 스트림 수신과 비디오 제어(VCR-like control)를 위한 CORBA 인터페이스가 구현된다. DMCF에서

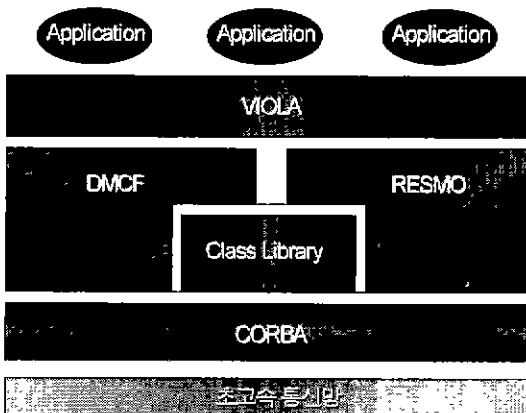


그림 3 RODEO 시스템의 논리적 구조

는 결과로 되돌려지는 객체를 화면에 표현해 주기 위한 기능은 가지고 있지 않으며 단지 그 객체 안에 포함된 MPEG-2 스트림에 대한 서비스만 제공해준다. 즉, 클라이언트 응용 프로그램 개발자는 DMCF를 이용함으로써 MPEG-2 스트림을 제어하는 모듈을 응용 프로그램 안에 삽입할 수 있다. 한 예로 MPEG-2 뷰어(viewer)가 Netscape Navigator의 한 plug-in으로 삽입될 수 있다.

CORBA의 객체로서 멀티미디어 객체와 검색 지원 기능을 표현하는 것은 많은 장점을 가지고 있다. CORBA 객체는 자신이 바로 독립적으로 동작하는 단위로 일종의 소프트웨어 IC 역할을 하여 쉽게 재사용이 가능하며, 또한 객체로 표현되었기 때문에 상속을 통하여 그 기능을 확장하거나 대체할 수 있어 응용 프로그램 개발에 많은 유연성을 준다. 그리고 CORBA가 분산환경을 지원하기 때문에 서비스를 제공하는 서버를 기능적으로 분산시킬 수 있다. 그림 4는 DMCF가 OMA 공통 지원 기능으로 제공되며, 응용 프로그램 개발자는 DMCF API(Application Programming Interface)를 이용하여 검색형 서비스를 개발할 수 있다는 것을 보여주고 있다.

3.2 DMCF의 특성

- ◆ 초고속 통신망에서 다수의 사용자에게 실

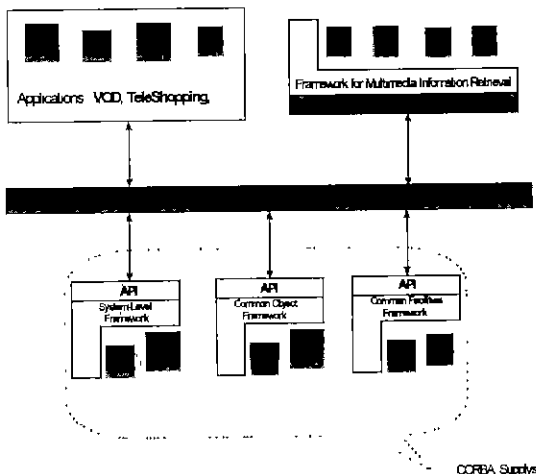


그림 4 DMCF in OMA

시간 검색형 멀티미디어 서비스를 제공할 수 있는 환경을 지원한다.

- ◆ 분산 객체 컴퓨팅 환경인 CORBA를 이용하여 이 기종간에 검색형 멀티미디어 서비스를 가능하게 하는 공통 기능으로 제공된다.
- ◆ 분산 데이터베이스를 통합 관리하는 브로커를 제공함으로써 사용자에게 투명성을 제공한다.
- ◆ 연속적인 미디어를 효율적으로 처리하기 위해 계층적인 저장 시스템을 지원한다.
- ◆ 서비스의 질을 보장하기 위한 QoS기능을 지원한다.
- ◆ 소프트웨어 IC 역할을 하여 쉽게 재사용이 가능하며, 또한 객체로 표현되었기 때문에 상속을 통하여 그 기능을 확장하거나 대체할 수 있다.
- ◆ DSM-CC 규격을 기본으로 확장하였다.
- ◆ WWW 환경과 연동하여 동작할 수 있다.

3.3 검색형 지원을 위한 DMCF의 기능

검색형 서비스를 효율적으로 지원하기 위한 기능은 DAVIC[7] 표준에 잘 표현되어 있다. DMCF는 그 중에서 다음의 기능을 만족하도록 설계하였다.

- ◆ 비트 전달(Bit Transport) 기능 - 비트 전달 기능은 주어진 대역 폭에 대한 물리적 논리적 링크 및 연결 지점간의 논리적 연결을 설정하기 위한 정보 교환 기능을 제공한다.
- ◆ 세션(Session) 기능 - 세션 기능은 논리적 연결을 설정 변경하고자 할 경우에 호출되어 비트 전달 기능을 제어한다. 연결 형태는 사전에 결정되거나 서비스 중간에 변화될 수 있으며, 연결 설정에 필요한 파라미터들은 세션과 비트 전달간에 사전에 상호 약속된 프로토콜에 의해 이루어진다.
- ◆ 접근 제어(Access Control) 기능 - 접근 제어 기능은 사용자에게 대한 인증 기능을 제공하며 통신망에 대한 액세스 권한을 결정한다. 또한 특정한 응용 서비스와 관련한 내용물, 서비스 등에 대한 확인 기

능에 대한 제어를 한다.

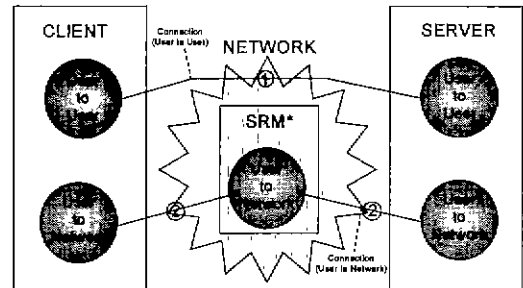
- ◆ 응용 프로그램 시작(Application Launch) 기능 - 응용 프로그램 수행 환경을 제공한다. 응용 프로그램 코드는 클라이언트에 상주하거나 서버로부터 세션을 통해 제공받을 수도 있다.
- ◆ 응용 제어(Application Control) 기능 - 응용 서비스에 대한 제어 기능으로 일시 정지, 재개, 고속 전 후진, 오디오 속도 제어 등이 있다.
- ◆ 사용 데이터(Usage Data) - 사용자의 자원 및 응용 서비스 사용에 관한 데이터의 수집, 저장, 공급 기능을 제공한다. 수집된 데이터들은 사용자 계정에 대한 기본 자료로 사용되며, 정보 제공자에게는 시장 조사의 자료로 시스템 관리자에게는 자원 이용도에 대한 분석 자료로 각각 이용된다.
- ◆ 사용자 프로파일(User Profile) - 각 사용자에 대한 정보 및 이전의 행동 양식에 대한 정보를 저장한 후 활용한다.
- ◆ 메타데이터 관리(MetaData Management) 기능 - 서비스의 종류 선택과 정보를 추출하기 위한 색인을 관리한다.
- ◆ 저장 관리(Storage Management) 기능 - 연속적인 미디어를 효율적으로 처리하기 위한 저장 장치 등을 관리한다.

3.4 DMCF의 구조

3.4.1 DSM-CC 참조 모델

DSM-CC는 MPEG-2 스트림을 관리하기 위한 제어 기능과 동작을 기술한 프로토콜의 집합이다. 그러나 DSM-CC에서 사용한 개념은 일반적인 응용에서도 사용할 수 있다. 그림 5에서처럼 DSM-CC는 전체 시스템을 논리적으로 세 개의 서브 시스템으로 나누었는데, 클라이언트, 서버 그리고 세션과 자원 관리자(Session & Resource Manager)이다. 이 중에서 스트림의 전달과 제어는 클라이언트와 서버 사이에서 일어나며 SRM은 클라이언트와 서버사이의 세션과 자원 할당의 역할을 담당한다.

DSM-CC에서 신호의 전달은 반드시 각 엔터티(그림 5의 동그라미)간에 일어난다. 그림



*May provide session, connection, and configuration management and control

그림 5 DSM-CC 참조 모델

에서 ①은 User-to-User(U-U) 프로토콜이며, ②는 User-to-Network(U-N) 프로토콜이다. U-N 프로토콜은 주로 서비스를 제공하기 위한 세션의 설정, 변경, 해제 관리 및 네트워크와 자원관리를 담당하는데 이 인터페이스는 메시지 전달의 형태를 가지고 있다. U-U 프로토콜은 실제 서비스를 지원하기 위한 기능을 기술하고 있는데 그 인터페이스는 CORBA IDL 객체로서 정의되어 있다.

3.4.2 DMCF의 구조

그림 6은 DSM-CC 모델에 검색을 위한 기능을 보강한 DMCF의 논리적 구조로 저장소 관리, 서비스의 질 관리 그리고 메타 데이터를 관리하는 기능과 DSM-CC 기본 기능(SRM, DSM-CC U-U 라이브러리, SessionGateway, ServiceGateway, etc.)으로 구성되어 있다.

DMCF에서 저장소는 비디오 스트림(MPEG-2)을 효율적으로 처리하도록 설계되

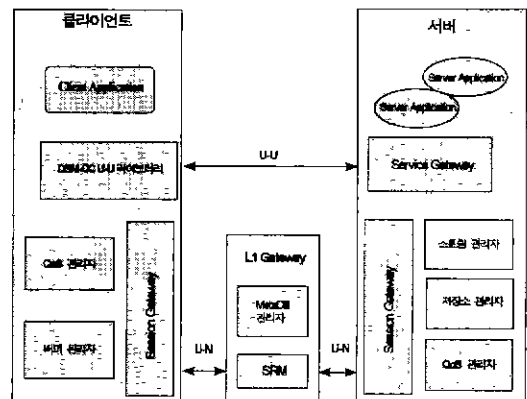


그림 6 DMCF의 논리적 구조

었다. 이를 위해 저장소는 계층적인 구조로 설계되었으며 응용 프로그램 개발자에게 직접적인 인터페이스를 제공하지는 않는다. 저장소는 DMCF API의 스트림 인터페이스를 구현할 때 전송과 제어를 위한 기능으로 이용된다.

L1 Gateway의 MetaDB 관리자는 응용 서비스(예 : VOD)가 필요로 하는 특별한 데이터(예 : VOD 서비스에서 영화 제목)를 관리하기 위한 인터페이스를 제공한다. 응용 프로그램 개발자는 이 인터페이스를 이용하여 데이터베이스 시스템에 투명하게 접근 할 수 있다.

QoS 관리자는 SRM과 저장소 관리자 그리고 시스템의 파라미터 값을 제어하기 위한 인터페이스로 제공이 되며, 응용 서비스가 물리적으로 분산되어 있는 환경에서도 서비스 질에 관한 요구를 처리 가능하게 해 준다. 실제 QoS를 만족하기 위한 기능은 DMCF의 각 기능에 분산되어 구현이 된다.

그리고 DMCF의 각 기능은 CORBA 규격을 만족하는 객체로 OMA의 객체 서비스와 공통 지원 기능을 이용하여 구현되기에 DMCF의 기능을 바꾸거나 확장하는 것은 객체를 상속받거나, 새로운 동작으로 덧쓰기 함으로써 가능하다.

DMCF를 실제로 구현할 때는 클라이언트 응용 프로그램을 위한 부분과 서버 응용 프로그램을 위한 부분을 따로 구현한다. 클라이언트 부분은 서버의 일부만 구현하면 되거나 서버 쪽 구현에 대한 인터페이스만을 가지고 있다.

3.4.3 저장소 관리자

저장소 관리자는 다양한 미디어들에 대해서 연속성과 효율성 및 실시간 검색이 보장되어야 하고, VCR-like 기능을 지원할 수 있어야 하며, 동시에 가능한 저렴한 가격으로 다수의 사용자에게 양질의 서비스가 가능해야 한다[13]. DMCF는 대규모의 비디오 스트림 저장을 위하여 분산 계층적인 저장 구조를 도입하여 비디오 스트림(MPEG-2)에 대한 비디오 제어(VCR-like 제어)를 효율적으로 처리하고 실시간으로 전송할 수 있게 설계하였다.

기존 연구 결과로 Berkely VOD 시스템[14]의 경우 하드디스크를 3차 저장 장치의 캐시로

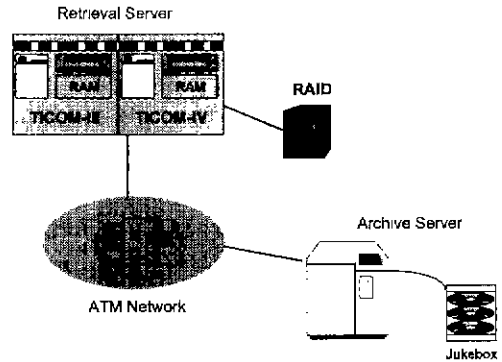


그림 7 계층적인 저장 구조

사용하는 계층적인 저장 구조를 채택한 반면 DMCF는 위에서 언급한 요구 사항을 만족하기 위해 그림 7과 같이 저장소를 RAM, 하드디스크(RAID), 3차 저장 장치 형태의 계층적 구조로 구성한다. 그리고 이러한 3단계 계층 구조는 사용자 등록 시에 부여되는 서비스 등급에 따라 최적의 가격 대 서비스 효율을 제공할 수 있도록 관리된다.

DMCF 저장시스템의 3차 저장 장치인 Archive Server는 물리적으로 분산이 되어 여러 개의 검색 서버에게 서비스를 제공할 수 있어 가격대 성능의 효율을 높일 수 있다. 다음은 저장소 관리자가 제공하는 기능이다.

- ◆ 신호도가 높은 멀티미디어 객체는 빠른 서비스가 가능한 저장소 즉, RAM에 배치하고 신호도가 낮은 객체는 보조 저장소 즉, 3차 저장 장치에 배치하여 서비스 효율성을 제고한다.
- ◆ 폭 넓고 다양한 멀티미디어 문서 형태를 처리한다.
- ◆ 사용자가 요구하는 서비스 수준을 만족하기 위해서는 필요시 높은 계층 저장소 등에 위치하고 있는 객체들이 낮은 계층 저장소로 미리 옮겨지거나 옮겨진 객체가 일정 기간 남아있게 하는 등의 동적인 저장소 관리가 제공된다.

3.4.4 MetaDB 관리자

멀티미디어 데이터의 특징으로 데이터가 대용량이라는 것과 비정형적이며 실시간 처리를 요한다는 것이다. 따라서 검색형 멀티미디어

서비스를 위해서는 방대하고 가변적인 멀티미디어 데이터를 효과적으로 저장 관리하고, 고속으로 질의 검색할 수 있는 데이터베이스 기술과 다양한 멀티미디어 정보를 효율적으로 처리할 수 있는 멀티미디어 처리 기술이 요구된다.

DMCF에서는 데이터베이스 브로커(DB Broker)를 제공함으로써 현재의 멀티미디어 데이터베이스 기술을 이용할 수 있게 하였다. 즉, MetaDB 시스템은 이질적이고 분산된 환경 하에서 복수의 데이터베이스에 대한 투명성을 클라이언트에 제공하여 접근 방법과 서비스에 대한 투명성을 보장한다.

DB 브로커가 동작하는 과정을 보면 클라이언트는 DB 브로커에 데이터를 요청하면 DB 브로커는 요청된 정보를 가지고 데이터베이스가 인식할 수 있는 형태로 변환하여 전달한다. DB 브로커는 데이터베이스로부터 요청한 응답에 관한 결과를 받아서 클라이언트에게 전달한다.

DB 브로커는 DBMS 등록 관리기, 서비스 처리기, DBMS 연결 라우터, 스케줄러로 구성된다.

신규 메타 데이터를 등록하거나 기존 메타 데이터를 변경하는 메타 데이터 관리기는 서비스 처리기 API를 이용하여 구현이 되며, 스케줄러는 각 서버에 대한 부하를 조절하여 서비스를 처리할 서버를 연결시켜 주는 기능을 제공한다.

다음은 MetaDB 관리자의 특징을 열거하였다.

- ◆ 데이터베이스 서버로는 관계형 서버, 객체지향 서버, 멀티미디어 서버를 모두 지원한다.
- ◆ 객체 형태로 요청을 접수하여 해당 서버에 대한 SQL 요청으로 변환한다.
- ◆ 서버로부터의 결과를 클라이언트가 요청한 객체 형태로 재구성한다.
- ◆ 객체지향 응용 프로그램을 위한 멀티미디어 데이터베이스 API를 제공한다.
- ◆ 사용자 정보 및 시스템 정보 등에 관한 로그 기록을 관리한다.

3.4.5 QoS 관리자

분산 멀티미디어 서비스 환경에서는 제한된 시스템 자원과 네트워크 대역 폭을 바탕으로, 여러 사용자로부터 요구되는 서비스를 일정 수준 이상의 질로써 제공하기 위하여 서비스 질(QoS) 관리가 필수적이다.

특히 VOD와 같이 연속적인 미디어 스트림 전달이 요구되는 응용 서비스에서는 내부적으로 연결된 시스템 (서버, 네트워크, 클라이언트) 각각의 요소들로부터 서비스 질(QoS)이 보장되어야 한다.

QoS는 응용 서비스의 주어진 기능을 수행하는데 필요한 분산 멀티미디어의 양과 질의 특성에 대한 집합으로 정의된다.

이 QoS를 제어하는 파라메타는 최대 값과 최소 값의 쌍으로 표현된다. 파라메타는 성능 지향, 형식 지향, 동기화 지향, 가격 지향, 사용자 지향으로 분류된다. QoS를 위한 실제 구현은 DMCF 시스템의 각 요소에서 구현이 되며 사용자에게는 QoS 파라메터 값의 설정, 수정, 조회 등과 같은 QoS 관리에 필요한 인터페이스를 제공한다.

QoS 관리자는 다음과 같은 기능을 제공한다.

- ◆ 사용자가 설정한 QoS 파라메터에 대한 사용자 프로파일을 제공함으로써, QoS 설정의 반복을 피하도록 하며, 최종 사용자의 선택은 자동적으로 시스템 파라메터로 사상되게 한다.
- ◆ 사용자가 원하는 데로 화질, 화면 크기, 화색 선명도, 성능, 동기화, 가격 등과 같은 응용 서비스의 질에 대한 QoS 요구를 설정한다.
- ◆ 사용자 QoS 파라메터를 서비스를 제공하는 시스템 요소나 계층에 대한 QoS 파라메터로 사상한다. 파라메터의 성격에 따라 네트워크 대역 폭 조정, 비트 에러율, 지터, 평균 대기 시간, 동기화, 데이터 전송 비등과 같이 시스템을 구성하는 각각의 요소에서 담당하는 파라메터로 사상된다.
- ◆ QoS 파라메터들이 각각의 시스템 요소들 간에 일-대-일 관계로 연결되지 않고 상충될 경우에 대한 조절 기능을 제공한다.
- ◆ QoS 파라메터 값을 모니터링하고 유지 관리해 주는 기능을 제공한다.

3.5 응용 프로그래밍 인터페이스(API)

DMCF를 이용하여 응용 프로그램을 개발하는 개발자는 인터페이스만을 참조하여 개발하기 때문에 DMCF의 실제 구현에는 영향을 받지 않는다. 이 인터페이스는 DSM-CC의 User-to-User 인터페이스, MetaDB 인터페이스, QoS 인터페이스로 구성되며 저장소 인터페이스는 응용 프로그램 개발자에게 제공되지 않는다.

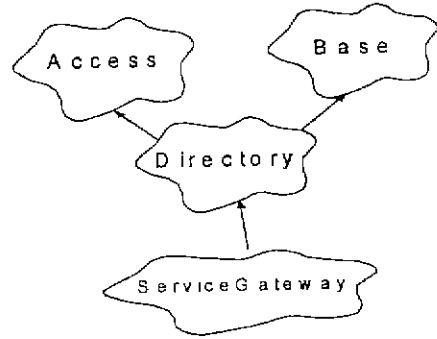


그림 8 ServiceGateway 상속 관계

3.5.1 DSM-CC의 User-to-User 인터페이스

- ◆ Base 인터페이스 - 이 인터페이스는 모든 객체에 있어서 공통적인 동작 즉, open, close, destroy, isA를 제공하는 추상 인터페이스이다. 따라서, 직접적으로 객체화 될 수 없으며 다른 객체에 대한 부모 역할을 한다.
- ◆ Access 인터페이스 - 크기, 히스토리(version과 date), 잠금(lock) 상태, 사용허락과 같은 공동으로 사용하는 속성을 표현한 추상 인터페이스이다.
- ◆ Directory 인터페이스 - 이 인터페이스는 이름, 서비스 및 데이터를 결합하기 위한 포괄적인 명명 공간(name space)을 제공한다. 그 구성은 Access 인터페이스와 CORBA의 객체 서비스 Naming 모듈에 정의된 NamingContext로부터 상속을 받고 기능을 추가한 형태이다. 그러나 그 실제의 구현은 ServiceGateway에서 이루어진다.
- ◆ Stream 인터페이스 - 연속적인 MPEG-2 스트림을 조작하기 위한 VCR 기능(일시 정지, 재개, 고속 전 후진, 오디오 속도 제어)을 모사하기 위해 사용하는 인터페이스이다. 클라이언트는 이 인터페이스를 호출하여 MPEG-2 스트림을 대화 식으로 제어한다. 이 인터페이스는 저장소 관리자의 기능으로 구현된다.
- ◆ File 인터페이스 - 파일에 대한 읽기와 쓰기 동작을 기술하고 있다. 이 인터페이스는 다양하고 이질적인 객체와 파일 시스템에서도 서버를 쉽게 구현하기 위한 방편으로 제공된다.

- ◆ ServiceGateway 인터페이스 - 이 인터페이스는 응용 서비스에 대한 정보를 가진 Directory를 이용하여 클라이언트가 한 서비스 영역에 접속할 수 있게 하는 동작(attach, detach, suspend, resume)을 기술하고 있다.

그림 8은 Directory로부터 속성과 동작을 상속받는다라는 것을 보여준다. 따라서 클라이언트가 ServiceGateway 인터페이스를 호출함으로써 접속하고자 하는 서비스를 찾을 수 있다. 이 인터페이스가 호출되면 내부적으로 세션을 확립하는 과정이 일어난다. 이 과정은 DSM-CC의 User-to-Network 부분에 프로토콜이 정의되어 있다.

- ◆ SessionGateway 인터페이스 - 이 인터페이스는 DSM-CC의 U-N 프로토콜을 구현한 것으로 SRM과 협상하여 세션을 관리하고 자원을 관리하는 기능을 정의하고 있다. 그리고 U-U 호출과 U-N 호출사이의 자원 협상을 위한 메시지의 변환을 담당한다.
- ◆ Service 인터페이스 - 응용 서비스를 제공하는 서버가 물리적으로 분산된 환경에서 ServiceGateway가 서버에 접속할 수 있게 도와주는 기능을 제공한다.
- ◆ Interface 인터페이스 - 서버 안의 객체 인터페이스는 일관성이 있고 서로 충돌이 일어나지 않아야 한다. 그러나 새로운 인터페이스가 필요하여 이것을 시스템에 설치할 때, 시스템 안에 이미 존재하고 있는 인터페이스와 충돌이 일어나는지를 검

사하고 인터페이스에 관한 정보를 관리한다.

- ◆ Life-Cycle 인터페이스 - 객체의 생성을 위해 사용되며, DSM-CC 환경에서 객체에 대한 참조자(reference)가 유일하도록 보장한다.
- ◆ Security 인터페이스 - 클라이언트가 인증에 필요한 정보를 서버에게 제공할 수 있게 한다.
- ◆ View 인터페이스 - 서버 안의 응용 서비스들에 관한 정보가 어떤 형식으로 저장되어 있는지 상관없이 사용자가 원하는 형식으로 정보를 보여주는 기능에 대한 인터페이스이다. 이는 Directory 인터페이스를 이용하여 구현한다.
- ◆ Event 인터페이스 - 클라이언트가 비동기적인 사건을 신청하거나 해제할 수 있게 해 주는 인터페이스이다.
- ◆ Composite 인터페이스 - 자식 객체의 집합을 하나의 부모아래에 결합한 복합 객체를 관리하는 인터페이스를 제공한다.

3.5.2 MetaDB 및 QoS 인터페이스

- ◆ MetaDB 인터페이스 - 데이터베이스 시스템의 종류에 상관없이 사용자에게 투명한 접근을 허용하는 데이터베이스 접근에 관한 인터페이스를 제공한다.
- ◆ QoS 인터페이스 - QoS 파라미터 값을 설정, 수정, 조회하는 기능에 대한 인터페이스를 제공한다. 그리고 응용 서비스가 물리적으로 분산되어 구현될 때 QoS 교섭을 지원하는 인터페이스를 제공한다.

4. 응용 프로그램 개발 예

응용 프로그램은 클라이언트와 서버 부분으로 나누어져 개발된다. 제안한 DMCF를 이용하여 설계한 VOD 시스템(그림 9)은 서비스를 제공하는 서버와 비디오를 보여주는 클라이언트로 구분되며, 클라이언트는 PC, 서버는 UNIX를 탑재한 주전산기를 이용하고 서버의 기능은 분산된 모습을 가지도록 하였다. 이 VOD 시스템은 WWW 기술과 연동이 되어 동

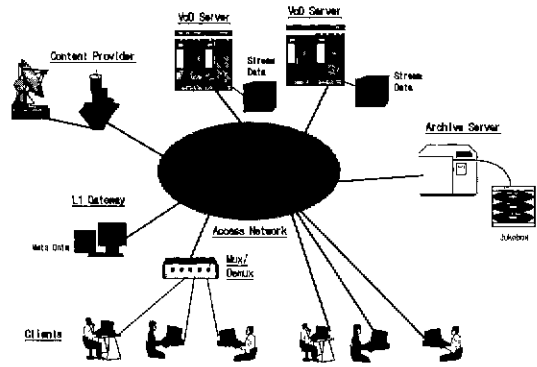


그림 9 VOD 시스템 구성도

작하도록 설계되었다. 즉 VOD를 위한 메타정보 처리는 HTML 문서 형식으로 표현이 되고, 비디오 스트림은 HTML 문서 안에 내포된 객체로 표현이 된다.

클라이언트를 위해 DMCF가 제공하는 기능의 핵심은 메타정보에 대한 인터페이스 제공과 비디오 스트림에 대한 제어이다. ServiceGateway로의 접근은 WWW을 이용하여 L1 Gateway에 접속하면, L1 Gateway가 ServiceGateway 인터페이스를 이용하여 제공되는 서비스의 종류(VOD, Teleshopping, etc.)를 판별하고 관련된 정보를 클라이언트에게 제공한다. 클라이언트는 HTML 문서로 제공된 정보를 이용하여 필요한 메타 정보(예: 비디오 타이틀)를 MetaDB 인터페이스를 이용하여 획득할 수 있다. 비디오 신호를 제어하는 것은 Stream 인터페이스를 통하여 직접 서버에 구현된 비디오 신호 제어 기능을 호출함으로써 이루어진다.

서버는 기능적으로 몇 개의 물리적으로 분산된 시스템의 집합으로 구현이 된다. L1 Gateway에는 DSM-CC의 User-to-Network 부분과 메타 정보를 관리하는 브로커가 존재하여 클라이언트에 대한 관문 역할을 한다. L1 Gateway를 접근하는 방법으로 WWW 환경의 표준 프로토콜인 HTTP를 이용하였다. 그리고 VOD를 서비스하는 그림 9의 VOD Server는 연속적인 스트림에 대한 제어를 실제로 구현하고 효율적으로 스트림을 송출하기 위해 DMCF의 저장소 기능을 이용한다. 그리고 방대한 양의 비디오 데이터를 관리하는 Archive

서버를 따로 두어 저장의 효율성을 높였다. 이 모든 분산된 서버의 기능은 클라이언트에게는 L1 Gateway를 통하여 투명하게 하나의 서버 처럼 보인다.

5. 결 론

멀티미디어용 서비스의 개발은 매우 복잡한 시스템 환경을 요구한다. 이러한 복잡성을 객체지향 프레임워크 기법을 이용하여 쉽고 확장 가능하게 바꿀 수 있다.

본 논문에서는 DMCF 프레임워크를 이루는 기반으로 CORBA 구조를 사용하고 프레임워크 자체를 위해 DSM-CC 모델을 채용하였다. 그리고 효율적인 멀티미디어 정보 검색을 위해 저장소 관리, MetaDB 관리, QoS 관리를 위한 기능을 추가하였다.

응용 프로그램 개발자는 프레임워크가 제공하는 API를 이용하여 쉽게 응용 프로그램을 구현할 수 있다. 그리고 프레임워크는 객체 지향 기법으로 설계되어 쉽게 기능을 확장, 추가, 변경이 가능하여 프로그램 개발자에게 유연성을 제공한다.

참고문헌

[1] Grady Booch, Object-Oriented Analysis and Design with Applications 2nd Ed., Addison-Wesley Publishing Company, 1994.

[2] Robert Orfali, Dan Harkey, Jeri Edwards, The Essential Distributed Objects Survival Guide, John Wiley & Sons, Inc., 1996.

[3] Thomas J. Mowbray, Ron Zahavi, The Essential CORBA : Systems Integration Using Distributed Objects, John Wiley & Sons, Inc., 1995.

[4] Randy Otte, Paul Patrick and Mark Roy, Understanding CORBA, Printice-Hall, Inc., 1996.

[5] Douglas E. Comer, The Internet Book, Printice-Hall, Inc., 1995.

[6] Zhigang Chen, See-Mong Tan, Roy H. Campbell, Yongcheng Li, "Real-Time

Video and Audio in the World Wide Web," Proceedings of 4th International World Wide Web conference (http://www.w3.org/pub/WWW/Journal/1/stan.211/paper/211.html), Dec. 1995.

[7] Digital Audio-Visual Council, DAVIC 1.0 Specifications, Digital Audio-Visual Council, 1995-1996.

[8] ISO/IEC, Extension for Digital Storage Media Command and Control, ISO/IEC 13818-6, 1995. 12.

[9] A. Vogel, B. Kerherve, G. von Bochmann, J. Gecsei, "Distributed Multimedia and QoS : A Survey," IEEE Multimedia, vol. 2, No. 2, pp.10-19, 1995.

[10] Ted Lewis and et.al. Objected-Oriented Application Frameworks, Manning, 1995

[11] Sean Cotter, Mike Potel, Inside Talgent Technology, Addison-Wesley Publishing Company, 1995.

[12] John December, Mark Grinsburg, CGI & HTML, Sams.net Publishing, 1995.

[13] Banu Ozden, Rajeev Rastogi, Avi Silberschatz, "On the design of a low-cost video-on-demand storage system," Multimedia System, Springer-Verlag 1996.

[14] Craig Frederighi and Lawrence A. Rowe, "A Distributed Hierarchical Storage Manager for a Video-on-Demand System," Proceedings of Symposium on Electronic Science & Technology, Feb. 1994.



이 범 식

1988 경북대학교 전자공학과 (공학사)
 1993 연세대학교 전산학과 (공학석사)
 1993~현재 한국전자통신연구소 컴퓨터연구단 연구원
 관심분야: 병렬 및 분산처리, 프로그래밍 환경

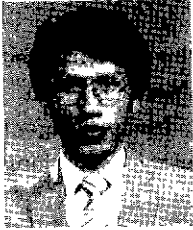
홍 철 의



1985 한양대학교 공과대학(공학사)
 1989 New Jersey Institute of Technology, Computer Science and Information (공학석사)
 1992 University of Missouri, Computer Science(공학박사)
 1992~현재 한국전자통신연구소 컴퓨터연구단 선임연구원

관심분야: 최적화 이론, 병렬 및 분산처리, 프로그래밍 환경

조 기 환



1985 전남대학교 계산통계학과 (전산학 전공)졸업
 1987 서울대학교 계산통계학과 (전산학 전공)석사
 1996 University of Newcastle (영국) 전산학과 박사
 1987~현재 한국전자통신연구소 컴퓨터연구단 선임연구원

관심분야: 이동 컴퓨팅, 분산 시스템, 컴퓨터 통신, 멀티미디어 서비스

임 기 욱



1977 인하대학교 공과대학 전자공학과 졸업
 1986 한양대학교 대학원 전자계산학 석사
 1994 인하대학교 대학원 전자계산학 박사
 1977~83 한국전자기술연구소 선임연구원
 1986~88 한국전자통신연구소 시스템소프트웨어연구실장

1988~89 미 캘리포니아주립대학(Irvine) 방문연구원
 1989~현재 한국전자통신연구소 시스템연구부장
 관심분야: 소프트웨어 아키텍처, 실시간 데이터베이스시스템, 컴퓨터시스템구조

● APSEC '96 ●

- 일 자 : 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