

CORBA 기반 화상회의 시스템의 설계 및 구현

김만수^o, 정목동
부경대학교 컴퓨터공학과

Design and Implementation of a CORBA-based Video Conferencing System

Mansoo Kim, Mokdong Chung
Dept. of Computer Engineering, Pukyong National University

요 약

분산 컴퓨팅 기술의 발전과 네트워크의 고성능화는 대용량의 멀티미디어 데이터를 송·수신 하여 처리 할 수 있게 되었다. 그러나 실시간 및 이기종간의 데이터 전송에는 네트워크 전송 지연과 프로그램 구현의 어려움 등과 같이 극복해야 하는 문제점들이 많이 있다. 이 문제들을 해결하기 위하여 본 논문에서는 객체지향형 분산처리 환경 하에서 OMG가 제시한 멀티미디어 서비스 설계를 확장하는 방안을 제시한다. 이를 위해서 CORBA와 JMF Java Media API 기반으로 Smart Explorer라고 불리는 효율적인 실시간 오디오/비디오 멀티미디어 소프트웨어 기반 구조를 제시하고 있다. 제어 데이터와 미디어 데이터의 전송 경로를 분리하고 효율적인 실시간 오디오/비디오 전송을 위해서 RTP/RTCP 프로토콜을 사용한다. 또한 제안하는 소프트웨어 기반 구조를 이용하여 오디오/비디오 스트림 시스템을 분산 객체 환경에 맞게 구현한 결과도 제시한다. 향후 이 시스템은 방송, 원격 교육, 화상회의 등 실시간 통신 소프트웨어에 응용 될 수 있을 것으로 기대된다.

1. 서론

최근 고속 네트워크와 멀티미디어 컴퓨터 기술의 발전과 더불어 일반 사용자들의 컴퓨팅 환경 또한 멀티미디어 서비스를 충분히 처리할 수 있는 수준으로 발전 하였다. 이러한 배경을 바탕으로 최근에는 화상회의나 주문형 비디오 시스템(Video On Demand : VOD), 인터넷 폰 등 새로운 형태의 분산 멀티미디어 응용에 대한 연구가 활발히 진행되고 있다. 그러나 멀티미디어 서비스 기술의 미래에 대한 기대

와는 달리 VoD, Voice over IP[1] 및 오디오/비디오 폰(Audio/Video Phone)과 같은 실시간을 요구하는 멀티미디어 응용은 이질적인 네트워크 환경이나 과중한 정보 전송으로 인한 네트워크 지연 등 아직 해결해야 할 많은 문제점들이 있으며 이러한 멀티미디어 서비스에서 발생하는 문제점들을 극복하기 위해 많은 대안들이 연구되고 있다.[2,3,4]

소프트웨어 업계를 중심으로, 객체 지향형 소프트웨어 모델링 기술을 분산처리 환경으로까지 확장하기 위한 연구가 계속되어, 이와 관련된 표준 사양

들이 이미 정의되었거나 검토 중에 있다. 그 중에서도 OMG(Object Management Group : OMG)에서 분산 컴퓨팅과 객체 지향 기술을 하나로 합친 표준 아키텍처로 제안된 CORBA(Common Object Request Broker Architecture : CORBA)[5]는 분산 시스템의 상호 운용성 문제를 해결하며, 객체 지향적인 방법을 제공함으로써 표준 개방 분산 환경의 플랫폼으로 최근 주목을 받고 있다.

그러나, 분산 객체 미들웨어가 제공하는 장점에도 불구하고 현재 대부분의 구현된 CORBA 프로그램은 성능의 최적화, 실시간 기능과 QoS(Quality of Service : QoS) 기능의 결여로 분산 멀티미디어 응용에 효율적으로 적용되기 어렵다. OMG에서도 이러한 점을 인식하고 현재 오디오/비디오 스트림 처리에 관한 RFP(Request For Proposal)를 제시하고 있다.[6,7]

본 논문에서는 위와 같은 문제점들을 해결하기 위해 분산 객체 미들웨어인 CORBA 환경을 확장하여 실시간 오디오/비디오 스트림의 전송과 제어를 효율적으로 하기 위한 소프트웨어 기반 구조를 설계하고, SUN사에서 제시한 다양한 환경의 플랫폼에서 구동 가능한 멀티미디어 어플리케이션 개발 Java Media API 인 JMF(Java Media Framework : JMF) [13]을 사용하여 이를 응용한 오디오/비디오 스트림 시스템 Smile Phone을 구현하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구 및 사례 등을 살펴보고 문제점들을 언급한 후, 본 논문의 배경을 간략히 소개한다. 3장에서는 Smile Phone의 기본 요건들과 이에 근거하여 설계된 시스템의 전체구성 및 세부 사항들에 대하여 기술하며, 4장에서는 3장에서 언급된 설계에 기초하여 시스템을 구현하고, 이를 시험한 내용들을 설명한다. 마지막으로 5장에서는 결론과 향후 연구과제에 관하여 기술한다.

2. 관련연구

컴퓨터 시스템들이, 각종 오디오/비디오 장치를 이용하는 응용소프트웨어를 운영하게 됨에 따라, 컴퓨터 네트워크 상에서 오디오/비디오 데이터를 효율

적으로 처리하기 위한 연구들이 활발히 진행되고 있다. 또한 이질적인 네트워크 환경을 위한 분산 객체 미들웨어가 이미 주요 소프트웨어 업체들에 의해 개발되고 있으면, 대표적으로 IONA의 Orbix, Microsoft DCOM, Inprise의 VisiBroker 등과 같은 여러 분산처리 미들웨어들은 이질적인 분산 환경에서 위치의 투명성, 프로그래밍 언어의 투명성과 더불어 사용자에게 하드웨어와 소프트웨어 및 네트워크 환경의 일관성을 제공하게 됨에 따라 분산 환경의 상호 운용성 문제를 자연스럽게 해결하고 있다.[10, 11, 12]

CORBA는 분산 객체들간에 서비스를 제공하거나 서비스를 획득할 수 있도록 하기 위한 통신 환경이 정의된 표준 기술이다. 그러나 CORBA는 텍스트 데이터 처리를 중심으로 설계되어 있기 때문에 분산 환경에서 멀티미디어 응용을 위한 대용량의 데이터 전송에 따른 과도한 통신 프로토콜의 Overhead, 시스템 내부처리의 Overhead 및 실시간 서비스를 제공하지 못하는 문제점이 있다. 이러한 이유로 OMG에서는 멀티미디어 스트림을 효율적으로 처리하기 위한 부가적인 표준을 정의 하였다. 현재 OMG에 제출되어 검토 단계에 있는 "오디오/비디오 제어 및 관리 사양"[7]은 CORBA기술과 오디오/비디오 데이터에 대한 처리 기술을 융합하기 위한 것으로, 오디오/비디오 데이터 처리와 관련된 소프트웨어 구성 요소들을 CORBA객체로 모델링하고, 이 CORBA 객체들간의 오디오/비디오 데이터 전송경로와 응용프로그램을 구성하는 객체들과의 제어정보 교환 경로를 분리하는 방안을 제시하고 있다. 그러나 분산응용프로그램을 작성하기에는 다소 복잡한 API를 정의하고 있으며, 아직 표준화 및 상용화 단계에는 이르지 못한 상태이다.

그림 1은 OMG의 오디오/비디오 스트림 서비스의 구조와 컴포넌트들을 보여준다. IDL 인터페이스들은 예를 들면, 스트림을 설정하고, 다운 받고, 전송을 시작하고, 정지하는 등의 스트림을 제어하고 관리하기 위해 사용되어 진다. 스트림은 연결을 위한 전송과 분리되어 운반되기 때문에 ORB 외부로 생각할 수 있다. 이것은 전화 통신에서 시그널링과 음성 채널

이 분리되는 것과 어느 정도 유사하다.

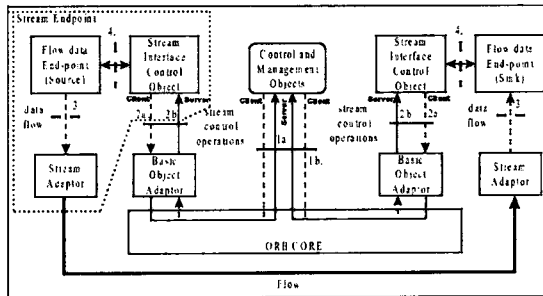


그림 1 OMG에서 Stream Service를 위해 제시한 시스템 구성도

본 논문에서 제시하는 Smile Phone은 분산처리 환경에서 오디오/비디오 데이터를 처리하는 데 필요한 요소들을 수용하기 위한 기반구조를 설계 및 구현한 것으로, 응용시스템의 운영환경에 따라 소프트웨어 구성 요소들을 추가 혹은 제거할 수 있는 유연한 구조를 가지고 있다.

기존의 오디오/비디오 스트림 시스템들은 TCP/IP 및 UDP/IP를 이용한 일반적인 네트워크 프로그래밍을 통해 서버 또는 클라이언트간의 통신을 하게 된다. 이런 경우 이질적인 분산 환경에서 상호 연동성이 어려워지고, 사용하는 프로토콜이 복잡해질수록 소프트웨어의 유지보수 비용이 증가하게 된다. 그렇지만, CORBA를 통해 구현된 오디오/비디오 스트림 시스템인 Smile Phone은 IDL(Interface Definition Language : IDL)를 통해 객체들의 메소드들 규정함으로써 운영 시스템 관계 없도록 상호 연동성 문제를 해결하였고, Java로 구현하여 기존 화상 전화 시스템의 단점인 플랫폼 종속성을 제거하였다. 또한 객체지향 설계 및 구현으로 프로그램의 복잡성이 현저히 줄어 소프트웨어의 수정 및 기능 추가 등 유지보수 비용을 최소화 하였다.

3. 시스템 설계

오디오/비디오 데이터 처리를 위한 Smile Phone은 분산처리 환경과 함께 미들웨어를 형성하는 부분으로서 미들웨어가 요구하는 요건들을 만족하여야 한다. 즉, 응용 프로그램들에게 잘 정의된 API를 제공함과 동시에, 하부 운영체제가 제공하는 시스템 자원들을 효율적으로 이용하도록 설계되어야 한다.

3.1 설계 목표

본 연구는 다음과 같은 설계 목표를 가진다.

- 상호운용성(interoperability) 제공
- 제어 데이터와 스트림 데이터의 전송 분리
- 플랫폼 독립성 제공
- 다양한 미디어 형식 지원
- 객체지향설계

3.2 전체 시스템 구조

본 논문에서는 OMG에서 제시한 CORBA 기반 멀티미디어 스트림 서비스 설계를 기초로 오디오/비디오 데이터를 처리하기 위하여 일반적인 스트림 제어에 관한 분산 처리 환경의 소프트웨어 설계와 클라이언트 프레임워크인 SmartExplorer를 설계하고, 이를 응용한 오디오/비디오 스트림 시스템 Smile Phone의 구현이 포함되어 있다. 그림 2는 Smile Phone의 분산처리 시스템 구조를 보여 주고 있다. 그림 2에서와 같이 여러 멀티미디어 제어 및 관리는 IDL로 정의된 분산 객체들에 의해 처리되어 ORB에서 통신하고, 오디오/비디오 멀티미디어 데이터는 JMF에서 제공하는 각 클라이언트의 디바이스 핸들러(Device Handler)에서 처리되어 RTP(Real-Time Transport Protocol : RTP)[8,9]로 전송 된다.

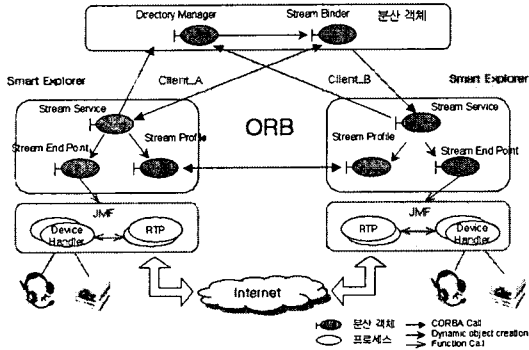


그림 2 Smile Phone 전체 시스템 구조

한정된 대역폭의 네트워크 환경은 실시간 멀티미디어 데이터 전송 시 데이터의 손실 및 지연이 발생할 수 있고, 이러한 손실 및 지연의 정보는 RTP 헤더 정보의 Serial Number, Jitter, RTT(Round-Trip Times : RTT)를 분석해 알 수 있다. 현재 위와 같은 문제점을 해결하기 위해 네트워크의 대역폭에 따른 손실을 최소화 하기 위해 오디오 정보의 우선 전송, 낮은 대역폭으로의 코덱 변환 등을 적용하고 있다.

3.3 Smile Phone 분산 객체들

그림 3은 Smile Phone의 오디오/비디오 멀티미디어 데이터 전송을 위한 분산 객체 관계의 설정을 나타낸다. 본 시스템에서는 DirectoryManager, StreamBinder, StreamService, StreamEndPoint, StreamProfile 분산 객체들이 오디오/비디오 데이터 관리 및 전송 제어를 담당한다. DirectoryManager와 StreamBinder는 분산 시스템에서 클라이언트로부터 연결을 위해 대기한다. StreamService, StreamEndPoint, StreamProfile은 각 클라이언트에서 생성되며 StreamBinder를 통해 클라이언트 상호간에 객체 참조를 하게 된다.

DirectoryManager

DirectoryManager 객체는 등록된 StreamService 객체로부터 클라이언트 연결 정보를 DB로 저장하고,

연결을 희망하는 클라이언트의 요구에 의해 등록된 다른 StreamService 리스트 정보를 제공하게 된다. 그리고 연결을 위해 StreamBinder 객체를 생성한다.

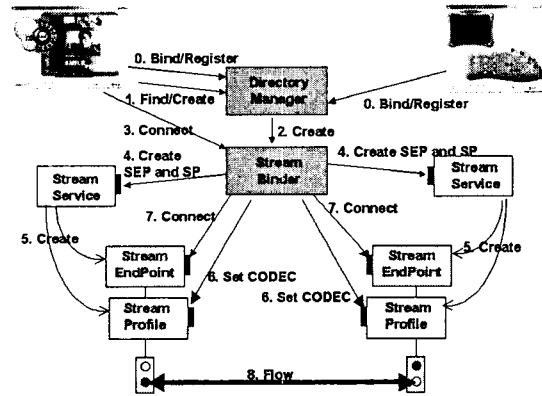


그림 3 Smile Phone의 스트림 전송을 위한 분산 객체 관계

StreamBinder

StreamBinder는 연결을 희망하는 두 클라이언트의 StreamService 객체의 연결을 관리한다.

StreamService

StreamService 객체는 StreamEndPoint, StreamProfile 객체를 생성한다.

StreamEndPoint

StreamEndPoint는 멀티미디어 데이터 전송 제어를 관리하게 된다. StreamEndPoint는 오디오/비디오 데이터 처리를 위한 AudioManager, VideoManager 객체를 생성하여 실시간 데이터 전송을 담당하게 하고, AudioManager, VideoManager 객체의 RTP 정보를 얻어 네트워크 상태 및 시스템 상태 등을 주기적으로 확인함으로써 원활한 전송을 보장하게 한다.

StreamProfile

StreamProfile 객체는 Stream Service 객체가 지원하고자 하는 미디어의 형식을 정의하고 있으며, Stream Binder에 의해 Stream Service 객체의 연결이 이

루어지는 시점에서 지원하는 미디어에 대한 상호 지원 여부를 확인하게 된다.

3.4 클라이언트 구조

그림 4는 통신 모듈인 Smart Explorer의 내부 구조이다. Smile Phone 응용 프로그램은 스트림 서비스 통신 모듈인 SmartExplorer의 API를 사용하여 연결 및 데이터 송,수신을 하게 된다. SmartExplorer는 StreamService, StreamEndPoint, StreamProfile 분산 객체들을 생성, 다른 클라이언트와 연결이 되면 오디오/비디오 데이터 송,수신을 위해 VideoManager와 AudioManager를 생성한다. 효율적인 스트림 전송을 위해 비디오 코덱은 JPEG을 사용하고, 오디오 코덱은 G.723를 사용한다.

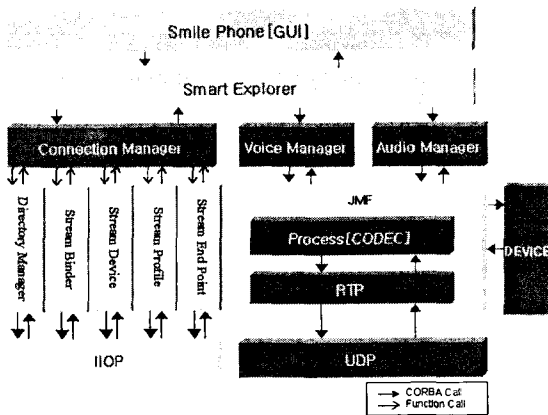


그림 4 클라이언트측 통신 모듈 구조

4. 구현 및 테스트

Smile Phone은 화상 카메라와 마이크를 장착한 Windows 95, 98/NT/2000를 운영체제로 사용하는 펜티엄(Pentium)급 PC들을 대상으로 구현되었다. 분산객체 환경을 구성하기 위해서 서버에 Inprise사의 VisiBroker 4.1 버전을 설치 하여 Stream Service CORBA객체를 구현 하였다. 클라이언트는 Windows 98 환경에서 JDK 1.3을 지원하는 J-Builder 4.0와 오디오/비디오 처리를 위해 JMF를 이용하여 구현하였다.

오디오 처리를 위해 JMF를 이용하여 구현하였다.

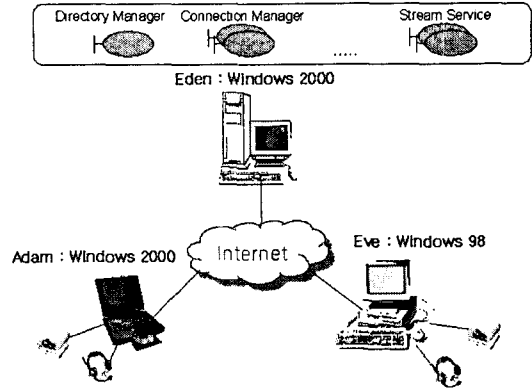


그림 5 시스템 테스트 환경

통화 연결이 이루어지면 네트워크 상태를 실시간으로 표시하는 그림 5처럼 화면이 보여진다.



그림 6 통화 화면

그림 6에서 보는 것과 같이 상대방주소 즉, 연결하고자 하는 다른 클라이언트의 StreamService 객체가 DirectoryManager에 등록된 상대방 IP를 입력하여 연결 버튼을 누르면, 3장의 설명과 같이 DirectoryManager에서 입력한 IP에 맞는 StreamService 객체를 검색하여 오디오/비디오 스트림 제어를 위한 StreamEndPoint 객체를 생성하고, 두 클라이언트간에 PC-to-PC 연결하여 오디오/비디오 데이터를 전송하게

한다. 이렇게 구현 함으로써 연결 및 오디오/비디오 스트림 제어 및 관리와 오디오/비디오 스트림 전송을 분리하였고, CORBA와 Java를 기반으로 함으로써 플랫폼에 상관없이 동작하게 된다.

그림 7은 일반적인 LAN을 통한 인터넷 접속을 하여 두 클라이언트의 연결시 발생하는 오디오 데이터의 손실과 56Kbps 모뎀 PPP 접속을 하여 발생하는 오디오 데이터의 손실을 그래프로 나타낸 것이다. 이 그래프는 56Kbps 모뎀 PPP 접속시 더 많은 오디오 데이터의 손실을 있음을 보여 주고 있다. 이러한 오디오 데이터의 손실을 최소화 하기 위해 비디오 데이터 보다 오디오 데이터의 우선 전송이나 오디오 코덱의 실시간 변경에 대해서는 지속적인 연구가 필요하다.

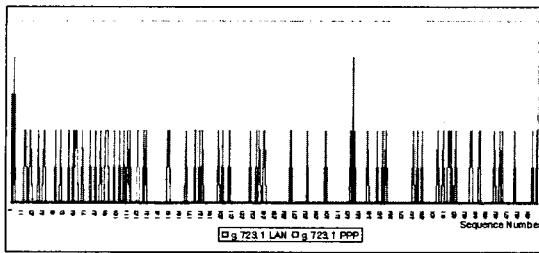


그림 7 네트워크 대역폭에 따른 오디오 데이터 손실

일반적으로 RTP를 사용하지 않은 CORBA기반 멀티미디어 스트림 시스템들은 비록 분산 객체들의 통신만으로 좀더 직관적인 API를 제공할 수 있는 장점이 있지만, 대용량의 실시간 데이터를 전송하는 것에는 아직 네트워크 인프라가 구축이 되어 있지 못한 실정이다. Smile Phone은 이러한 점에서는 효율적인 데이터 전송을 보장하지만, 오디오/비디오 코덱 및 전송을 위해 JMF로 구현 함으로써 사용자가 본 시스템을 사용하기 위해서는 JMF를 설치하는 번거로움이 있으며, 아직 Java의 처리 속도에 따른 높은 시스템 사양이 요구되는 단점이 있다.

5. 결론 및 향후 과제

본 논문에서는 텍스트 기반 CORBA의 실시간 멀티미디어 데이터 처리에 따른 문제점을 해결하고, 효율적인 스트림 제어, 관리 및 전송을 위해 객체지향형 분산 프로그래밍 환경 하에서 OMG의 오디오/비디오 스트림 서비스 설계를 기반으로 한 오디오/비디오 등과 같은 스트림 형식의 실시간 데이터를 체계적으로 처리하기 위해 소프트웨어 기반 구조를 설계하고, 이를 응용한 오디오/비디오 스트림 시스템인 Smile Phone을 구현하였다. 본 논문에서는 분산 컴퓨팅 환경을 효과적으로 지원하기 위해 CORBA를 선택하였고 또한 연속 미디어인 멀티미디어 데이터의 특성을 살리며, 효과적으로 송,수신하기 위해 RTP를 이용하여 시스템을 설계 및 구현 하였다.

본 논문에서 소개한 이러한 소프트웨어 기반 설계를 바탕으로 향후에는 이질적인 분산 환경에서 효율적이고 융통성이 있는 다양한 분산 멀티미디어 응용 프로그램 개발을 위해 QoS 및 자원 관리에 대한 연구와 실험을 하고, 인터넷상에서 방송, 교육, 화상회의 등과 같은 실시간 정보 표현 분야에 응용할 예정이다.

[참고문헌]

- [1] D. Minoli and E. Minoli. "Delivering Voice Over IP Networks". John Wiley, 1998
- [2] Guojun Lu, "Communication and Computing for Distributed Multimedia Systems", Artech House, Boston, London, 1996
- [3] T.H. Yun, J. Y. Kong and J. Won-Ki Hong, "A CORBA-based Distributed Multimedia System", Proc. of 1997 Pacific Workshop on Distributed Multimedia Systems, pp. 1-8, 1997
- [4] 이성환, "CORBA에 기반한 DSM-CC 구현", KRNAT '97, pp.505-517, 1997
- [5] OMG, "The Common Object Request Broker: Architecture and Specification Revision 2.4", OMG Document formal, October 2000
- [6] OMG, "Control and Management of A/V Streams Request For Proposal", OMG TC Document, 199
- [7] OMG, "Audio/Video Streams v1.0", OMG Document

- formal, 2000
- [8] RFC 1889 RTP: A Transport Protocol for Real-Time Applications, January 1996
 - [9] RFC 1890 RTCP : "RTP Profile for Audio and Video Confernces with Minimal Control", January 1996
 - [10] Jon Siegel, "CORBA Fundamental and Programming", John Wiley & Sons, Inc. , 1996
 - [11] IONA Technologies PLC, "Orbix Programmer's Guide", IONA Technology Ltd., 1997
 - [12] Inprise Corporation, "Programmer's Guide". VisiBroker for Java, 2000
 - [13] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison-Wesley, 1998
 - [14] Inc. Sun Soft. "Java Media Framework", <http://java.sun.com/products/javamedia/jmf/>, 1998