

# Non-CORBA 응용을 통합하는 분산 멀티미디어 플랫폼의 설계 및 구현

° 김 중현, 정 형석, 정 광수  
광운대학교 전자공학부 컴퓨터통신연구소

## Design and Implementation of A Distributed Multimedia Platform Integrating Non-CORBA Applications

° Jonghyun Kim, Hyungseok Chung, Kwangsue Chung  
School of Electronic Engineering, Kwangwoon University

### 요 약

본 논문에서는 CORBA 분산처리 환경을 갖추지 않은 운영체제 상에서 동작하는 Non-CORBA 응용들이 CORBA 분산환경의 객체 서비스를 이용 가능하도록 하는 통신 플랫폼을 설계하고 구현하였다. 현재 CORBA 표준에서는 서로 다른 ORB(Object Request Broker)들간의 상호운용(Interoperability)을 위하여 GIOP/IOP (General/Internet Inter-ORB Protocol) 프로토콜을 정의하고 있다. 구현된 분산 멀티미디어 플랫폼은 IOP Engine API를 제공함으로써 Non-CORBA 응용들을 통합하였으며, 디렉토리, 스트림, 파일 서비스 인터페이스를 제공함으로써 분산환경에서 멀티미디어 정보 서비스를 이용할 수 있는 환경을 제공한다

## 1. 서 론

CORBA 버전 2.0 스펙에서는 서로다른 ORB들간 상호운용을 지원하기 위하여 IOP 프로토콜을 정의하고 있다.

상호연동이란 CORBA 스펙을 따르는 서로 다른 ORB들간에 공통의 메시지 형식과 통신 프로토콜을 정의하여 상호 객체 참조 및 객체 서비스를 투명하게 교환할 수 있도록 한 것이므로 클라이언트는 객체 서비스의 위치에 투명하게 서비스를 제공받을 수 있다.

본 논문에서는 상호운용을 위해 정의된 IOP Engine을 이용함으로써 Non-CORBA 응용들이 CORBA 분산환경과 통합될 수 있도록 하는 분산 멀티미디어 플랫폼의 설계 및 구현을 다룬다. 일반적으로 Realtime Operating System을 이용하는 Set-Top Box와 같은 시스템상에서는 CORBA 분산처리 환경을 탑재하는 것이 상당한 오버헤드로 적용하는데, 이러한 경우 IOP Engine을 이용함으로써 오버헤드 문제를 해결 할 수 있다.

멀티미디어 분산플랫폼은 디렉토리, 스트림, 파일 인터페이스 등으로 구성되며, 각 인터페이스는 OMG IDL(Interface Definition Language) 언어로 기술된다. 이런 인터페이스들은 ISO/IEC JTC1/SC29/WG11 MPEG 그룹에서 정의한 DSM-CC(Digital Storage Media Command and Control) 표준을 근간으로 정의하였다.

DSM-CC는 다양한 통신망 환경하에서 대화 형식의 비디오 응용

을 지원하기 위해 정의된 프로토콜로서, 통신망을 통해 MPEG-I과 MPEG-II 등의 비디오 스트림을 관리하고 제어하기 위한 표준 방식을 기술하고 있다.

## 2. 상호연동의 개요 및 IOP 엔진의 구조

### 2.1 상호연동의 개요

ORB들간의 상호연동이란 CORBA 스펙을 따르는 ORB들간에 상호 객체 참조 및 서비스 제공을 가능하도록 하는 것을 의미한다. 상호연동의 구현을 위한 필수 요소들은 ORB들간의 상호연동을 위해 정의한 자료구조와 브릿징 메커니즘, 그리고 ORB들간의 공유 프로토콜 등이다. CORBA 2.0 스펙에서는 ORB간의 통신을 위한 프로토콜을 정의하고 있는데 이는 범용 목적의 GIOP 프로토콜과 TCP/IP를 지원하는 IOP의 형태로 구현되어 있다. 네트워크 환경이나 분산 환경이 특수하게 구축되어 있는 경우를 위한 프로토콜로는 ESIOP가 정의 되어 있으며, 구현 프로토콜로는 DCE-CIOP(DCE Common Inter-ORB Protocol)등이 있다. 그림 1에 이런 프로토콜들의 관계를 도시하였다.



그림 1. CORBA Interoperability Protocols

본 논문은 한국과학재단 특정기초연구(과제번호 97-01-00-12-01-5)의 일부 지원에 의한 연구결과임

GIOP(General Inter-ORB Protocol) 프로토콜은 ORB들간의 메시지 교환을 위하여 CDR(Common Data Representation)과 GIOP Message Format을 가진다. CDR은 ORB들간의 메시지 전달시에 IDL 자료형을 통신상에 적합한 자료 표현으로 변환하는 과정을 정의하고 있으며, GIOP Message Format은 자료의 교환과 에러 확인을 위하여 CORBA 버전 2.0 스펙의 GIOP 버전 1.0 스펙은 총 7가지의 Message Format을 정의하고 있으며, 개정된 GIOP 버전 1.1 스펙에서는 총 8개의 Message Format이 정의되어 있다.

ORB들은 자체적으로 고유한 객체 표현 방식을 사용한다 따라서 이렇게 서로 다른 ORB들 상에서 생성된 객체를 어떻게 표현화시킬 것인가 하는 것도 중요하다. 이를 위해서 CORBA 버전 2.0에서는 표준객체 참조자 양식으로 IOR(Interoperable Object Reference)을 정의하였다. IOR에는 객체 타입, 프로토콜, 이용 가능한 ORB 서비스, 객체 참조자의 널 값 유/무 등의 정보를 가지고 표현함으로써 원격지상의 ORB를 통해 해당 객체의 서비스를 호출하는 데 필요한 모든 정보가 포함된다.

### 2.2 IIOP Engine의 구조

GIOP 버전 1.1 스펙에 따라 작성되었으며 GIOP의 8개의 메시지를 처리할 수 있는 Request-level 브릿지 형태로 구현하였다

IIOP Engine은 크게 두 부분으로 나누어진다. 첫째 Runtime 부분이다. 이 부분에서는 OMG-IDL 데이터 타입을 CDR로 만들어 주는 Marshaling Engine과 TypeCode Interpreter가 있다 두 번째는 Bridge부분이다 이 부분에서는 IIOP 메시지를 Runtime 부분과 TCP 기반의 소켓으로 send, receive, dispatch 하는 교량 역할을 한다. 그림 2는 IIOP Engine의 세부적인 구성을 나타낸 것이다.

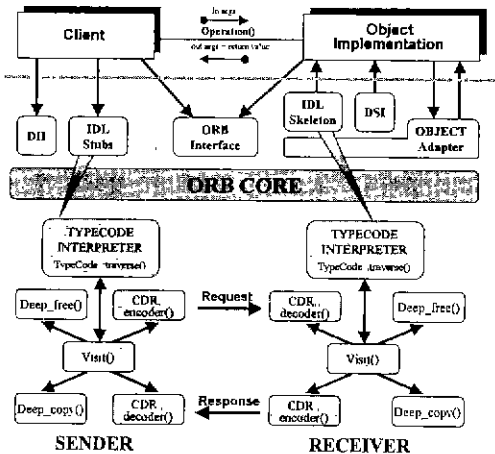


그림 2. IIOP Engine의 구성도

IIOP 엔진은 실행 시에 TypeCode Interpreter에서 객체의 파라메타를 encode 혹은 decode한다

#### 2.2.1 TypeCode traverse method

모든 데이터 구조를 marshaling / demarshaling 하고, visit

method로 보내진다.

#### 2.2.2 TypeCode visit method

visit method는 그림 2에서 보는 바와 같이 4개의 method중에서 현재 해석중인 하나의 method에 대한 주소를 가지는 포인터이다. CDR::encoder method는 자신의 호스트에서 사용하는 데이터형을 CDR로 변환하는 method이며, CDR::decoder method는 그 반대의 변환과정이다. deep\_copy method는 DII 메카니즘에서 메모리를 할당 파라메타를 marshaling 하여 TypeCode interpreter에서 CDR 스트림을 이용하도록 한다. deep\_free method는 서버의 DSI에서 입력된 데이터가 demarshaling되고 서버 응용으로 전달되기 전에 동적으로 할당된 메모리를 해제시킨다.

### 3. 분산 멀티미디어 통신 플랫폼의 구성

CORBA 분산환경에서 대화형 멀티미디어 통신 플랫폼을 구성하기 위한 기본 구성 중에서 원격지 서버의 비디오, 오디오, 그래픽, 텍스트 등의 미디어 데이터를 검색하고 수신할 수 있는 서비스는 다음과 같으며 멀티미디어 정보 서비스 응용 개발을 위해 필수적인 서비스이다.

#### 3.1 디렉토리 서비스

통신망을 통해 분산되어 있는 모든 서버 객체는 고유한 식별자를 갖고 있는데 이를 해당 객체의 객체 참조(Object Reference)라고 한다. 디렉토리 서비스는 클라이언트 객체가 서버 객체의 객체 참조를 획득하여 서비스를 이용할 수 있도록 해준다

#### 3.2 스트림 서비스

통신망을 통해 전달되는 비디오 스트림에 대한 재생, 정지, 일시정지, 고속 전, 후진등과 같은 VCR과 유사한 제어 기능을 제공한다.

#### 3.3 파일 서비스

클라이언트가 통신망을 통해 파일 서비스 객체에 접근하여 전체 또는 특정 부분의 파일 내용(content)을 얻을 수 있도록 한다. 전체적인 통신 플랫폼의 구성은 그림 3에 도시하였다.

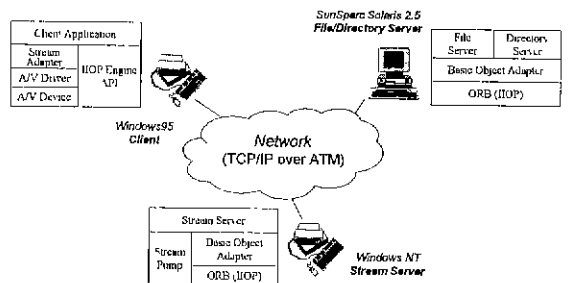


그림 3. 대화형 멀티미디어 통신 플랫폼의 구성

서버측에는 일반 CORBA 분산환경상에서 구축되었으며 클라이언트측에는 DII(Dynamic Invocation Interface) 방식을 이용하는 IIOP Engine API를 제공한다. 각 호스트는 155Mbps ATM 통신망을 통해 분산되어 운영된다. 서비스 객체들은 서버 형태로 존재하며 통신망을 통해 분산되어 운영된다. 파일 서버는 텍스트, 이미지 등과

같은 비연속 미디어 데이터의 저장, 관리, 전송 서비스를 제공하며, 디렉토리 서버는 객체참조에 대한 name resolution 서비스를 제공한다.

#### 4. 분산 플랫폼의 구현

정의된 인터페이스를 구현 언어로 매핑하고 이를 바탕으로 클라이언트와 서버를 구현하기 위해 IONA Orbix 2.1에서 제공하는 IDL 컴파일러 및 런타임 라이브러리를 이용하였다. 사용된 구현 언어는 C++이며, 실행코드를 생성하기 위해 Visual C++ 컴파일러 및 Sparc C++ 컴파일러가 사용되었다.

스트림 서버에서는 요청된 MPEG 비디오 스트림 객체에 대하여 GOP(Group of Picture) 단위로 스케줄링 되어 전송되며, 고속 진, 후진시 GOP 단위로서 처리가 된다.

클라이언트 응용은 Windows95에서 구현되었으며, 여기에 IOP Engine이 탑재된다. 서버는 WindowsNT 4.0과 SunSparc Solaris 2.5 상에서 구현되었다. 또한 이들은 ATM 통신망을 통해 상호 접속되어 있다. 전체적인 분산 플랫폼에서 동작하는 모습을 그림 4에서 도시하였다.

는 이를 통해 제어 메시지의 처리 여부를 결정하게 된다 예를 들면, 현재의 스트림 상태가 '전송' 중일 때 play나 resume 오퍼레이션이 기동되면 이를 무시하게 된다.

#### 5. 응용 예제

분산 멀티미디어 정보 서비스를 위해 본 논문에서 설계 구현된 플랫폼을 바탕으로 대화형 비디오 검색 서비스 응용을 구현하여 서비스 인터페이스들의 기능성을 검사하였다. 그림 5는 여러 분산 서버로부터 전송된 멀티미디어 데이터를 사용자에게 표현한 화면이다.

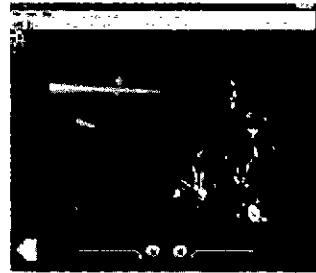


그림 5. 대화형 비디오 검색 서비스 응용

#### 6. 결론 및 향후 계획

본 논문에서는 대화형 비디오 서비스를 위한 CORBA 기반의 분산 플랫폼의 설계 및 구현을 다루었다 IOP Engine을 통하여 Non-CORBA 응용들도 CORBA 분산플랫폼상의 서비스를 제공할 수 있게 하였다. 구현된 스트림 인터페이스를 통하여 오디오, 비디오 스트림의 제어 및 관리 기능을 제공한다.

향후 과제로는 실시간적 특성을 고려한 스트림서버의 실시간 스케줄링 기법과 QoS 보장 메커니즘을 연구할 예정이다.

#### 참고문헌

- [1] OMG, *The Common Object Request Broker Architecture and Specification Revision 2.2*, OMG Inc., February 1998
- [2] OMG, *Control and Management of Audio/Video Streams RFP*, IONA Technologies Ltd. et al, 1997
- [3] ISO/IEC 13818-6 N1300, *Digital Storage Media Command and Control*, ISO/IEC JTC1/SC29/WG11, July 1996
- [4] SunSoft, "Inter-ORB Engine" <http://ftp.omg.org/pub/interop/iop.zip>, Feb, 1997
- [5] IONA, *Orbix2 Reference & Programming Guide*, Iona Technologies Ltd., 1996

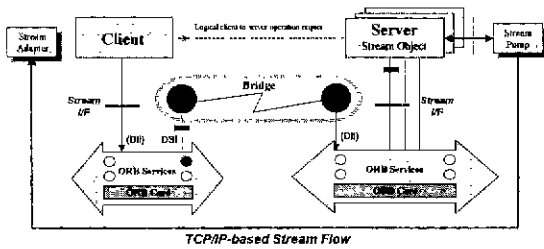


그림 4. 대화형 분산 플랫폼의 전체 흐름도

클라이언트 객체는 Stream 인터페이스에 대해 DII 방식을 사용하여 스트림 서비스 객체에 대해 필요한 오퍼레이션을 호출한다. 이 호출은 DSI(Dynamic Skeleton Interface)를 사용하는 양쪽 ORB 사이의 브릿지를 통해 전달된다. 이를 통해 클라이언트는 다른 ORB상의 구현 객체를 마치 자신의 ORB상에 있는 구현 객체처럼 호출한다.

클라이언트 종단에는 스트림 어댑터 모듈이 추가적으로 포함된다. 이 모듈은 스트림 서비스 객체가 존재하는 종단의 스트림 펌프 모듈과 스트림 플로우를 설정하여 연속 미디어 데이터를 수신하는 기능을 수행한다. 이를 위해 스트림 어댑터는 내부적으로 버퍼를 유지, 관리하며 오디오나 비디오 디바이스 드라이버를 이용하여 해당 미디어 데이터를 사용자에게 표현하는 역할을 담당한다.

오디오, 비디오 스트림에 대한 제어 메시지의 전달은 ORB를 통한 클라이언트로부터 스트림 서비스 객체로의 오퍼레이션 기동(invocation)이라는 형태로 이루어진다. 스트림 서비스 객체는 요구된 제어 메시지를 바탕으로 스트림 펌프와 상호동작하여 연속 미디어 데이터의 전송을 조절하게 된다 스트림 서비스 객체는 연속 미디어에 대한 status를 일관성있게 관리하기 위해 상태 전이 머신(state transition machine)을 운영한다 스트림의 status는 클라이언트로부터의 제어 메시지에 따라 동적으로 변화되는데, 스트림 서비스 객체