

CORBA 기반의 의료영상회의 시스템을 위한 Web Client 설계

김정현, 강재효, 박세명, 최항목
인제대학교 전산학과

Design of Web Client for Medical Image Conference System Based on CORBA

Jeung-Hyun Kim, Jae-Hyo Kang, Se-Myung Park, Hang-Mook Choi
Department of Computer Science, Inje University

요약

본 연구에서는 Web 환경에서 분산객체 기술인 CORBA(Common Object Request Broker Architecture)를 이용하여 Java Applet으로 구현되는 의료영상회의 시스템의 Web Client를 설계 구현하였다. 플랫폼(platform)에 독립적이고, 다양한 개발 언어를 지원하는 CORBA를 기반으로 한 이 의료영상회의 시스템은 시스템의 개발, 확장성 및 유지 보수 측면에서 아주 뛰어나다. 또한 객체 지향 언어이면서 다양한 플랫폼에 구애를 받지 않는(Architecture Neutral) Java 언어를 이용하여 Web 환경에서 쉽게 실행 가능한 Applet으로 의료영상회의를 할 수 있는 Web Client를 설계 구현하였으므로 Web 브라우저를 이용할 수 있는 환경이라면 손쉽게 이용할 수 있는 장점을 가진다.

1. 서론

최근 들어 병원내의 의료정보 전산화 작업이 활발히 이루어지고 있는 추세이다. 또한 네트워크를 통해 전산화 된 의료정보 및 환자의 의료영상(MRI영상, CT영상, X-ray영상 등)등이 공유되어 사용되고 있다[1]. 이런 환경에서 정보의 공유만이 아닌 이러한 정보를 이용한 회의 시스템 개발이 필요하게 되었다.

이러한 필요성에 입각하여 기존의 Web을 기반으로 브라우저를 통해 쉽게 의료영상 회의를 할 수 있는 클라이언트 프로그램에 대해 설계 구현하였다. 현재 분산 환경에서 플랫폼에 독립적이고, 객체 지향적이며 실시간으로 데이터 처리를 가능하게 하는 CORBA[2,3]를 기반으로, HTTP의 비연결성 및 Java Applet 보안성등의 단점을 보완하여 사용자의

요구를 수용할 수 있도록 설계하였다.

의료영상회의 시스템(Medical Image Conference System)은 회의 개설자가 회의를 개설하고 진행하면서 화이트보드를 통해 참여자들과 양방향으로 정보를 주고받으며 이미지 처리를 요구할 수 있는 권한을 가진다. 서버는 기본적으로 ORB(Object Request Broker)가 내장된 'Netscape Enterprise Server'[4,6,7] 환경이며, 의료영상의 이미지 처리 및 사용자 요구에 대한 서비스를 처리하는 속도를 빨리 하기 위해 'Visual C++'로 구현하였으며 Client는 Web 브라우저를 통해 쉽게 실행 가능한 Java 애플릿으로 구현하였다.

본 논문에서는 의료영상회의 시스템을 개발함에 있어 고려된 개발환경과 클라이언트 부분의 구성요소 및 기능에 대해 살펴보고, 각 기능을 구현함에

있어 애플릿의 특수성을 고려하여 어떻게 구현되어 지는지 살펴보겠다.

2. 개발 환경

현재 Web 상에서 상호작용(interactive)적으로 동작하면서, 원격 시스템의 객체를 호출할 수 있는 방법에는 Java RMI(Remote Method Invocation)와 DCOM(Distributed Component Object Model), RPC(Remote Procedure Call) 그리고 CORBA를 이용하는 방법이 있다.

RMI는 언어중립의 메시징 서비스(language-neutral messaging)를 지원하지 않는다. 다른 언어의 구성 요소를 가지지 않는 순수한 자바 솔루션(Pure Java Solution)을 구축하기를 원한다면 RMI를 이용하는 것이 쉬운 것이다. 또한 RMI는 동적 호출(Dynamic Interface)과 인터페이스 저장소를 지원하지 않는다.

DCOM object는 OOP(Object-Oriented Programming) 관점에서의 object는 아니다. DCOM의 가장 큰 단점은 플랫폼 의존적이라는 것과 분산 Naming 서비스를 지원하지 않는다. 윈도우 플랫폼을 기반으로 서버측에 MTS(Microsoft Transaction Server)의 설치가 필요하다.

RPC는 객체의 메소드 호출이 아니라 단지 함수 호출이다. 이는 같은 이름의 함수를 호출하는 것에서 문제를 발생시킨다. 즉 같은 이름의 함수는 같은 방법으로 구현되어 지는 것이다[4].

본 연구를 위해서는 'Visigenic'사의 'Visibroker'[9,10]라는 ORB를 내장한 'Netscape'사의 'Netscape Enterprise Server'를 Web 서버로 사용하였으며, 개발언어로는 'SUN'사의 'JDK1.1.5'[8]를 이용하였다.

본 논문에서 개발하고자 하는 의료영상회의 시스템은 Java 언어로만 개발하는 것이 아니라 서버에서의 사용자 요구에 대한 서비스 속도향상을 위해 'Visual C++'를 이용하였다. CORBA는 각기 다른 언어로 만들어진 객체간의 연결 통로로써 매우 뛰어난 미들웨어이며, 플랫폼에 독립적이라서 호환성이 뛰어나고 개발 및 유지 보수가 쉽다[2,3].

2.1 Java Applet

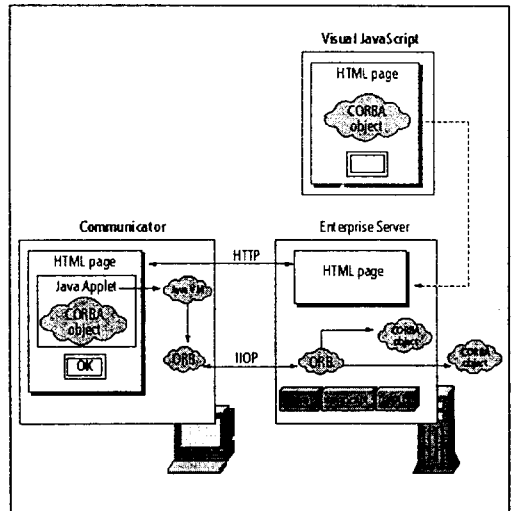
Java Applet을 이용하여 회의 시스템의 Web Client 구현시 주요 문제점은 Java 자체의 보안(security)이다. 보안상의 문제로 Java Applet은 한번 접속한 서버에 대해서만 접속하고, 로컬 머신(local machine) 내부의 파일에 대해 읽고 쓰는 것이 금지되어 있다.

회의 시스템에서는 아주 심각한 문제이다. 회의 개설자의 시스템 내에 있는 이미지를 서버에 전달하여 회의 참여자들이 모두 받아 볼 수 있어야 하는데 개설자의 시스템 내의 이미지 파일을 읽고 쓸 수 없다는 것이 문제가 되므로 이러한 Applet 보안문제를 안정하게 해결하는 것이 필요하다.

이 문제는 Netscape사에서 제공하는 'Java Capabilities API'를 이용하여 해결한다. 디지털 서명(digital signature)을 획득한 후 Capabilities API를 이용한 Applet 프로그램은 이러한 보안 문제를 해결할 수 있다. Netscape Navigator는 이미 Capabilities API를 포함하고 있다[5].

2.2 Netscape Enterprise Server

Netscape Enterprise Server를 이용하여 Client Applet이 구동형태는 <그림 1>과 같으며 각 구성요소의 기능은 다음과 같다[5].



<그림 1> Enterprise Server의 Applet 구동

● Netscape Communicator

ISB(Internet Service Broker)를 탑재하여, Client가 CORBA 객체의 서비스를 사용 가능하도록 해준다. 즉 ORB를 탑재하고 있어 CORBA의 IOP(Internet Inter-ORB Protocol), Naming Service 등을 사용 할 수 있게 한다.

● Netscape Enterprise Server

'Visigenic'사에서 개발한 'Visibroker for Java'와 'Visibroker for C++'를 포함하고, Java와 C++을 위한 Netscape ISB를 포함한다. 또한 IDL(Interface Definition Language) 컴파일러와 WAI(Web Application Interface)를 포함하고 있다. WAI는 IDL 명세를 가지고 네트워크 상에서 CORBA object들과 통신을 가능하게 해주며 NSAPI(Netscape Server Application Interface)의 기능을 증대 시킨다.

● Netscape's Visual JavaScript

CORBA object들을 HTML 문서에 통합할 수 있게 한다. 사용자는 이런 object를 통해 상호작용을 통해 실시간으로 작업한다.

그림<1>에서와 같이 CORBA의 IOP를 이용하여 C++언어로 구현된 서버의 이미지 처리 객체를 Java언어로 구현된 Client를 이용하여 서비스를 받을 수 있으며 회의 Web 페이지에 접속하여 Java Applet을 다운로드 받아 실행하는 순간부터는 HTTP 프로토콜을 이용하는 것이 아니라, CORBA 객체간의 통신으로서 IOP를 통해 서버와 클라이언트 사이에서 통신을 한다.

3. Client 부분의 회의 구성요소

3.1 고려사항

회의 시스템에 참가하는 참가자들은 지역적으로 멀리 떨어져 있으면서 네트워크를 통하여 공동작업을 하게 된다. 공동작업에서 고려되어야 할 점은 통신(communication), 협동작업(collaboration), 중재(coordination)등이다. 네트워크를 통한 회의 시스템에 필요한 본 시스템에서 처리하게 되는 자료로는 정지화상, 음성, 텍스트, 그래픽 등이 있는데 그 목적으로는 사용자간의 협동 작업을 보다 원활히 지원

하기 위함에 있다[12].

시스템의 작업 공간 제어구조를 살펴보면 응용 프로그램이 실행되는 위치에 따라 시스템 내에 오직 하나의 응용 프로그램만이 존재하는 단일 실행구조(centralized architecture)와 회의에 참여하는 모든 시스템들이 동일한 응용프로그램을 가지고 실행하여 결과를 얻는 복제실행구조(replicated architecture)가 있다[12]. 이 중 본 시스템에서는 Web 환경에 맞추어 Web 브라우저로 서버에 접속만 하면 쉽게 실행할 수 있도록 단일 실행구조를 선택하였다.

공유 작업 공간을 사용하는 형식에는 참가자들이 원하는 시점에 동시에 사용하는 자유사용법(simultaneous access)과 사용권한이 있는 한 참가자만이 공유 작업 공간을 사용하는 사용권한법(one-person-at-a-time access)이 있는데[12], 본 연구에서는 동적 수행이 가능한 CORBA를 기반으로 하였기에 공유 작업 공간을 자유롭게 수행할 수 있는 자유사용법을 채택하였다.

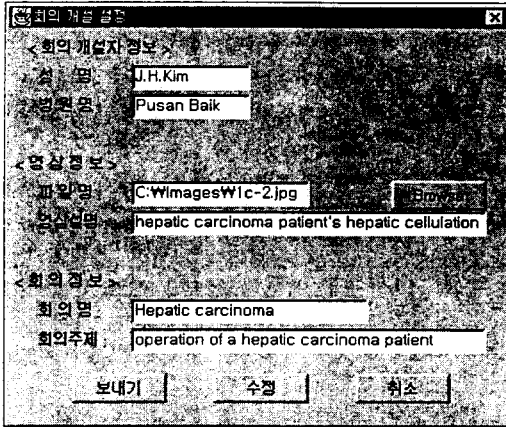
위에서 살펴본 바와 같이 단일 실행구조를 사용하는 이유는 회의의 모든 참여자들이 이미지처리를 요구했을 시 네트워크 및 서버의 부하가 커지며 실행속도가 늦어지는 것을 방지하고, 회의의 원활한 진행을 위해서이다. 하지만 화이트보드인 공유 작업 공간은 모두 사용할 수 있으며 텍스트 및 음성 데이터를 주고받을 수 있도록 자유 사용법을 사용하였다.

3.2 회의 개설자 Applet의 주요기능

회의를 개설하는 개설자는 회의 Web 페이지에 접속할 때 회의를 개설할 수 있는 권한을 획득한 후, 회의 개설자 Applet을 다운로드하여 실행한다.

개설자 Applet은 회의를 개설할 수 있는 메뉴와 이미지 처리를 서버에 요구할 수 있는 권한을 가지고 회의를 진행한다. 회의를 개설하는 메뉴는 <그림 2>와 같다.

<그림 2>에서 보듯이 회의를 개설할 사용자는 회의 개설에 필요한 정보를 입력하게 된다. 'Browse' 버튼을 누르게 되면 윈도우 자체의 불러오기 다이얼로그 박스가 실행되고, '보내기' 버튼을 누르면 입력된 정보와 이미지 파일을 서버로 보내면서 회의를 개설하게 된다[3,7,8,9,10,11].



<그림 2> 회의의 개설 설정 화면

```
//Maker.idl
module Maker{
// 이미지를 전송하기 위한 byte 배열
typedef sequence<octet> octetSeq;
// 회의 개설 정보 structure
struct ConfInfo{
    string fileName; string MakerName;
    string Hospital; string Conf_Subject;
    string Image_desc; string Conf_Name;
};
//회의 서버와 interface 정의
interface ImageConf{
// 회의 개설 정보 서버로 전달
void send_info(in ConfInfo CInfo);
//이미지전송 및 수신
void image_send(in octetSeq Imagedata);
void Image_recieve(out octetSeq Imagedata);
// 마우스 드래그 이벤트 발생시
void drawPoint(in long Px, in long Py);
//이미지처리 이벤트 발생시
void Contrast();
void Threshold(in long s, in long e);
void Bitsplain(in long n);
    ...
};
};
```

<그림 3> 개설자 Applet IDL 정의

<그림 3>은 개설자 Applet의 IDL 파일 내용 [2,3,7,8,9,10]이다. 이미지 파일은 JPEG 파일로 압축하여 Java byte 배열 형태로 전송하게 된다. 전송된 파일을 임시로 저장하여 뷰(View)로 불러온다. 처리된 이미지를 저장 할 수 있도록 하였다.

3.3 회의 참여자 Applet의 주요기능

회의 참여자 Applet은 개설자 Applet과는 달리 회의 개설 메뉴와 이미지 처리 메뉴는 없다. 이미지 처리는 개설자만이 가능하도록 하고, 단지 화이트보드 기능 및 텍스트 데이터 와 음성 데이터를 회의 참여자들과 주고받을 수 있다.

개설자가 이미지 처리를 했을 시 처리된 이미지를 받아보면서 회의를 진행한다.

3.4 회의 진행 절차

회의 Web 페이지에 접속하면 먼저 지정된 ID와 패스워드를 입력하여 사용가능 여부를 인증 받고, 회의 개설과 참여 여부에 따라 다른 페이지도 이동한다. 개설페이지에서는 회의 참여인원을 입력하여 개설자 Applet을 다운로드하여 회의를 개설하게 되고, 참여 페이지는 현재 개설되어 있는 회의 정보 및 참여 인원, 참가자 명단, 개설시간 등을 볼 수 있고 참여하고자 하는 회의에 참여 할 수 가 있다.

개설자는 회의 참석인원이 모두 참석하게 되면 회의를 진행해 나간다.

4. 향후 발전 방향

현재 병원의 의료정보들이 전산화되어 가는 과정이기는 하나 병원 하나에만 국한된 경우가 많다. 많은 병원들이 환자의 신상 정보를 제외한 질병에 관한 정보를 중앙 집중식으로 전산화여 관리한다면 더욱더 체계적인 관리가 될 것이며 많은 양의 정보를 얻을 수 있을 것이다.

본 시스템은 현재 의료영상회의에 초점을 맞추었으나 중앙 집중식으로 의료정보들이 체계적으로 관리가 될 때 회의 진행중에도 의료정보 검색과 특수한 병력에 대한 정보등을 쉽게 얻을 수 있을 것이다.

서버는 이미지 파일의 크기, 전송속도, 처리속도등을 고려하여 객체들의 배치 및 원활한 통신방법에

대한 연구가 이루어져야 하며, 또한 회의 진행 상황을 개설자 관점에서 DB에 저장 가능하도록 하고, 회의 목록, 주제등을 목록화하여 사용자가 검색 할 수 있도록 구성되어야겠다.

참 고 문 헌

- [1] Tom af Klercker, Máns af Klercker, "Decision support system for primary health care in an inter/intranet environment", ELSEVIER Computer Methods and Programs in Biomedicine 55 P31-P37, 1998.
- [2] Robert Orfali, Dan Harkey, "Client/Server Programming with JAVA and CORBA", Wiley, 1997.
- [2] Jon Siegel, "CORBA Fundamentals and Programming", Wiley, 1996.
- [3] Netscape, "CORBA:Catching the Next Wave", [Http://developer.netscape.com](http://developer.netscape.com), 1998.
- [4] Netscape, "JAVA CAPABILITIES API", [Http://developer.netscape.com](http://developer.netscape.com), 1998.
- [5] Netscape, "Writing Applications with WAI", [Http://developer.netscape.com](http://developer.netscape.com), 1998.
- [6] Netscape, "Netscape Enterprise Server 3.5.1 Notes for Javal", [Http://developer.netscape.com](http://developer.netscape.com), 1998.
- [6] Sun, "JAVA Development Kit 1.1.5 Documentation", [Http://java.sun.com/doc/](http://java.sun.com/doc/),1997.
- [8]Visigenic, "VisiBroker for JAVA Programmer'sguide 3.2", [Http://www.visigenic.com](http://www.visigenic.com), 1997.
- [9] Visigenic, "VisiBroker for JAVA Reference guide 3.2", [Http://www.visigenic.com](http://www.visigenic.com), 1997.
- [10] 박재현 외 1명, "코어 자바", 영한, 1997.
- [11] 정진호 외 2명, "DeskShrare : LAN상에서 그룹작업을 위한 멀티 미디어 탁상 회의 시스템", 정보과학회 논문지 제 1권 제 1호 p12 - p21, 1995.9.