

의료영상 회의시스템을 위한 서버 시스템의 요구분석 및 설계

○
김정현, 강재효, 성병우, 성재철, 김상균, 박세명, 최항목, 최홍국
인제대학교 전산학과

Requirement Analysis and Design of Server System for Medical Image Conference System

○
Jeung-Hyun Kim, Jae-Hyo Kang, Byung-Woo Sung, Jae-Cheol Sung, Sang-Gyun Kim,
Se-Myung Park, Hang-Mook Choi, Heung-Kook Choi
Department of Computer Science, Inje University

요 약

본 논문에서는 객체 웹(Object web) 환경에서 구현된 의료영상 회의시스템이 가지는 다양한 영상 처리 객체에 대한 추가, 삭제 등의 관리 문제와 Web환경의 특성에 따른 서버 시스템의 요구분석에 대해 고찰해 보고자 한다. C++로 구현된 영상 처리 객체는 그 특성상 이미지처리에 따르는 서버에서의 부하 문제와 이미지 전송에 따르는 네트워크 부하 문제를 고려하여 설계, 구현되어야 한다. 분산처리 객체 환경의 표준인 CORBA(Common Object Request Broker Architecture)를 이용하여 위 문제를 해결한 영상 처리 객체 관리자틀 제시하고, 본 시스템의 전체 구성도 및 각 객체간의 인터페이스를 정의하였다.

1. 서론

인터넷, 인트라넷(Intranet), 엑스트라넷(Extranet) 환경에서 운영될 차세대 웹은 기업 대 기업, 고객 대 기업간 트랜잭션의 복잡한 요구사항을 다룰 수 있어야 한다. 이를 위해 웹은 업종별 애플리케이션을 실행할 수 있는 고수준의 상호 작용 매개체로 발전되어야 한다. 기존의 HTTP/CGI 구조의 많은 결함 및 기능 부족으로 인하여, cookies, ISAPI(Internet Server API), NSAPI(Netscape Server API), ASP(Active Server Page) 등이 개발되었지만, 보다 한 수준 위로 도약하기 위해 웹은 분산객체를 필요로 하고, 이러한 이유로 'Object

Web' 기술이 대두되고 있다[1]. 즉 웹 환경은 이제 단순히 저장된 문서만을 해석하여 브라우저 상에서 보여주는 단계에서 동적인 분산시스템을 구성하는 단계로 발전하고 있다[2].

본 논문에서는 Java/CORBA 결합의 객체 웹 환경에서 의료영상 회의 시스템을 위한 기능을 분석하고 이를 설계하였다. Java를 이용하면 사용자는 웹 서버에 접속하여 Java Applet을 다운 받아 CORBA 클라이언트를 실행하는 환경을 제공할 수 있다. 이는 단기간 내에 여러 클라이언트에게 프로그램을 사용자에게 배포 할 수 있으며, 설치 및 갱신이 용이함을 의미한다. 또한 Java의 다중 스레드, 가비지

콜렉션(garbage collection), 에러 관리는 강력한 네트워크 객체 작성을 보다 용이하게 한다. 즉 Java의 플랫폼 독립성과 CORBA의 위치 투명성을 고려하면 Java는 클라이언트 및 서버용 CORBA 객체를 구현하기 위한 매우 이상적인 언어이다.

이러한 장점을 이용하여 Java Applet으로 클라이언트를 구현하고, 전반적인 회의 관리 서버는 Java 응용 프로그램으로, 영상처리 객체는 처리속도를 고려하여 C++로 구현한다.

본 논문에서는 우선 의료영상 회의시스템에 대해 설명하고, 객체 웹 환경에서의 서버에 대한 요구분석 및 영상객체 관리자에 대해서 살펴본다. 그리고 시스템의 구조 및 설계, 구현설명으로 구성된다.

2. 의료영상 회의시스템

2.1 의료영상 회의시스템의 개요

의료분야에서는 환자의 진료를 위해 환자의 영상자료에 대한 다양한 영상처리가 요구되고 있다. 이러한 요구에 따라 초기에는 하나의 컴퓨터에 영상처리를 위한 기능과 화면표시기능을 가지는 시스템에서 통신망과 인터넷의 보급확대로 인해 원거리에 있는 전문가들이 동일한 영상자료에 대한 회의 및 진료시스템으로 발전되고 있다[3]. 이러한 시스템의 필요기능을 살펴보면 크게 동일한 영상을 여러 회의참가자들이 공유할 수 있는 기능과 영상 처리기능, 그리고 효율적인 회의 진행을 위한 Whiteboard기능과 의사전달기능으로 나눌 수 있다. 회의참가자들에게 동일한 영상에 대하여 토론할 수 있는 환경, 즉 Whiteboard기능과 의사전달기능을 제공하며, 토론에 보다 객관적인 근거자료를 제공하기 위해 주어진 영상에 대한 다양한 분석 및 처리기능을 제공하여야 한다. 이러한 의료영상 회의시스템은 보다 빠르게 전문가의 의견을 수렴할 수 있다는 측면에서 미래의 의료체제의 변화를 주도할 중요한 매체가 될 것이다 [4].

2.2 서버시스템의 요구분석

의료영상 회의 시스템의 서버에 관한 요구분석은 다음과 같다.

첫째, 모든 회의 참여자들이 전송하는 메시지, Whiteboard 기록 정보, 회의 개설자에 의한 영상처리에 관한 요청[2]등을 처리하는, 회의의 전반적인 것을 관리하는 회의 관리 객체가 필요하다.

둘째, 회의 시스템의 특성상 클라이언트 객체의 참조를 관리하는 기능이다. 1: n의 통신이 가능하도록, 콜백(CallBack) 매커니즘을 이용하며, 회의 관리자는 클라이언트의 콜백 객체의 참조를 관리하며, 이를 이용하여 서버가 클라이언트를 호출 할 수 있도록 한다.

셋째, 다양한 영상처리 객체를 관리하는 객체 관리자가 필요하다. 영상처리 객체가 활성화 상태인지, 비 활성화 상태인지를 체크하고 만약 비 활성화 상태라면 Visibroker의 OAD(Object Activation Daemon)[5,6]을 이용하여 활성화 시켜, 요청을 처리한다. 또한 분산된 영상처리 객체에 대해 서버부하 및 네트워크 부하를 고려하여 영상처리 요청을 하게 하여야 한다. 객체 관리자의 또 다른 기능은 영상처리 요청에 대한 해석이다. 이는 클라이언트로부터의 영상처리에 대한 요청은 하나의 영상처리 객체가 추가, 삭제등에 관계없이 하나의 메소드로 이루어져 있기 때문이다. 객체 관리자의 기능에 의해 영상처리 객체의 비 활성화 상태가 가능하므로 서버의 자원 낭비를 줄이고, 서버 부하 및 네트워크의 부하를 보완 할 수 있으며, 시스템의 확장 및 유지 보수를 용이하게 할 수 있다[4].

넷째, 회의에 사용되어지는 영상을 관리하는 영상 관리 객체가 필요하다. 영상관리 객체는 영상처리 객체와 같은 서버에 존재하며, 원본 영상과 이미지 처리를 한 영상 모두를 관리하고 있다. 이는 이미지 전송에 따른 시간을 줄여 준다. 즉, 영상처리 요청은 처리할 영상의 이름만 전달하게 되고, 영상 처리 객체는 영상관리 객체를 참조하여 처리할 영상을 얻는다.

마지막으로, 영상처리 객체의 분산이다. 영상처리 객체를 분산했을 때 그 객체에 대한 참조를 관리하는 문제와 영상처리 객체를 어떻게 분산해야 네트워크 및 서버 부하를 줄일 수 있는지는 현재 연구중인 본 시스템의 성능 분석 후 그 결과를 토대로 이루어져야겠다.

3. 의료영상 회의 시스템 설계

3.1 의료영상 회의 시스템의 전체 구성도

위에서 살펴본 서버 시스템의 요구분석에 따라 설계된 의료영상 회의 시스템의 전체 구성도는 그림 1과 같다.

그림 1에서 클라이언트 Applet 대 회의 관리자

(Conference manager), 회의 관리자 대 객체 관리자 (Object manager), 객체 관리자 대 영상처리 객체는 각각 CORBA의 IIOP(Internet Inter-ORB-Protocol)를 이용한다.

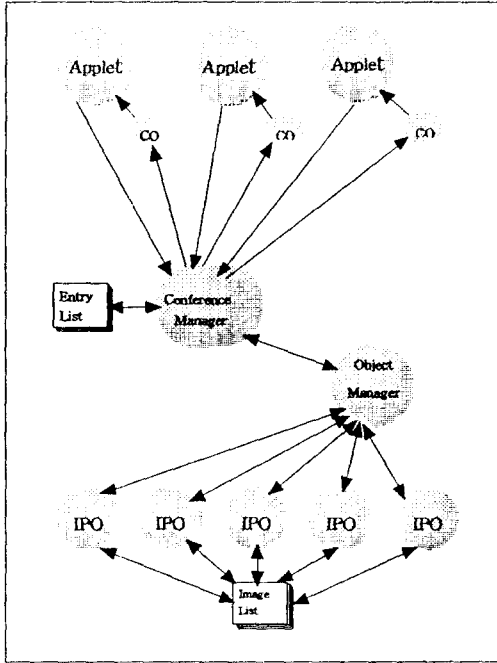


그림 1. 의료영상 회의 시스템의 전체 구성도

회의 관리자와 객체 관리자는 웹서버와 동일한 시스템에 존재하는 객체이며, 영상처리 객체는 같은 시스템에 존재하거나 다른 서버 시스템에 분산되어 운용 될 수도 있다. 영상 관리 객체는 영상처리 객체가 존재하는 모든 시스템에 설치되어야 한다.

3.2 IDL(Interface Definition Language) 정의

3.2.1 클라이언트 Applet 과 회의 관리자

그림 2[1,5,7]는 클라이언트 Applet과 회의 관리자 간의 인터페이스 및 메소드를 정의 한 IDL이며 선언부는 객체 관리자와 영상 처리 객체가 공통으로 사용하게 된다. 인터페이스 및 메소드는 중요한 메소드만 나열하고 나머지는 지면 관계상 생략하였다.

선언부는 영상을 전송하기 위한 byte 배열과 영상처리 요청 파라미터를 위한 정수형 배열, 회의 개설 정보 및 영상 처리 요청 파라미터 구조체 및 영

상 처리 객체에 접근을 못할 경우를 대비한 예외사항을 선언한다.

인터페이스 및 메소드에서는 회의 참여자들의 의사소통을 위한 메시지, Whiteboard 기록 정보 그리고 영상을 전송하기 위한 메소드 및 영상처리 요청 메소드를 정의한다. 콜백 메카니즘을 위한 인터페이스 및 메소드는 생략하였다.

```
//Mics.idl
module Mics {
    // 선언부
    typedef sequence<octet> Image_data;
    typedef sequence<long> param_Index;
    struct Conf_Info {
        string fileName; string MakerName;
        string Hospital; string Conf_Subject;
        string Image_desc; string Conf_Name;
    };
    struct ip_Param {
        string Ip_operation;
        param_Index param;
        long index;
    };
    exception IPONotFound(string IPOName);

    // 인터페이스 및 메소드 정의
    :
    interface ConferenceM {
        :
        oneway void sendMessage(in string msg);
        oneway void sendImage (in Image_data r_image,
                               in long len, in string filename);
        void ip_Operation(in ip_Param ippar)
            raises(IPONotFound);
        oneway void Dragpoint(in long Px, in long Py);
    };
};
```

그림 2. Mics.idl Interface 정의

3.2.2 회의 관리자와 객체관리자

회의 관리자와 객체 관리자간의 인터페이스 정의는 단지 처음 회의 개설 시 회의에 사용될 원본 영상을 영상처리 객체에게 전달하기 위한 메소드와 영상처리를 요청하는 메소드 만으로 구성된다.

객체 관리자는 영상처리 요청 메소드 부분에서 요청을 해석하여, 해석된 결과에 따라 적합한 영상처리객체에게 영상처리를 요청하고 처리된 영상을 회의의 관리자에게 돌려주는 것이다.

그림 3[1,5,7]은 회의 관리자와 객체 관리자간의

인터페이스를 정의한 'Omanager.idl' 내용이다.

```
//Omanager.idl
// Mics.idl의 선언부를 공동으로 사용
#include "Mics.idl"
module Omanager{
    interface ObjectM{
        // 회의 개설시의 원본 영상 전송
        oneway void sImage(in Mics::Image_data
            r_image, in long len, in string filename);
        // 영상 처리 요청
        Mics::Image_data Image_OP(in ip_Param)
            raises(Mics::IPONotFound);
    };
};
```

그림 3 Omanager.idl 인터페이스 정의

3.2.3 객체 관리자와 영상처리 객체

객체 관리자와 영상처리 객체간의 인터페이스 정의는 거의 동일하다. 단지 영상처리 객체에는 현재 상태를 체크하는 메소드가 추가된다. 영상 처리 객체는 최초 활성화 시 Visibroker OAD에 등록하는 작업을 선행해야만 한다.

3.3 의료영상 회의시스템의 구현 예

그림 4[1,5,8,9]는 앞서 살펴본 요구분석에 의해 설계된 의료영상 회의 시스템에서 간 조직 영상을 이용하여 회의를 진행하는 클라이언트 실행 화면이다.

그림의 좌측은 원본 영상이며, 우측은 영상처리 요청에 의해 처리된 영상이다. 의료영상 아래에는 어떤 영상을 어떻게 영상처리를 했는지에 관한 현재 활성화 된 영상의 설명 박스가 있으며, 좌측 하단은 의사소통 기능으로서 채팅 기능을 나타내는 것이다. 또 다른 의상 소통 기능인 Whiteboard 기능의 예로 좌측 영상에 마우스로 드래그 했을 때의 실행 화면이다.

상하, 좌우 스크롤 바의 기능은 영상이 실행 화면보다 클 경우를 고려한 것으로, 스크롤 시 좌측의 원본 영상과 우측의 처리된 영상이 똑 같은 좌표로 이동한다.

우측하단의 빈 공간은 차 후 개발 될 기능을 위하여 남겨 놓았으며, 회의에 관한 전반적인 정보는 첫 탭 다이얼로그 박스에 표현된다.

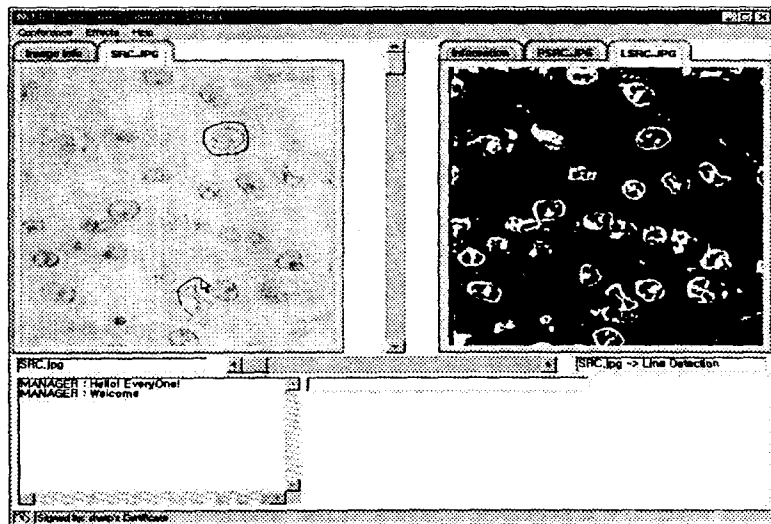


그림 4. MICS의 실행 화면

4. 결론 및 향후 연구 과제

기존의 Web환경을 보완하기 위해 분산객체를 더한 객체 Web 환경에서 개발된 본 시스템에서는 Web 환경에서의 장점과 분산 객체 기술의 장점을 취합함으로써 시스템의 개발, 확장이 용이하고 유지보수가 쉽다.

또한 사용자에게 프로그램의 설치, 유지 보수에 신경 쓰지 않고, 위치 투명성을 확보하여 어디에서나 브라우저를 통하여 회의에 참석 할 수 있는 편의성을 제공한다.

객체 관리자틀 들으로써 영상처리 객체의 관리, 추가, 삭제 및 보완이 용이하다.

앞으로 연구되어야 할 것으로는 현재 지원하고 있는 의사소통 기능인 채팅과 Whiteboard 기능만으로는 원활한 회의 진행이 어려우므로 음성데이터를 지원하여 의사전달 체계를 개선해야 할 것이다. 그리고 전산화된 의료정보를 실시간으로 직접 검색 할 수 있는 기능을 추가하여 회의에 필요한 추가자료를 손쉽게 활용할 수 있는 환경의 제공도 고려되어야 할 것이다. 특히 본 시스템의 정확한 성능 분석을 통하여 영상 파일의 크기, 전송속도, 처리속도등을 고려하여 영상처리 객체들의 분산배치 및 원활한 통신 방법에 대해 지속적인 연구가 필요하다.

참 고 문 헌

[1] Robert Orfali, Dan Harkey, "Client/Server Programming with JAVA and CORBA", Wiley, 1997.
[2] 김정현 외 3인, "CORBA를 이용한 의료영상회의 시스템을 위한 Web Client 설계", 멀티미디어학회 춘계발표대회 논문집, June. 1998.
[3] A. Samperdro 외 5인, "Open and Distance Learning in Citopathology on the Internet", 5th. ESACP Congress, Oslo. May 1997.
[4] 김정현 외 6인, "CORBA 기반 의료영상회의 시스템을 위한 Server 설계", 한국 정보과학회 가을학술발표 논문집 Vol. 25, No. 2, October, 1998
[5] Visigenic, "Vsibroker for Java Programmer Guide 3.2", <http://www.visigenic.com>, 1997
[6] Visigenic, "VisiBroker for JAVA Reference guide 3.2", <Http://www.visigenic.com>, 1997.

[7] Jon Siegel, "CORBA Fundamentals and Programming", Wiley, 1996.
[8] Netscape, "Netscape Enterprise Server 3.5.1 Notes for Java", <Http://developer.netscape.com>, 1998.
[9] Netscape, "JAVA CAPABILITIES API", <Http://developer.netscape.com>, 1998.