

3계층 모델을 이용한 웹 환경에서의 원격 제어 게이트웨이 시스템의 설계 및 구현

정 경 열*, 윤 병 수**, 하 주 식***, 류 길 수***

Design and Implementation of Remote Control Gateway in Internet Environment Using 3-Tier Model

Kyung-Youl Jung, Byung-Soo Youn, Joo Sik-Ha Keel-Soo Rhyu

- * 한국기계연구원
- ** 한국해양대학교 대학원 컴퓨터 공학과
- *** 한국해양대학교

Abstract : In spreading an internet service and developing a network technology, The interest of study about the remote control system using web environment increased. In treatise, We describe a design and an implementation of the gateway system that is based 3-tier model using corba in middleware. This system is applicatble various system environment because of using middleware and overcome a time limit and range limit because of using web browser interface.

Key words: Corba(코바), Middleware(미들웨어), 3-Tire(3 계층)

1. 서 론

선박이나 산업 현장에서의 인력난 해소와 생산성의 향상을 위해 많은 작업이 표준화되어 각 공정상의 장비들이 점점 자동화되고 이러한 장비에 대하여 원격의 장소에서 제어를 하고자 하는 연구가 많이 이루어졌다. 하지만 기존의 원격 제어 시스템은 클라이언트/서버 형태의 2계층의 소켓 통신으로 그 거리에 있어서 제한된 범위내의 한계를 가지고 있으며 대부분의 Master 시스템은 PC기반의 윈도우를 운영체제로 이용하고 있다.

현재 인터넷을 이용한 서비스의 확산으로 고속의 데이터 전송을 지원하는 장비의 보급과 보다 많은 데이터를 전송할 수 있는 네트워크 전송 기술의 발달로 많은 네트워크 부하를 요구하지만 언제 어디서나 접속이 가능한 웹을 이용한 원격 제어 시스템에 대한 관심과 연구가 활발하다.[1] 그리고 PC기반의 시스템보다는

용 단말기의 개발로 가격 경쟁력에서 Master 시스템의 변화가 예상된다.[2]

본 논문에서는 클라이언트/미들웨어/서버 형태의 3계층으로 미들웨어로서 CORBA를 이용하여 게이트웨이를 구현함으로써 이중의 시스템이나 운영체제에 독립적으로 여러 환경에 적용가능하고 자바 애플릿을 이용하여 웹 브라우저 상에서 원격의 장비에 대하여 거리와 시간의 한계를 극복하여 제어할 수 있는 시스템을 구현하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 3계층기반 모델의 제 2계층에서 구동되는 미들웨어들에 대한 소개와 특징을 서로 비교한다. 3 장에서는 미들웨어로서 CORBA를 이용하여 웹 환경에서 원격 제어 게이트웨이 시스템을 구현에 관하여 기술하고 마지막으로 4 장에서는 결론 및 향후의 연구 과제에 대하여 논한다.

2. 미들웨어

2.1 미들웨어 계층의 개요

계층이란 클라이언트와 서버에 걸친 어플리케이션의 논리적 분할을 의미한다. 2 계층 구조에서는 클라이언트 계층은 사용자 인터페이스를 통해 수행된 명령데이터를 서버에 전송하는데 있어 어떻게 서버측에서 조직되고 저장되는지 알아야 한다. 하지만 3 계층 구조에서는 클라이언트 계층은 사용자 인터페이스로부터 요청된 명령데이터를 미들웨어에 전송하거나 미들웨어를 통해 수신된 서버측 응답을 처리한다. 서버계층은 클라이언트로부터의 요청을 수신한 후 데이터를 처리하여 그 응답을 제공한다. 3 계층 구조에서 미들웨어 계층은 클라이언트와 서버 계층 사이에 위치하여 이들간의 네트워크 상에서 사용하기 쉽고 관리하기 편한 중간 계층을 제공함으로써 클라이언트와 서버가 독립적으로 보다 쉽고 빨리 사용자가 원하는 어플리케이션을 작성할 수 있게한다.[3] 대표적인 미들웨어로서 마이크로 소프트의 유니버설 플러그 앤 플레이(이하 UPnP), 썬 마이크로시스템즈의 지니와 OMG의 CORBA가 있다.

2.3 유니버설 플러그 앤 플레이(UPnP)

마이크로 소프트사의 UPnP는 기존 윈도우의 플러그 앤 플레이기능을 네트워크로 확장하여 미들웨어로서 적용가능하도록하는 기술이다. 유니버설 플러그 앤 플레이는 제품 간 상호 접속을 위해 이루어지는 복잡한 설정 작업이나 환경 설정 작업을 생략하고 네트워크상에 접속하면 자동으로 기기를 찾아 사용할 수 있는 기술이다.

지니 시스템은 RMI라는 분산 객체 기술을 통해 객체 간 통신을 수행하는데 비해, 유니버설 플러그 앤 플레이는 윈도우 2000에서부터 제공하는 IPP(Internet Printing Protocol)와 같은 새로운 네트워크 프로토콜을 사용해 네트워크에 접속된 장치들을 사용하는 것이다. 그리고 어떤 인터페이스(ISA, PCI, VESA, USB 등)든지 상관하지 않고 모두 지원한다. 또한 현재 윈도우에서 제공하는 모든 네트워크 프로토콜을 지원한다.[4]

2.2 썬 마이크로시스템즈의 지니(Jini)

지니는 썬 마이크로시스템즈에서 자바의 분산 객체 기술인 RMI(Remote Method Invoke)를 이용하여 네트워크상의 여러 기기를 연결한 연합체를 형성하고, 이 연합체를 통해 각 제품들 간의 상호 통신이 가능하도록 기본적인 준비를 하게 된다.

지니 시스템에 참여하고 있는 모든 기기들은 각각 하나의 서비스 개념으로 서비스 이용자, 서비스 제공자, 서비스 관리자로 정의된다.[5]

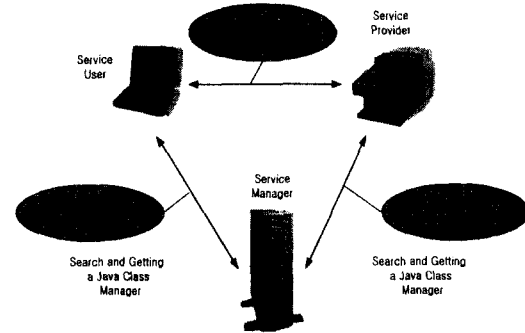


Fig. 2.1 Structure of Jini System and Working theory

Fig. 2.1은 지니 시스템의 구조와 구동 원리를 나타내고 있다. 서비스 이용자가 서비스 관리자에게 서비스 검색을 요청했을 때 서비스 관리자는 서비스 이용자에게 원하는 서비스의 위치를 알려주고 서비스 제공자에게는 어떤 서비스를 이용자가 어떤 요구를 하는지 통지함으로써 서비스 제공자와 서비스 이용자간을 연결하여 주고 서비스 제공자와 서비스 이용자간에는 자바 클래스 통신을 이용한 RMI를 통해 직접 통신을 수행하여 서비스 관리자의 부하를 줄이게 된다.

2.4 OMG의 CORBA

CORBA(Common Object Request Broker Architecture)는 OMG(Object Management Group)에서 분산 객체 컴퓨터 환경의 표준을 정의한 것이다. CORBA는 서로 다른 기종들간의 분산 환경에서 운영체제나 프로그래밍 언어에 관계없이 독립적이고 표준화된 환경을 제공하며 객체 지향 어플리케이션간의 통신을 담당한다. CORBA에서는 IDL(Interface Definition

Langage)라는 표준 언어를 통해 클라이언트와 서버의 구현 객체간의 인터페이스를 정의한다. CORBA의 ORB(Object Request Broker)는 클라이언트와 구현객체간의 중계자 역할을 수행하는 여러 개의 컴포넌트와 인터페이스로 구성되어 분산된 객체들을 클라이언트/서버 관계로 만드는 미들웨어이다.[6, 7, 8, 9]

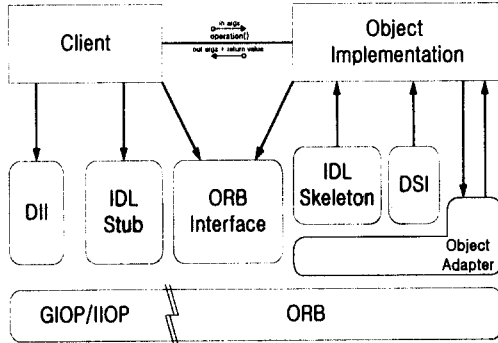


Fig. 2.3 Communication between CORBA Client and Object Implementation

Fig. 2.3은 CORBA의 ORB를 통해 분산 시스템에서의 클라이언트/객체 구현간에 통신의 과정이다. 클라이언트에서 서버를 호출하는 방식은 정적호출방식과 동적호출방식의 두가지를 제공한다. 정적호출방식은 IDL 파일을 통해 생성된 IDL Stub와 IDL Skeleton이 각각 클라이언트와 객체 구현과 링크된다. 동적호출방식은 클라이언트의 실행시간에 ORB함수를 이용하여 클라이언트의 DII와 객체구현의 DSI를 통해 연결된다. 이를 통해 클라이언트의 요청을 서버 객체 구현에 전송하고 객체 구현은 이 요청에 대하여 처리 후 그 결과를 클라이언트에 전송한다. 서비스 요청시 전달되는 정보는 IDL을 통해 생성된 함수와 파라미터의 정보를 가지고 ORB를 통해 수행한다.[10]

ORB는 클라이언트 객체가 구현 객체의 함수를 요청하면 등록된 구현 객체의 정보를 참조하여 구현 객체의 위치를 파악하여 클라이언트가 보내준 서비스 요청 정보를 구현 객체에 전달하고 구현 객체의 함수의 수행이 종료되면 ORB는 그 결과 값을 클라이언트에게 전달한다.

OMG의 CORBA는 소프트웨어적인 버스역할을 수행하는 개념적 표준으로서 Ada, Java, C++, Delphi등의 여러 가지 언어를 지원하여

클라이언트와 객체 구현간의 이종의 프로그램 언어로 구현하더라도 통신이 가능하다.

3 시스템 구현

3.1 시스템의 전체 구조

Fig. 3.1은 미들웨어로서 CORBA를 이용한 3계층 기반의 원격 제어 게이트웨이 시스템의 전체 구조를 나타낸다.

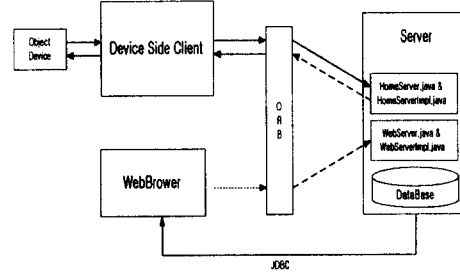


Fig. 3.1 The Structure of Remote Control System using 3-Tier

본 논문에서는 크게 두 개의 ORB와 한 개의 콜백 서비스가 작동된다. 하드웨어 장치와 연결되는 1계층의 장치측 클라이언트와 3계층의 장치측 서버사이의 2계층의 미들웨어로서의 ORB와 1계층의 웹 브라우저 클라이언트와 3계층의 웹서버간의 2계층의 ORB가 구성되었고 웹으로부터의 명령을 장치측 클라이언트에 전송하도록 하기 위해서 장치측 서버와 웹 서버간의 객체간 전송과 장치측 서버에서 장치측 클라이언트의 이벤트를 호출하기 위해 CORBA에서 제공하는 콜백 서비스를 이용하였다. 콜백 서비스란 자신의 서버로부터의 메시지를 포함하기 위해 클라이언트의 이벤트 구조를 확장한 것이다. 이러한 클라이언트와 서버간의 메소드와 변수에 대한 정의는 CORBA의 IDL파일로 정의되어 이루어진다.

3.2 RS-232 시리얼 통신 클래스

원격 제어 게이트웨이를 이용하여 웹으로부터의 명령을 수신받은 장치측 클라이언트에서 대상 장치에 전송하기 위해 썬의 자바에서 제공하는 시리얼 통신 class로부터 상속받은 객체를 선언하여 이용하였다. 하나의 포트를 통해 장치와 데이터를 송수신하기 위해 쓰레드를 구현하였으며 전송속도는 9600bps이다. 여러 대상의 장치를 통합 제어를 수행하기 위해 각 장치

별로 데이터 포맷상에 주소를 할당하였으며 RS-232를 컨버터를 통해 RS-485로 변환이 요구된다.

이ენტ트는 이 DB로부터 각 장치의 정보를 액세스하여 보여준다.

3.3 IDL

다음은 장치측 클라이언트와 서버의 구현객체간의 인터페이스를 정의한 IDL이다.

```
module CorbaSettop{
    interface SettopServer{
        string sendData(in string home_read,
            in CbkObject cbk);
    };
    interface CbkObject{
        string command(in string server_comm);
    };
};
```

모듈 CorbaSettop은 두 개의 인터페이스를 가진다. 첫 번째 인터페이스 SettopServer는 대상 장치로부터의 상태나 에러에 대한 정보를 서버의 구현객체에 전달하는 string형의 home_read와 콜백 객체인 cbk를 전달하는 sendData메소드를 정의하고 두 번째 인터페이스 CbkObject는 웹 인터페이스로부터의 명령어를 클라이언트측의 콜백 객체에 전송하는 command메소드를 정의하였다.

다음은 웹 인터페이스와 서버의 구현객체간의 인터페이스를 정의한 IDL이다.

```
module CorbaWeb{
    interface CorbaWebManager{
        string sendCommand(in string web_comm);
    };
};
```

모듈 CorbaWeb은 웹 서버에 설치된 웹 인터페이스로부터의 원격 명령 정보를 웹 서버 상에 구동되는 구현객체에 string형의 web_comm 파라미터를 통해 전송하는 sendCommand 메소드를 정의하고 있다.

3.4 SQL DB

대상장치에서의 에러 이벤트나 상태 그리고 웹으로부터의 명령에 대한 정보를 기록하기 위해서 일반 파일 형태가 아니라 데이터 베이스에 저장한다. 장치측 서버의 객체 구현에서 JDBC-ODBC 브릿지 형태의 드라이버를 이용하여 SQL DB에 접속하여 기록하고 웹측 클라

3.5 사용자 인터페이스

웹 브라우저를 이용한 사용자 인터페이스는 자바 애플릿과 CORBA를 연동한 Orblet으로 구현하였다. 일반적인 웹환경을 위한 CGI 프로그램은 외부로부터의 요청이 있을 경우 매번 실행되어 결과를 반환하고 종료함으로써 호출간의 상태를 유지하지 못하나 CORBA를 이용한 Orblet은 서버 객체가 계속 존재하여 클라이언트와 서버간의 부하가 적고 네트워크간의 트래픽도 적다.

4. 결론 및 향후 과제

본 논문에서는 기존의 C++ 환경인 클라이언트/서버의 2계층구조 제어시스템을 미들웨어인 CORBA를 이용한 3계층 기반의 원격 제어 게이트웨이를 구현함으로써 기존의 다양한 형태로 존재하는 선박이나 산업현장의 하드웨어 및 소프트웨어간의 이질적인 시스템들을 통합 제어할 수 있는 환경을 제시하였다. 자바와 CORBA를 연동한 Orblet 웹 인터페이스를 구현하여 ORB를 바탕으로 한 객체와 객체 사이의 상호 간의 호출로 인터넷을 통한 통신을 제공하는 다른 환경보다 네트워크의 병목현상을 해결할 수 있었고, 또한 CORBA는 확장성을 가지는 서버와 서버 사이의 객체 통신이 가능한 하부 구조를 제공하여 다양한 서비스를 시간과 공간의 한계를 극복하여 제공할 수 있었다.

앞으로의 연구는 본 논문에서 구현된 3계층 기반의 게이트웨이 시스템을 PC 기반이 아니라 가격 경쟁력을 가지는 단말기 형태의 장치에 바이너리 코드인 자바의 클래스 파일을 임베딩하는데 있어서의 연구 및 인터넷을 통한 원격의 제어를 수행함으로써 발생하는 시간을 지연을 극복하여 엄격한 실시간을 만족시킬 수 있는 알고리즘의 개발과 현재 소프트 실시간의 범위내에서의 적용분야에 대한 연구를 수행하고자 한다.

참고문헌

- [1] "초고속정보 통신망을 이용한 원격 탐색/감지 이동로봇 제어시스템에 관한 연구", 개발보고서, 한양대학교 1997.

- [2] "임베디드 시스템과 임베디드 OS",
정원식, 프로그램세계 May, 2000. pp162-171
- [3] "The Second Generation of Client/Server"
Tim Quinian, Database Programing &
Design, May 1995, pp 31-39
- [4] Microsoft Develop Group, Universal Plug
and Play Start Page: http://msdn.microsoft.com/library/psdk/upnp/upnpport_6zz9.htm
- [5] SUN Microsystem JINI Network
Technology : <http://www.sun.com/JINI/>
- [6] Object Management Group.
<http://www.omg.org>
- [7] Object Management Group. The Common
Object Reques Broker : Architecture and
Specification. Revision 2.2. Object
Management Group, Framingham, MA
1998.
- [8] S. Vinoski, "New Features for CORBA
3.0," CACM, vol.41, no. 10, pp 44-52,
October 1998.
- [9] Object Management Group. CORBA
Messaging Joint Revised Submission.
Object Management Group, Framingham,
MA 1998, document orbos/98-05-05.
- [10] 김석원, "CORBA 구조 및 제품현황", 정보
과학회지, 제 122 호, pp. 5-14, July. 1999.