

# CORBA 기반 원격 서버 관리 시스템의 설계 및 구현

박남섭, 김태운  
고려대학교 컴퓨터학과  
e-mail:nspark@korea.ac.kr

## A Design and Implementation about CORBA-based Remote Server Management System

Nam-Sup Park\*, Tai-Yun Kim\*  
\*Dept of Computer Science, Korea University

### 요약

기존의 원격 서버 관리 시스템은 웹 기반의 클라이언트가 하나의 서버에 접속하는 소켓(socket) 프로 그래밍 형태로 개발되었다. 즉, 시스템 관리자는 웹 환경의 클라이언트에서 하나의 서버 시스템에 접속하여 서버를 관리해야만 했다. 본 논문에서는 다수의 같은 형태의 서버들 혹은 UNIX와 NT 서버와 같은 운영체제가 다른 서버들을 하나의 웹 클라이언트에서 동시에 관리할 수 있는 모델을 제시하고자 한다. 제안하는 시스템에서는 두 서버의 시스템 호출을 중간에서 변환하는 CORBA 기반의 미들웨어를 설계 및 구현하였다. CORBA 기반의 미들웨어 명세(specification)를 결정한 후 그에 따른 서버 측의 서비스들을 정리하여 서버 측에서 개발하였고, 클라이언트 측에서는 서버 측의 서비스들을 호출할 수 있도록 자바 Swing을 이용하여 애플릿 형태로 구현하였다. 그리고, CORBA의 특징을 살려 다수의 서버들을 동적으로 등록시킬 수 있는 기법을 설계하고 구현하였다.

### 1. 서론

웹 기반의 기업 관리 시스템 (WBEM : Web-based Enterprise Management) 은 NMS (Network Management System) 와 SMS (System Management Software) 그리고, EMS (Enterprise Management System) 을 통합한 형태로 발전하고 있다. 이러한 기업 관리 시스템의 통합적인 개발에 있어 최근 기반 기술로 부각되고 있는 기술이 CORBA (Common Object Request Broker Architecture) 이다. 본 연구는 WBEM 시스템의 개발 중 분산 서버 시스템의 통합적인 관리를 CORBA 기반으로 설계 및 구현하도록 한다.

기존의 연구는 단일 서버 시스템 관리에 치중해 있다. 예를 들어 UNIX 운영체제에서의 서버 시스템 관리 혹은 윈도우즈 NT 서버 시스템 관리 등과 같이 각각의 서버 시스템 관리는 서버 관리 시스템의 아키텍처(Architecture) 등이 잘 설계되어 있다. 그

리고 인터페이스도 깔끔하고 이해하기 쉽게 구현되어 있다. 그렇지만 통합적인 형태의 분산 서버 시스템 관리에 대한 연구는 아직도 부족한 편이고 아키텍처도 아직 설계가 부족한 실정이다. 또한 같은 종류의 서버 시스템이라 하더라도 다수의 서버 시스템을 동시에 관리할 수 있는 서버 관리 시스템의 개발은 미미한 실정이다. 본 논문에서는 CORBA 기반 원격 서버 관리 시스템의 설계와 구현에 관한 내용을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 목표 시스템의 설계에 관한 내용을, 3장에서는 목표 시스템의 구현 및 인터페이스에 관한 내용을 4장에서는 성능 평가를 마지막으로 5장에서 결론을 제시한다.

### 2. 목표 시스템의 설계

CORBA 기반 원격 서버 관리 시스템의 전체적인 구조는 그림 1에서 보는 바와 같다. 그림에서 보는 것처럼 클라이언트에서는 웹 브라우저를 통해 다수의 서버들이 위치하고 있는 CORBA 도메인 중 IIOP (Internet Inter-ORB Protocol) 서버로 접속하도록 한다. IIOP 서버에서는 다른 서버들 예를 들어 NT 서버, UNIX 서버 등과 같은 서버들의 도메인 내에서의 등록 상태를 확인하여 서버들의 리스트를 보유하고 있다.

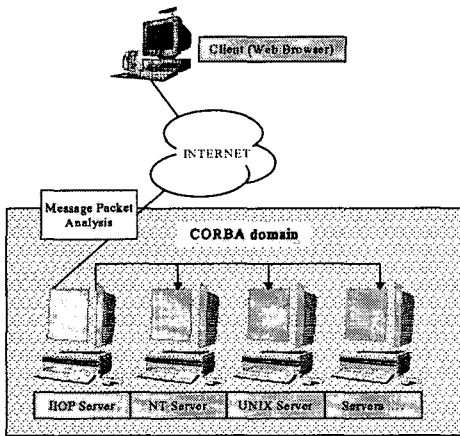


그림 1 CORBA 기반 원격 서버 관리 시스템 모델

클라이언트에서는 특정 서버에 대한 원격 관리 및 조회 명령을 메시지 패킷 (Message Packet) 형태로 IIOP 서버에게 전송한다. IIOP 서버는 클라이언트에서 전송된 메시지 패킷을 분석하여 어떤 서버의 어떤 관리 메시지인가를 확인한 후 해당 서버를 호출한 후 메시지를 수행하도록 한다. 해당 서버에서 클라이언트의 메시지가 수행되고 나면 그 결과는 다시 IIOP 서버로 보내지고 IIOP 서버에서는 결과를 클라이언트로 전송한다. 이런 형태로 CORBA 도메인 내의 모든 서버들을 동일한 인터페이스에서 원격 관리가 가능하도록 처리한다.

### 3. 시스템 구현

CORBA 기반 원격 서버 관리 시스템의 사용자 인터페이스는 그림 2에서 보는 바와 같다. 클라이언트의 요청에 동작할 수 있도록 CORBA 도메인 내의 특정 서버를 IIOP 서버로 선택하고 웹서비스를 지원하도록 한다. 그림 2의 상단 그림은 접속시의 홈페이지가 되고 하단 그림은 홈페이지를 클릭 했을

때 동작하는 자바 애플릿으로 Swing을 이용하여 구현하였다.

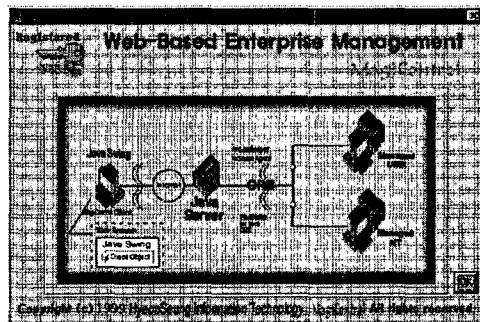
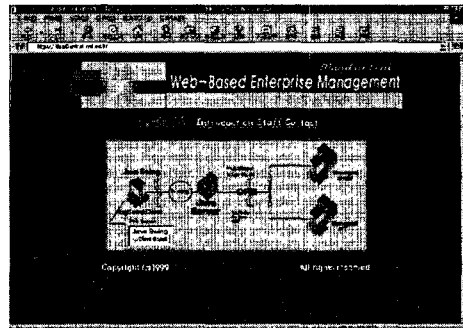


그림 2 사용자 인터페이스

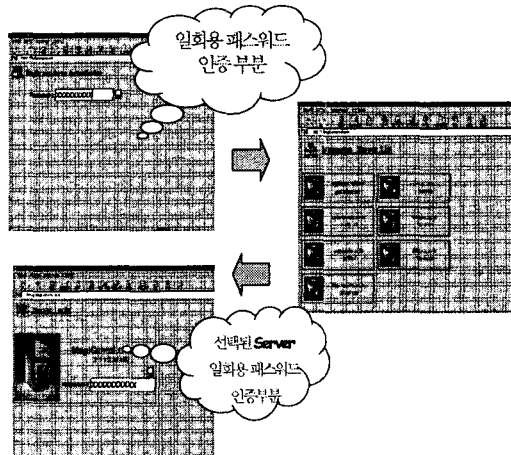


그림 3 일회용 패스워드 인증 모듈

자바 애플릿 프레임이 활성화된 후 해당 서버의 선택과 서비스의 요청은 보안상의 문제 때문에 그림 3에서 보이는 것처럼 일회용 패스워드 인증 모듈을 구현하였다. 클라이언트에서는 패스워드 파일들의 리스트가 있는 일회용 패스워드 인증 디스켓을 항상

가지고 있도록 하였다. 디스켓이 없을 경우는 서버로의 접속이 안 되도록 처리하였다. 그 디스켓을 이용하여 서버에 접속하면 그림 3에서 보이는 패스워드 인증 절차 홈페이지가 보인다. 인증 절차는 IOP 서버의 패스워드 인증 후 해당 서버의 패스워드 인증 절차를 거치도록 하였다.

인증 절차를 거치면 해당 서버 시스템의 관리 서비스를 이용할 수 있다. 서버 시스템마다 특징이 다른 면들이 있지만 서버 관리시 비중이 큰 서비스들을 구분하여 그림 4에서 제시하는 서버 시스템의 관리 서비스 그룹을 만들었다.

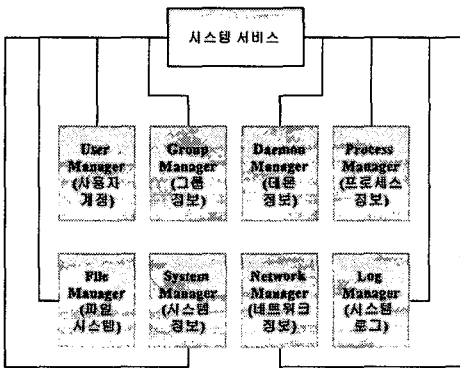


그림 4 서버 시스템 관리 서비스

User Manager는 사용자의 계정 등록, 삭제, 갱신 등의 서비스를, Group Manager는 사용자들이 속해 있는 그룹의 등록, 삭제, 갱신 등의 서비스를 담당한다. Daemon Manager는 시스템에서 서비스하고 있는 각종 데몬들의 서비스 시작, 종료, 정보 확인 등의 서비스를, Process Manager는 시스템에서 서비스되고 있는 프로세스들의 서비스 시작, 종료, 정보 확인 등의 서비스를 담당한다. File Manager는 시스템에 있는 파일 시스템의 관리를, System Manager는 해당 서버 시스템이 갖는 로그내용의 조회를 담당한다. Network Manager는 해당 시스템 네트워크의 정보를 확인해 주고, Log Manager는 IOP 서버에서 동작하는 서비스로서 목표 시스템 자체적으로 갖는 로그를 확인해 준다.

해당 서비스에 대한 실질적인 IDL언어 명세는 그림6에서 보는 바와 같다.

```

module MgrControl {
  module manager {
    interface UserManager {
      boolean add(in string name, in short UID, in short GID,
                 in string homeDir, in string shell, in string fullTime);
      boolean delete(in short UID);
      boolean modify(in short UID, in string homeDir, in string shell, in string newLogin,
                    in string newHome, in string oldHome);
      string getUidList();
    }

    interface GroupManager {
      boolean add(in string name, in short GID);
      boolean delete(in short GID);
      boolean modify(in short GID, in short newGID, in string newName, in string oldName);
      string getGroupList();
    }

    interface DaemonManager {
      boolean start(in string name);
      boolean stop(in string name);
      boolean restart(in string name);
      boolean pause(in string name);
      string getDaemonList();
    }

    interface ProcessManager {
      string getProcessList();
      boolean kill(in short PID);
    }
  }
}
    
```

```

module MgrControl {
  module manager {
    interface FileManager {
      string getFsList(in string fileSystem);
      string getDirList(in string fileSystem);
      boolean delete(in string fileSystem);
      boolean copy(in string srcFsPath, in string destFsPath);
      boolean move(in string srcFsPath, in string destFsPath);
      boolean rename(in string srcFsPath, in string destFsPath);
      boolean unmount();
      boolean mount(in string fileSystem);

      string getFs(in string fileSystem);
      //boolean save(in string fileSystem);
      boolean save(in string fileSystem, in string fileContent);
    }

    interface NetManager {
      string getNetLog();
    }

    interface SystemManager {
      string getSystemInfo();
    }

    interface NetworkManager {
      string getIP();
      string getGateway();
      string getHostFile();
      string getHost();
      string getDNS();
      string getNetMask();
      string getNetCard();
    }
  }
}
    
```

그림 5 IDL Specification

패스워드 인증 절차를 거친 사용자는 그림 6에서 보는 바와 같은 자바 프레임 환경 하에서 시스템 관리 작업을 진행할 수 있다. 그림 6에서는 해당 서버 시스템의 데몬 서비스를 지원하고 있는 인터페이스를 보여 준다.

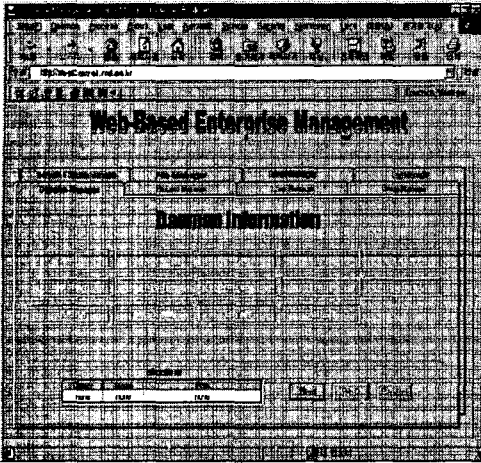


그림 6 Damon Manager 시스템 관리 서비스

#### 4. 성능 평가

본 논문에서는 CORBA 기반 원격 서버 관리 시스템의 설계 및 구현에 관한 내용을 살펴보았다. 클라이언트-서버 모델은 아직도 많이 쓰여지고 있는 업계 표준이지만 확장성, 유지보수, 안정성 등에서 많은 문제점을 안고 있다고 볼 수 있다. 현재 CORBA에서 제시하는 향후 기업 환경의 표준은 그런 면에서 시기 적절하다고 볼 수 있다. 본 논문에서는 CORBA를 기반으로 원격에서 다른 플랫폼에서 동작하는 다수 서버들을 관리할 수 있는 방법적 모델을 제시한 것으로 현재의 서버 시스템의 운영 환경을 제고하고 새로운 모델의 운영 시스템을 설계하고 구현하였다. 그리고 분산 객체를 이용함으로써 서버 부하를 최소화하고 프로그램 재사용성을 증대시키는 효과를 가진다.

서버측 개발시 도출된 문제점은 UNIX 와 NT 시스템의 관리 서비스를 통일하는 것이 부분적으로 잘 맞지 않는 부분이 있다는 점이다. 설계나 구현 면에서 각각의 운영체제에 맞는 독자적인 모델의 개발 또한 좀더 세부적으로 고려되어야 할 것이다.

#### 5. 결론 및 향후 연구 과제

본 논문에서는 시스템 관리자의 시스템 관리 모델에 대한 새로운 방법을 제시하고 있다. 다수의 다종 서버 시스템들을 CORBA 도메인에서 일관된 사용자 인터페이스로 시스템을 관리하도록 하는 모델을 제시 및 구현했다.

향후 연구 과제로 고려되는 것은 서버 관리 시스템의 실시간성의 보장이라는 문제다. 또한 고려될 사항은 현재는 DCOM의 개발이 미미한 실정인데 이 부분과 CORBA가 좀더 완벽하게 동작할 수 있도록 미들웨어 및 클라이언트, 서버측 프로그램을 개발하는 것이다.

#### 참고문헌

- [1] Robert Orfali, Dan Harkey, "Client/Server Programming with Java and CORBA"-2nd, 1998.
- [2] Pedrick, Weedon, Goldberg, Bleifield, "Programming with VisiBroker", 1998.
- [2] David Flanagan, "JAVA Examples in a nutshell", Brian Maso, 1997.
- [3] Robert Orfali, Dan Harkey, Jeri Edwards, "Instant CORBA", 1997.
- [4] 전병선, "Inside ATL/COM Programming with Visual C++", 1998. 6.
- [5] Microsoft PRESS, "Inside Distributed COM", 1998.