

# 자바 서블릿을 이용한 인터넷 웹 기반의 3-Tier 데이터베이스 시스템 구축

°곽재홍, 곽정필, 최재원  
경성대학교 전기전자·컴퓨터공학부

Jae-Hong Kwak, Jung-Pill Kwak, Jae-weon Choe

Department of Computer Engineering, Kyung Sung University

E-mail : choejw@star.kyung Sung.ac.kr

## 요 약

과거 PC 통신 환경 상에서 제공되었던 텍스트 위주의 명령에 기초한 데이터베이스 서비스를 사용자의 정보 접근이 용이한 멀티미디어 웹에 기초한 데이터베이스 서비스로 전환이 이루어지도록 하고자 한다. 따라서 사용자는 인터넷에 기반을 둔 웹서비스를 통해 통합 데이터베이스로부터 유용한 정보를 빠르고 용이하게 접근 가능하도록 하고, 정보 제공자는 기존 정보의 사장을 막고 새로운 정보의 추가와 시스템 확장을 용이할 수 있는 플랫폼을 가질 수 있다. 현재 서비스 중인 타 IP 제공자들이 GUI환경의 웹서비스로 전환하는 하기 위한 모델링의 역할을 할 수 있으며, Java 서블릿을 이용한 3-Tier 시스템의 구현으로 기존의 데이터베이스 뿐만 아니라 타 데이터베이스까지 수용할 수 있는 데이터베이스 뱅크 역할을 할 수 있다.

## 1. 서 론

인터넷은 상당히 오랜 시간동안 전문가의 전 유물로 인식되어 왔으나 인터넷과 하이퍼텍스트를 이용한 멀티미디어 웹 기술의 만남은 전문가 뿐만 아니라 비전문가의 폭발적인 관심을 끌기에 충분하였다. 인터넷은 전 세계 컴퓨터와 이용자를 한 데 묶어 다양한 서비스의 토대가 되는 통신 인프라를 제공하였고, 멀티미디어 웹 기술은 다양한 멀티미디어 서비스를 비전문가도 손쉽게 이용 가능하도록 네트워크 화하였다.

기존의 공중전화망(PSTN)을 기반으로 한 PC 통신 서비스가 최근 몇 년 사이에 인터넷에 기반을 둔 멀티미디어 웹 서비스로 빠르게 대체되고 있는 실정이다. 이로 인해 과거 많은 인력과 자금을 투자하여 개발한 PC 통신 환경 상의 많은 서비스가 사용자로부터 도외시되어 사장되는 문제가 발생하고 있다. 그래서 본 연구에서는 과거 PC 통신 환경 상에서 제공되었던 텍스트 위주의 명령에 기초한 데이터베이스 서비스를 사

용자의 정보 접근이 용이한 멀티미디어 웹에 기초한 데이터베이스 서비스로 전환이 이루어지도록 하여 과거 많은 인력과 자금이 투자되어 개발된 유용한 데이터베이스와 각종 서비스가 사장되는 것을 막도록 하고자 한다.

본 연구의 대상이 되는 PC 통신 기반의 데이터베이스 시스템은 현재 하이텔의 인포샵 서비스를 통해 해외유학정보, 지역의료정보, 지역생활정보 등을 제공하여 일반 대중의 정보교류와 복리증진 촉진에 일익을 담당하고 있는 데이터베이스 시스템으로 HP9000/H70 워크스테이션에 Sybase 10.01 DBMS를 근간으로 구축되어 있다.

본 연구의 궁극적인 목표는 첫째, 텍스트 기반의 명령에 의해 제공받던 정보 서비스를 GUI 기반의 인터넷 웹에 의해 제공받을 수 있도록 하여 사용자의 정보 접근과 관리자의 유지·보수를 용이하게 하고자 한다. 둘째, 복수의 사용자가 동시에 접속하여 서비스를 요청하더라도 사용자의 서비스 응답속도가 저하되지 않도록 빠르고 효율

적인 수행이 이루어지도록 하고자 한다. 셋째, 향후 시스템의 확장을 고려하여 구축되어 있는 자체 데이터베이스는 물론이고 다른 시스템의 데이터베이스도 접근하여 서비스 할 수 있는 데이터베이스 뱅크 역할을 수행할 수 있도록 하고자 한다. 넷째, 프로그램의 중복을 최소화하고 프로그램의 재사용을 극대화할 수 있도록 하여 프로그램의 개발과 추후 프로그램의 유지보수가 용이하도록 하고자 한다. 즉, 본 연구에 의해 사용자는 인터넷에 기반을 둔 웹서비스를 통해 통합 데이터베이스로부터 유용한 정보를 빠르고 용이하게 접근 가능하도록 하고, 정보 제공자는 기존 정보의 사장을 막고 새로운 정보의 추가와 시스템 확장을 용이할 수 있는 플랫폼을 가질 수 있다.

본 논문의 2장에서는 시스템의 동작을 기술하고, 3장에서는 시스템의 설계 및 구현 방법을 기술한다. 4장에서는 시스템 구현을 위하여 데이터베이스 제어 프로그램과 조회결과 출력 프로그램과 상세결과 출력 프로그램을 기술하고, 결론으로 결론을 기술하였다.

## II. 시스템 동작

기존의 IP를 통해 정보를 제공받던 사용자들이 GUI환경에서 쉽게 정보를 접근하기 위해서는 웹서버를 미들웨어에 구축하여 서블릿을 동작하게 하고 인터넷 사용자들은 이 서버를 경유하여 기존의 정보를 동일한 형태로 얻게 하였다.

미들웨어에는 아파치 웹서와 자바를 사용하기 위한 JDK를 설치하였으며, 서블릿을 사용하기 위하여 JDSK와 JServ가 동작한다. 즉 사용자들은 인터넷을 경유하여 아파치와 자바와 서블릿의 환경구축이 되어진 미들웨어를 경유하여 데이터베이스 서버로 접근하는 3-Tier 시스템이다.

인터넷 서버를 두어 서블릿이 동작하는 구조는 그림 1과 같다.

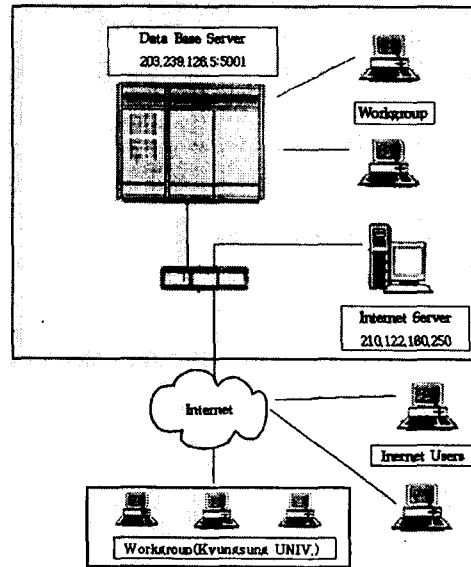


그림 1. 인터넷 환경에서의 시스템 동작

2-Tier방식은 java applet, application이 데이터베이스로 바로 접근하는 방식을 말한다. 2-Tier 방식의 시스템 동작구조는 그림 2와 같다

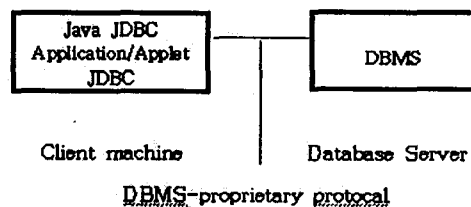


그림 2. 2-Tier 방식의 시스템 동작구조

여기서 데이터베이스 서버는 반드시 웹서버와 같은 곳에 있을 필요는 없다. 하지만 applet을 사용할 경우에는 자바의 보안에 위배되기 때문에 문제가 있다. 전형적인 클라이언트/서버 시스템들은 2-tier 아키텍처로 구성된 카테고리를 가지고 있다. 애플리케이션은 모두 각 클라이언트 PC에 존재하고, 다른 곳에 있는 서버에 데이터베이스를 놓는다. 2-Tier 애플리케이션은 작업처리 부하가 PC에 많이 주어지고, 상대적으로 기능이 강력한 서버는 단지 애플리케이션과 데이터베이스 사이의 트래픽 제어기 역할만 수행하게 된다. 결과적

으로 애플리케이션 수행 속도는 PC의 제한된 자원 때문에 시달리게 되고, 네트워크 트래픽은 점점 증가하게 된다.

모든 애플리케이션을 PC에서 실행하는 경우, 애플리케이션은 사용자가 모든 것을 다 보지 못했더라도 데이터에 대해 여러 개의 요청을 만들어내야 한다. 이런 중첩된 데이터베이스 요청은 네트워크에 심각한 부담을 주게 된다.

2-Tier 접근의 다른 문제점은 유지 보수성이다. 애플리케이션에 아무리 사소한 변경이 가해져도 전체 사용자에게 다시 다운로드해야 한다. 이 과정을 몇번 거치고 나면, 어떤 버전이 어디에 존재하는지 관리하기가 매우 어려워질 수 있다. 이런 문제점들을 해결하기 3-Tier 아키텍처 개념을 도입하였다.

2-Tier 방식과는 달리 middle-ware가 도입된 개념으로 일반 자바 applet이나 application은 HTTP, RMI, CORBA등을 사용하여 서버에 있는 자바 Application이나 객체에게 데이터베이스 처리를 요청한다. 이때 서버의 자바 Application은 JDBC Driver를 통해 데이터베이스를 접근한다. 3-Tier 방식의 시스템 동작구조는 그림 3와 같다.

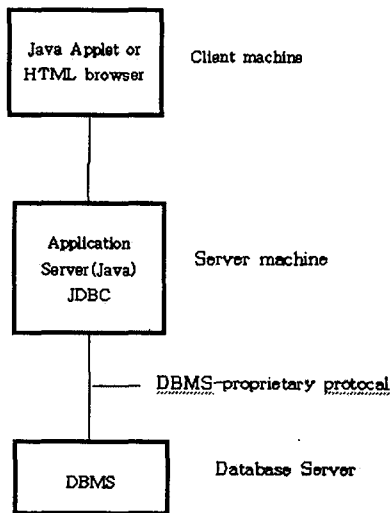


그림 3. 3-Tier 방식의 시스템 동작구조

3-Tier 아키텍처를 사용하면 수행속도, 네트워크 트래픽, 유지보수 문제가 모두 해결된다.

### III. 시스템의 설계 및 구현방법

CGI기반 시스템의 경우 사용자의 CGI 요구가 많아질수록 자식(child) 프로세스의 생성으로 인한 오버헤드로 인해 시스템의 자원이 소비되므로 실행 속도 및 시스템 전체의 성능이 저하되는 문제점이 있다. 향상된 CGI나 기타 방법들도 각각 기존 CGI의 문제점을 충분히 해결하지 못하며, 또한 CGI를 사용한 데이터베이스 접속의 경우 시스템의 응답 속도가 서블릿과 JDBC를 사용한 시스템의 경우보다 떨어지므로 본 인터넷 웹기반의 유확정보 데이터베이스 시스템 구축을 위해서 본 논문에서 제안하는 시스템의 설계 및 구현 방법은 다음과 같다.

- (1) Text mode에서 GUI 환경의 인터넷 웹기반의 데이터 베이스 접근 시스템을 구현하도록 한다.
- (2) 복수의 사용자가 접속하여도 시스템의 실행 속도가 느려지지 않도록 하기 위해 기존 CGI의 단점을 보완한 서블릿과 JDBC를 사용하여 데이터베이스 시스템에 접근하는 시스템을 구현한다.
- (3) DBMS와 서블릿 서버를 따로 두는 3-tier 구조로서 시스템을 구현한다.
- (4) 기존의 서비스 중인 데이터베이스의 스키마를 분석하여 데이터베이스에 접근할시 Client는 파라메타 전달방법으로서 각 요구 정보를 얻을 수 있게 하고, 프로그램의 중복화를 최소화하여 차후 프로그램 유지 보수하는데 전체의 프로그램을 수정하지 않아도 될 수 있도록 구현한다.

#### IV 시스템 구현

본 장에서는 자바 서블릿을 이용한 인터넷 웹 기반의 대용량 데이터베이스 시스템 구축 연구 목적 및 목표에서 제시되었던 기존의 국내통신망에서만 텍스트 형태로 제공되던 시스템을 GUI 환경의 웹상에서 누구나 정보를 쉽게 접근 할 수 있게 하였다. 실행속도를 향상시키고 오버헤드를 감소시키기 위해 서블릿을 사용하였으며, 프로세

스 관리와 파일의 유지 보수를 위하여 파라메타 전달방식으로 구현하였다. 프로그래밍은 공통 루틴으로서 데이터베이스에 접근할 수 있게 하는 데이터베이스 제어 프로그램, 항목과 상세내용을 보여줄 수 있게 하는 항목과 내용조회 출력 프로그램 파일이 있으며, 각 서비스 별로 항목과 내용을 조회할수 있는 애플리케이션이 동작한다.

##### 4.1 데이터베이스 제어프로그램

ConnectionDB.java는 Client가 각 해당정보를 요구할 때 데이터베이스와의 제어를 담당한다. 어플리케이션은 한 개의 데이터베이스와 하나 이상의 연결을 할수 있거나 또는 많은 서로 다른 데이터베이스에 연결할 수도 있다.

JDBC를 드라이버를 통하여 시용자와 패스워드를 얻어 데이터베이스에 접근 가능하게 한다. 그림 4는 데이터베이스 제어하기 위한 서블릿 프로그램이다.

```

public class ConnectionDB
{
    static String      _dburl = "jdbc:sybase:Tds:203.239.126.5:5001"; //DB가 있는 곳의 주소
    static String      _user = "gong";                               //사용자 ID
    static String      _password = "gonggong";                       // Password
    static String      _charSet;
    static String driverName = "com.sybase.jdbc2.jdbc.SybDriver";

    /* DB 접속을 위한 함수 */
    public void connectDB() throws ClassNotFoundException, SQLException, Exception {
        try { /* JDBC 드라이버를 로드하고, 드라이브 매니저에 등록한다. */
            Class.forName(driverName); /* 데이터베이스에 접속한다. */
            cont = DriverManager.getConnection(_dburl, _user, _password);
            /* SQL 명령 실행을 위한 Statement를 얻는다. */
            stmt = cont.createStatement();
        } catch (ClassNotFoundException ex) {
            ... //예외상황을 처리하기 위한 catch문
        }
    }

    /* DB 접속 해제를 위한 함수 */
    public void disconnectDB() throws SQLException {
        try {
            stmt.close();
            cont.close();
        } catch (SQLException ex) { //예외상황을 처리하기 위한 catch문
            throw ex;
        }
    }
}

```

그림 4. 데이터베이스 제어프로그램

## 4.2 항목조회 출력 프로그램

ABR\_ListTable.java는 ABR\_31\_A.java와 같은 프로그램이다. 그림 5에 ABR\_ListTable.java 의 각 애플리케이션에 의해 Database에서 조회된 결과 프로그램을 나타내었다. 과를 화면에 출력해 주기 위한 공통 테이블 작성

```
public class ABR_ListTable
{
    PrintWriter out ;          // 출력을 위한 객체
    /* 리스트 테이블의 생성자 함수 */
    public ABR_ListTable(PrintWriter out) { this.out = out; }
    /* 리스트 테이블의 헤드 출력을 위한 함수 */
    public void tableHead(String h1, String h2) {
        out.println("<table height=50 align=center width=100%
        cellpadding=0 cellspacing=0>" + "<tr bgcolor=#CCCCFF>" +
        "<td width=14%>" + h1 + "</td>" + "<td width= * >" + h2 + "</td>"
        + "</tr>");
    }

    /* 리스트 테이블의 내용 출력을 위한 함수 */
    public void tableContent(int c1, String c2, String link) {
        out.println("<tr bgcolor=#EAEAD5 bordercolor=#E0E0E0>"
        + "<td width=14%> <a href=" + link + ">" + c1 + "</a></td>"
        + "<td width= * >" + c2 + "</td>" + "</tr>");
    }

    /* 리스트 테이블의 꼬리 출력을 위한 함수 #1 */
    public void tableTail() {
        out.println("</table>");
    }

    /* 리스트 테이블의 꼬리 출력을 위한 함수 #2 */
    public void tableTail(int total, int current, String prelink, String nextlink) {
        out.println("<tr><td colspan=1><div align=center>"
        + "<a href=" + prelink + "> [이전]</a>" + "<a href=" + nextlink + ">
        [다음]</a>" + "</td>" + "<td colspan=2> Page NO : "+ current + "/" +
        total + "</div></td>" + "</tr></table>");
    }
}
}
```

그림 5. 항목조회 출력 프로그램

### 4.3 내용조회 출력 프로그램

ABR\_ResultTable.java는 List 애플리케이션에 공통 테이블 작성 프로그램이다. 그림 6에 의하여 출력되어진 리스트 테이블에서 선택된 상 ABR\_ResultTable.java 의 프로그램을 나타내었다. 세 정보를 조회하여 화면에 출력해 주기 위한

```

import java.io.*;
public class ABR_ResultTable
{
    PrintWriter out;          // 출력을 위한 객체

    /* 결과테이블의 생성자 함수 */
    public ABR_ResultTable(PrintWriter out) { this.out = out; }

    /* 결과테이블의 헤드 출력을 위한 함수 */
    public void resultTablehead(String h1, String h2) {
        out.println("<table width=69% border=0 align=center
            bordercolor=#CCFFCC bgcolor=#FFFFCC>"
            + "<tr><td width=16% bgcolor=#CCCCFF>번호</td>"
            + "<td width=30% bgcolor=#CCFFCC>" + h1 + "</td>"
            + "<td width=10% bgcolor=#CCCCFF>제목</td>"
            + "<td width=44% bgcolor=#CCFFCC>" + h2 + "</td></tr>"
            + "<tr bgcolor=#CCCCFF><td colspan=4>[정보]</td></tr>"); }

    /* 결과테이블의 내용 출력을 위한 함수 */
    public void resultTableContent(String c1) {
        out.println("<tr bgcolor=#FFFFFF><td colspan=4 height=300>"
            + "<pre>" + c1 + "</pre>" + "</td></tr>");
    }

    /* 결과테이블의 꼬리 출력을 위한 함수 */
    public void resultTableTail() {
        out.println("</table>");
    }
}

```

그림 6. 내용조회 출력 프로그램

#### 4.4 항목조회 프로그램

ABR\_31\_A.java와 같은 파일 명명법에 의한 각 애플리케이션은 각 해당 정보에 대한 List를 출력 하는 프로그램이다. 각 해당 정보를 얻기 위해서 데이터베이스에 쿼리를 보내는데 각

입력된 파라메타 전달에 따라 내용을 조회하게 된다. 그림 7에 ABR\_31\_A.java의 프로그램을 나타내었다.

```
public class ABR_31_A extends HttpServlet
{
    String currentProgram = "ABR_31_A";           // 현재 수행중인 프로그램
    String nextProgram = "ABR_31_R";           // 다음에 수행될 프로그램
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        . . .

        printPageHead(out, "해외유학정보/국가별 학교안내/대학원 안내");
        /* HTML문서의 시작 */ { /* 서브릿 프로그램의 입력 파라메타를 구한다. */
        n_code = req.getParameter("n_code");           // Nation Code
        r_code1 = req.getParameter("r_code1");         // Region Code1
        r_code2 = req.getParameter("r_code2");         // Region Code2
        . . .

        rs = conDB.stmt.executeQuery ("select SERIAL_NO, NAME from
        ABR_DB.SCHOOL_GRAD " + "where NATION_CODE = '" + n_code + "'
        and REGION_CODE1 = '" + r_code1 + "' and REGION_CODE2 = '" + r_code2 +
        "' Order By NAME asc");
        /* 리스트 테이블에 조회된 내용을 출력한다. */
        while(rs.next() ) { number++; listTable.tableContent(number,
            rs.getString(2), nextProgram + "?number=" + number +
            "&serial_no=" + rs.getString(1));
        }
        . . .

        /* 리스트 테이블의 꼬리를 출력 */
        listTable.tableTail();
        printPageTail(out);
        // HTML 문서의 종료
    }
}
```

그림 7. 항목조회 프로그램

#### 4.5 내용조회 프로그램

ABR\_31\_R.java와 같은 파일 명명법에 의한 각 애플리 케이스는 List 프로그래밍에 의한 정보중에서 상세정보를 원할 경우 각 해당 정보에 대한 상세정보를 출력하는 프로그램이다. 그림 8에 ABR\_31\_R.java의 프로그램을 나타내었다.

```

public class ABR_31_R extends HttpServlet
{
    String currentProgram = "ABR_31_R";           // 현재 수행중인 프로그램
    String nextProgram = "";                       // 다음에 수행될 프로그램

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        printPageHead(out, "해의유학정보/국가별 학교안내/대학원 안내");
        /* HTML문서의 시작 */
        /* 서브릿 프로그램의 입력 파라메타를 구한다. */
        number = req.getParameter("number");      // 리스트 테이블 내의 순번
        serial_no = req.getParameter("serial_no"); // DB 내의 대학 일련번호

        /* ABR_DB의 SCHOOL_GRAD 테이블로 부터 요청된 내용을 조회한다. */
        rs = conDB.stmt.executeQuery("Select SERIAL_NO,TEXT_LCNT, NAME,
        SCHOOL_DESC" + "from ABR_DB..SCHOOL_GRAD " + "where SERIAL_NO =
        " + serial_no);

        /* 결과테이블을 출력한다. */
        while(rs.next()) {
            /* 결과테이블의 헤드를 출력한다. */
            ResultTable.resultTablehead(number, rs.getString(3));
            /*결과테이블에 조회된 내용을 출력한다. */
            ResultTable.resultTableContent
                (rs.getString(4));
        }

        /* 결과테이블의 꼬리를 출력 */
        ResultTable.resultTableTail();
        printPageTail(out);
        // HTML 문서의 종료
    }
}

```

그림 8. 내용조회 프로그램



## V. 결론

현재 많은 서비스 중인 IP는 불과 몇 년전에 힘들게 많은 인력과 투자로 하여 만들어졌으나 현재 개선되지 않고 있는 서비스로 인하여 아직도 텍스트 위주의 명령을 기초로 서비스되어 사용자들의 정보 접근이 힘들고 DB가 사장되는 결과를 초래한다. GUI 환경으로 변환하게 되므로 유용한 정보를 쉽게 접근하게 되었다. IP 제공자의 측면에서는 기존의 DB를 활용할 수 있도록 하며, 추가적인 정보의 유지보수를 할 수 있다.

인터넷 웹기반의 유학정보 데이터베이스 시스템 구축 방법에 있어서 객체지향언어이며 플랫폼 독립적 언어인 자바를 사용하여 전체 시스템을 구성하였으며 응답 속도가 느리고 오버헤드가 큰 CGI의 대안으로 Java 서블릿을 사용하였다.

Java 서블릿의 경우 스레드를 지원하므로 매 사용자의 요구에 대해 프로세스를 생성시키지 않고 스레드로 처리하여 서버의 부하가 감소되므로 사용자의 접속회수가 증가하여도 시스템 응답 속도의 저하를 줄일 수 있게 된다.

CGI를 통한 웹과 데이터베이스의 연동 시 발생하는 속도의 저하 및 프로그램의 복잡성, 데이터베이스 종속성을 개선하기 위해 JDBC API를 사용하였다.

파라메타 전달방법으로 인하여 프로그램의 수를 줄임으로서 프로세스의 증가를 막아 서버의 오버헤드를 줄이게 되게 하였다.

본 논문의 주제인 인터넷 웹기반의 유학정보 데이터베이스 시스템 구축함으로써 기존의 텍스트 형태로 서비스중인 데이터베이스들의 웹 환경으로 전환하는 모델링이 될 것이며, 차후 연구 과제로는 현재 유학정보 뿐만 아니라 5개의 데이터베이스 모두 웹 환경 시스템으로 구현할 것이며, 관리자 모드를 만들어 시간과 공간의 제약없이 데이터베이스를 유지 보수 할수 있게 할 것이다.

## 참고문헌

- [1] 송주영, "Internet과 초고속정보통신망", 한국정보과학회학회지, 제14권, 제11호, 1996
- [2] 정기상, 김대인, 황부현 "이기종 데이터베이스에서 서블릿과 JDBC를 이용한 상품 검색", Proceeding, 한국정보처리학회지, 04 v.6, n.1, 1999
- [3] 유은민, 서범수, 김영국, 정석찬, "웹과 데이터베이스 연동기법분석", Proceeding, 한국정보처리학회지, 04 v.6, n.1, 1999
- [4] L. Orozco-Barbosa, "Real-Time Delivery of Multimedia Documents over DQDB MANs", Proc. of ICCCN, 1994.
- [5] Hunter & Crawford, "Servlet Programming", 1998
- [6] Bruce Eckel, "Thinking in Java", 1998IP 통합 정보망 구축