

PC 통신 DB 서비스의 인터넷 웹 DB 서비스로의 효율적 전환을 위한 구현방법 및 기술

정회원 최재원*

Implementation Methods and Techniques for Efficient Updates from Command-based to Web-based Database Services

Jae-Weon Choe* *Regular Member*

요 약

기존의 PC 통신 서비스가 최근 인터넷에 기반을 둔 멀티미디어 웹 서비스로 빠르게 대체됨으로 인해 과거 많은 인력과 자금을 투자하여 개발한 유용한 데이터베이스와 PC 통신 환경 상의 많은 서비스가 사용자로부터 도외시되어 사장되는 문제가 발생하고 있다. 이에 본 연구에서는 과거 PC 통신 환경 상에서 제공되었던 텍스트 위주의 명령에 기초한 데이터베이스 서비스를 사용자의 정보 접근이 용이한 멀티미디어 웹에 기초한 데이터베이스 서비스로의 효율적 전환을 위한 구현방법 및 기술에 대해 연구하였다.

ABSTRACT

Conventional command-based communication services are rapidly substituted by multimedia web-based communication services at most recent times. As the result, many valuable database and services developed by investing lots of human and material resources become to be useless. Therefore in this paper we researched the implementation methods and techniques for making the updates efficiently from the conventional command-based database services to a multimedia web-based database system that made users easily access to database and services.

1. 서론

인터넷은 상당히 오랜 시간동안 전문가의 전유물로 인식되어 왔으나 인터넷과 하이퍼텍스트를 이용한 멀티미디어 웹 기술의 만남은 전문가뿐만 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 통

신 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 서비스와 유용한 데이터베이스가 사용자로부터 도외시되어 사장되는 문제가 발생하고 있다.^[1] 이러한 문제는 기존의 PC 통신 환경에서 정보제공(IP) 사업을 하였던 수많은 기업의 당면과제가 되었다. 그러나 기존의 PC 통신서비스를 인터넷 웹서비스로의 효율적 전환을 위한 구현방법과 기술에 대한 연구결과나 논문이 없어 기업 나름의 노하우와 기술 수준에 따라 형태대로 시스템 전환을 수행하고 있는 실정이다.

* 경성대학교 컴퓨터공학과

논문번호: 020127-0319, 접수일자: 2002년 3월 19일

* 이 논문은 2002학년도 경성대학교 특별과제연구비에 의하여 연구되었음

그래서 본 연구에서는 과거 PC 통신 환경 상에서 제공되었던 텍스트 위주의 명령에 기초한 데이터베이스 서비스를 사용자의 정보 접근이 용이한 멀티미디어 웹에 기초한 데이터베이스 서비스로의 효율적 전환을 위한 구현방법 및 기술에 대해 연구하였다. 본 연구는 기존의 PC 통신서비스를 인터넷 웹서비스로의 효율적인 시스템 전환을 원하는 IP 사업자에 있어 참조기술로서 그 활용도가 높을 것이라 생각된다.

본 연구의 대상이 되는 PC 통신 기반의 데이터베이스 시스템은 HP9000/H70 워크스테이션에 Sybase 10.01 DBMS를 근간으로 구축되어 있으며, 현재 하이텔의 인포샵 서비스를 통해 해외유학정보, 지역의료정보, 지역생활정보 등을 제공하고 있다.^[2]

본 연구의 궁극적인 목표는 기존의 PC 통신 기반의 데이터베이스 서비스를 최소 비용으로 인터넷 웹 기반의 데이터베이스 서비스로 효과적인 전환이 이루어지도록 하여, 사용자는 유용한 정보를 빠르고 용이하게 접근할 수 있고, 정보 제공자는 유용한 데이터베이스와 서비스의 사장을 막고 시스템의 유지보수와 확장을 용이하게 할 수 있도록 하는데 있다. 이를 위한 개발 전략으로는 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, 서비스의 연속성을 위하여 기존의 PC 통신서비스는 계속 유지될 수 있도록 하고, 최소 비용으로 효과적인 웹서비스로의 전환이 이루어질 수 있도록 하고자 한다. 셋째, 복수의 사용자가 동시에 접속하여 서비스를 요청하더라도 사용자의 서비스 응답속도가 저하되지 않도록 빠르고 효율적인 수행이 이루어지도록 하고자 한다. 넷째, 향후 시스템의 확장을 고려하여 기 구축되어 있는 자체 데이터베이스는 물론이고 다른 시스템의 데이터베이스도 접근하여 서비스 할 수 있는 데이터베이스 백 역할의 수행할 수 있도록 하고자 한다. 다섯째, 프로그램의 중복을 최소화하고 프로그램의 재사용을 극대화할 수 있도록 하여 프로그램의 개발과 추후 프로그램의 유지보수가 용이하도록 하고자 한다.

II. 시스템 구현방법

본 장에서는 기존의 PC 통신 기반의 데이터베이스 서비스를 최소 비용으로 인터넷 웹 기반의 데이터베이스 서비스로 효과적으로 전환하기 위한 시스

템 구현방법에 대해 연구하였다.

1. 서비스 접근이 용이한 GUI 기반 웹서비스

PC 통신 환경 상의 많은 서비스는 텍스트 기반의 명령에 의해 서비스 요청이 이루어지고, 이에 대한 서비스 제공 역시 단순 텍스트 위주로 이루어진다. 텍스트 기반의 서비스는 사용자가 이용하는데 불편하고 식상함으로 인해 지루함을 야기 시킨다. 그리고 최근 인터넷의 활성화로 과거 많은 인력과 자금을 투자하여 개발한 PC 통신 환경 상의 많은 서비스와 유용한 데이터베이스가 사용자로부터 도외시되어 사장되는 문제가 발생하게 된다. 그러므로 사용자의 정보 접근과 관리자의 유지·보수가 이하도록 하이퍼텍스트와 GUI에 기반을 둔 멀티미디어 웹에 의해 서비스의 요청과 제공이 이루어지도록 사용자 인터페이스를 구현하여야 한다.^[1]

예를 들어, 유학정보 시스템에서 미국 동부지역의 대학 정보를 조회하고자 할 경우 그림 1과 같이 유학정보 메인메뉴에서 마우스 클릭에 의해 [ABR_TOP_M.html(유학정보 주메뉴) → ABR_12_M.html(국가별대학) → ABR_12_1_M.html(미국) → ABR_12_2_1_M.html(동부지역)] 하이퍼링크를 통해 이동하여 최하위 메뉴에 도달하게 되고, 미국 동부지역의 해당 대학을 클릭하면 응용프로그램 ABR_12_A.java가 작하여 해당 대학의 대학정보를 데이터베이스에서 조회하고, 조회결과를 ABR_12_R.java가 출력양식에 맞추어 출력한다.

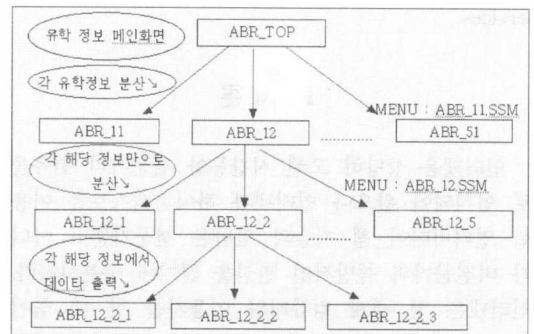


그림 1. 하이퍼텍스트와 GUI에 기반을 둔 사용자 인터페이스의 구조

앞서 설명한 바와 같이 사용자의 서비스 요청은 유학정보(혹은 지역의료정보) 서비스 메인메뉴 화면으로부터 하이퍼링크를 통해 최하위 메뉴로 이동하여 요청 서비스를 클릭 함으로써 이루어지고, 이의 요구사항이 하이퍼링크에 연결되어 있는 응용프로그램

램에 파라메타로 전달되고, 응용프로그램은 넘겨받은 파라메타 정보에 기초하여 SQL 명령을 만들어 데이터베이스를 조회하여 조회결과를 정의한 양식으로 화면에 출력한다.

사용자 인터페이스와 응용프로그램의 연동은 일반 HTML 문서의 하이퍼링크 설정작업과 유사하지만 약간의 차이가 있다. 그림 2에서와 같이 사용자가 최하위 메뉴에서 뉴욕시를 클릭하였을 경우 응용프로그램(ABR_31_A.java)이 동작되도록 하이퍼링크를 설정하기 위해서는 웹에디터의 하이퍼링크 편집창에서 주소란에 웹서버의 주소(http://210.122.180.250:8080)와 동작되어질 응용프로그램의 주소(/examples /servlet/ABR_31_A)와 전달되어져야 할 파라메타(?n_code=USA&r_code1=1&r_code2=01)를 입력하면 된다.

2. 최소 비용의 효과적인 웹서비스로의 전환

기존 PC 통신서비스의 연속성을 유지하면서 최소 비용으로 효과적인 웹서비스로 전환하기 위해서는 우선 구축되어 있는 데이터베이스를 그대로 이용하는 것이다. 즉, 데이터베이스를 재 구축하지 않고 데이터베이스의 스키마 역시 변경하지 않고 그대로 가져가는 것이다. 이리 하면 기존의 PC 통신 사용자는 PC 통신을 통해 계속해서 동일한 서비스를 제공받을 수 있고 웹서비스를 위해 새로운 응용프로그램을 개발 시에도 기존 제작된 PC 통신서비스를 위한 응용프로그램의 복잡한 SQL 명령을 해석이나 수정 없이 그대로 새 응용프로그램으로 옮겨 놓으면 되기 때문이다.

시스템 관리자는 데이터베이스의 유지·보수를 위해 데이터 조회·입력·수정·삭제 등의 기능이 요구되지만 일반 사용자의 경우에는 데이터 조회 기능만이 제공되면 되므로 웹서비스로의 전환 시에도 모든 기능을 구현하기 보단 조회 기능만을 구현하고 데이터 입력·수정·삭제 등의 관리자 기능은 기존의 PC 통신 응용프로그램을 이용하여 수행하면 최소 비용으로 효과적인 웹서비스로 전환할 수 있다.

ODBC나 JDBC 드라이버는 특정 데이터베이스 관리시스템(DBMS)의 접근 및 제어를 위한 세부 내용을 모르더라도 드라이버에서 제공되는 API를 통해 DBMS의 접근·제어를 가능하게 할뿐만 아니라 DBMS가 바뀌더라도 해당 DBMS의 ODBC나 JDBC 드라이버만을 변경하면 응용프로그램은 수정하지 않아도 되는 장점을 갖게 된다. 그러므로 Sybase DBMS를 근간으로 구축되어 있는 데이터베

이스를 접근·제어하는 응용프로그램을 개발하기 위해선 먼저 ODBC나 JDBC 드라이버를 설치한 후, 드라이버에서 제공하는 API 함수만을 통해 DBMS를 접근·제어하도록 구현하여야 DBMS 독립성이 보장된다.

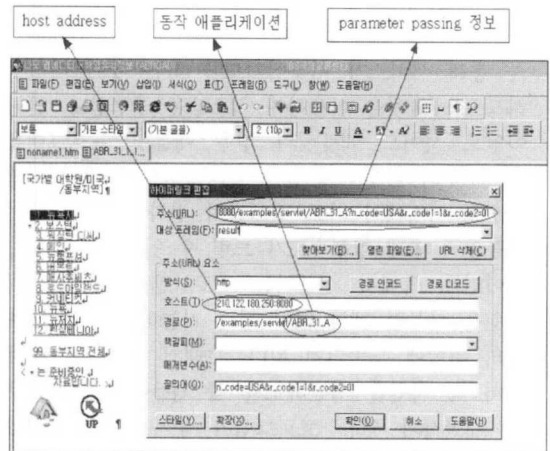


그림 2. 사용자 인터페이스와 응용프로그램의 연동

3. 서비스 요청에 빠른 실시간 서비스 제공

이전에는 동적인 웹 애플리케이션을 제작하기 위한 일반적인 방법으로 CGI를 많이 사용해 왔다. 그러나 CGI는 사용자의 서비스 요청이 있을 때마다 지식 프로세스를 생성하여 서비스를 제공하므로 수많은 사용자에게 동시 서비스 요청이 이루어지는 데이터베이스 시스템의 구현방법으로 적절하지 못하다.^[3] 그러므로 다수의 사용자가 동시에 서비스를 요청하더라도 빠른 실시간 응답이 이루어지도록 하기 위해선 스레드에 의해 서비스되도록 시스템을 구현하여야 한다.

자바 컴파일러는 이중의 하드웨어와 소프트웨어 플랫폼에서 실행되어질 수 있는 아키텍처 중립적인 바이트 코드를 생성해 내고, 이는 자바 가상머신이 설치되어 있으면 어떤 플랫폼에서도 실행되어질 수 있다.^[4] 웹 응용프로그램은 C, PERL, PHP 등 다양한 언어로 구현되어질 수 있으나, 플랫폼 독립적인 실행이 이루어지도록 하기 위해 JAVA 언어로 시스템을 구현하도록 하고, JDBC 드라이버 함수를 이용하여 데이터베이스 응용프로그램을 구현하여 데이터베이스 관리시스템(DBMS)이 바뀌더라도 데이터베이스 응용프로그램의 수정을 최소화할 수 있도록 시스템을 구현하여야 한다.

자바 프로그램은 자바 인터프리터에 의해 실행되

는 어플리케이션(application)과 웹브라우저에 의해 실행되는 애플릿(applet)으로 대별되어 진다. 그리고 애플릿은 웹브라우저에 의해 클라이언트 측에서 실행되어지는 애플릿(client-side applet)과 서버 측에서 실행되어지는 서블릿(server-side applet: servlet)으로 구분되어 진다. 사용자의 서비스 접근을 용이하도록 하기 위해서는 웹브라우저 단일 인터페이스를 통해 서비스가 이루어지도록 하는 것이 바람직하므로 애플릿을 이용하여 시스템을 구현한다. 클라이언트 사이드 애플릿의 경우 애플릿 프로그램을 서버로부터 다운로드 받아 클라이언트 측에서 실행하므로 서버에서 바로 실행하여 결과만을 전달받는 서블릿에 비해 실행속도가 느릴 수 있다. 그리고 클라이언트 측에서 실행되는 프로그램이 서버 측의 자원을 아무런 제약 없이 접근 가능하도록 하면 해킹의 위험에 노출되게 된다. 그래서 애플릿을 통해 서버 측 자원을 접근하고자 할 경우에는 웹브라우저 개발사의 인증 프로그램이 추가로 설치되어야 하나 상당히 고가라는데 더 큰 문제가 있다. 그러므로 사용자의 서비스 접근은 웹브라우저 통해 이루어지도록 하고, 서비스 제공을 위한 응용프로그램은 서블릿에 의해 구현되도록 한다.

결론적으로 복수의 사용자가 동시에 접속하여 서비스를 요청할 경우에도 빠른 응답 서비스를 제공하고, 플랫폼 독립적인 실행이 가능하도록 하기 위해 쓰레드 기반의 자바 서블릿으로 시스템을 구현하여야 한다.

4. 시스템 확장을 고려한 데이터베이스 뱅크

앞 서 언급하였듯이 응용프로그램이 JDBC 드라이버에서 제공하는 API 함수만을 통해 DBMS를 접근·제어하도록 함으로써 DBMS가 바뀌더라도 응용프로그램은 수정하지 않아도 되는 DBMS 독립성이 보장된다. JDBC의 API는 2-tier와 3-tier를 모두 지원하고 있으며, 2-tier는 자바 애플릿이나 어플리케이션 클라이언트가 데이터베이스를 직접 접근하는 구조를 가지고, 3-tier란 클라이언트는 서비스를 위한 미들티어(middle-tier)에 명령을 보내고 미들티어가 데이터베이스를 접근하는 구조를 가진다.¹⁵⁾ 전형적인 2-tier 구조의 클라이언트/서버 시스템은 프로그래밍이 간단하다는 장점이 있지만 클라이언트 측의 어플리케이션이 SQL 명령을 직접 DBMS에 전달·실행하기 때문에 DBMS 노출로 인한 해킹의 우려가 크다. 또한 데이터베이스 구조에 변경이 가해지면 2-tier 구조의 경우 클라이언트 측의 프로그

램 수정이 불가피하므로 클라이언트가 많은 경우 유지보수에 있어 많은 어려움이 뒤따른다. 그리고 짧은 시간에 동시다발적인 서비스 요청으로 한 곳의 DBMS 서버에 많은 부하가 걸리는 경우에도 부하분산(load balancing)의 어려움과 많은 네트워크 트래픽을 야기 시키는 단점을 가지고, 여러 DBMS 서버를 접근하여 통합 데이터베이스 서비스를 제공하는 것이 어려워 시스템의 확장 측면에 있어서도 문제가 있다.

그러므로 시스템의 보안, 유지보수, 부하분산 및 시스템의 확장이 용이한 3-tier 구조의 데이터베이스 응용시스템이 되도록 구현하여야 한다. 즉, 사용자는 웹브라우저 단일 인터페이스만을 이용하여 웹서버에 서비스를 요청하고, 웹서버에 의해 구동되는 미들티어의 데이터베이스 응용프로그램이 DBMS 서버를 제어하여 분산 데이터베이스 환경에서도 서비스가 제공되어질 수 있도록 시스템을 구현하여야 한다. 그래서 기 구축되어 있는 자체 데이터베이스는 물론이고 분산된 관계기관의 정보 DB를 연계하여 통합 서비스도 가능한 시스템 구조를 갖도록 하여 시스템 확장이 용이하도록 하여야 한다.

본 시스템은 그림 3과 같은 3-tier 구조의 클라이언트/서버 시스템으로 동작한다. 사용자는 웹브라우저 단일 인터페이스를 통하여 웹서버에 서비스를 요청하고, 웹서버는 쓰레드를 생성하여 미들티어의 데이터베이스 응용프로그램을 실행하고, 응용프로그램은 DBMS 서버를 제어하여 분산환경의 데이터베이스를 접근하여 요청된 서비스를 제공한다.

클라이언트의 시스템 환경은 넷스케이프 혹은 익스플로러의 웹브라우저 설치만으로 충분하고, 미들티어의 웹서버에는 리눅스 기반의 아파치 웹서버와 Java 서블릿의 실행을 위한 JSDK와 JServ와 해당 DBMS의 접근·제어를 위한 JDBC 드라이버를 설치하고, DBMS 서버에는 Sybase DBMS와 기 구축된 데이터베이스가 위치하도록 한다.

사용자들은 웹브라우저를 통하여 미들티어의 아파치 웹서버를 접근하여 하이퍼링크에 의해 구조화되어 있는 메뉴를 통해 정보조회 등의 서비스를 요청하면, 이를 파라메타 화하여 해당 서블릿 프로그램에게 전달하게 된다. 해당 서블릿 프로그램은 넘겨받은 파라메타 정보에 따라 SQL 명령을 만들어 JDBC 드라이버를 통해 해당 DBMS 서버를 제어하여 데이터베이스를 접근하고, 조회결과를 사용자가 원하는 형태의 웹문서 형식으로 변환하여 웹서버로

넘겨주면, 웹서버는 이를 사용자의 웹브라우저로 전송하여 서비스 결과가 사용자에게 보여지게 된다.

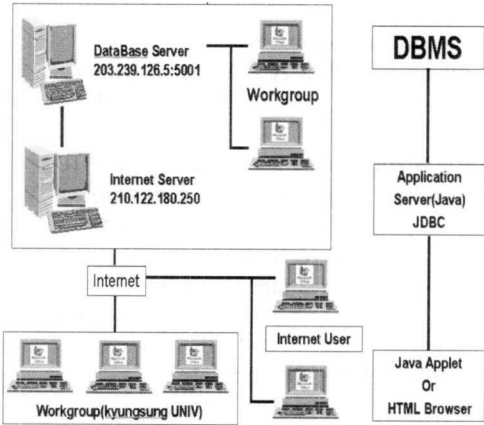


그림 3. 인터넷 환경에서의 3-tier 클라이언트/서버 시스템의 동작

5. 프로그램 중복의 최소화과 재사용의 극대화

프로그램의 중복을 최소화하기 위해서는 우선 데이터베이스 응용프로그램의 구조 및 동작을 면밀히 분석하여 반복적으로 수행되는 루틴을 공통 루틴으로 독립시키는 것이 중요하다. 특히, 사용자 서비스를 위한 입출력 양식을 최소화하고 사용자 인터페이스를 간소화하는 것이 불필요한 노동력의 낭비를 막는 첩경이 된다. 그러므로 응용프로그램의 구조 및 동작 분석을 통하여 공통 루틴을 만들고, 최소화한 공통의 입출력 양식을 통해 서비스 제공이 가능하도록 시스템을 구현하여야 한다.

사용자의 데이터베이스 시스템에 대한 정보 요청은 다양하지만 이를 처리하는 데이터베이스 응용프로그램은 SQL 명령만을 달리할 뿐 실제 그 처리절차는 거의 동일하다. 그러므로 사용자의 정보요청별로 별도의 서비스 프로그램을 작성하는 것 보다 사용자의 정보요청을 파라메타로 전달받아 이를 기초로 SQL 명령을 만들고 공통 루틴을 통해 데이터베이스 관련 서비스를 제공하는 형태로 프로그램을 개발하면 프로그램의 수를 대폭 줄일 수 있으며, 차후 프로그램의 유지보수에 있어서도 매우 유리할 것이다. 그래서 사용자의 정보요청을 파라메타로 전달하고 데이터베이스의 접근은 공통 루틴에 의해 이루어지도록 하여 프로그램의 중복은 최소화하고, DBMS의 접근제어를 위한 주요 루틴은 객체화하여 프로그램의 재사용이 극대화될 수 있도록 프로그램을 개발하여 차후 프로그램의 유지보수와 새로운

데이터베이스 응용시스템의 개발이 용이하도록 시스템을 구현하여야 한다.

구현하고자 하는 데이터베이스 시스템의 서비스 종류가 다양하고 복잡할 경우 많은 메뉴파일과 응용프로그램 파일이 요구된다. 이로 인해 차후 관리 또는 수정·보완 시에 해당 파일을 찾는 데 어려움을 겪을 우려가 있어 파일 이름만으로도 관련 DB, 서비스의 종류, 파일의 종류를 쉽게 파악할 수 있도록 파일을 그림 4와 같이 명명한다.

| 데이터베이스 이름 | 서비스의 종류 (메뉴와 응용프로그램의 계층구조) | 파일의 종류 |
|--|-------------------------------|--------------|
| 유학 D B: ABR | 메인 메뉴: TOP | 메뉴: M |
| 의료 D B: MED | 서비스 종류: 상 하위 서비스번호 | 응용프로그램: A, R |
| 예) A B R_TOP_M.html 유학정보 메인 메뉴 파일 | | |
| A B R_12_1_A.java 유학정보_대학원_미국 정보조회 응용프로그램 | | |

그림 4. 파일의 체계적 관리를 위한 명명법

III. 시스템 구현

제한한 구현방법을 따라 구현한 본 시스템은 하이퍼텍스트와 GUI에 기반을 둔 멀티미디어 웹에 의해 서비스가 이루어지고, 기 구축된 데이터베이스를 변경없이 수용하여 기존 PC 통신서비스의 연속성을 유지하면서 최소 비용으로 웹서비스로의 전환이 이루어지도록 하였고, 빠른 서비스를 위해 쓰레드 기반의 자바 서블릿⁶⁾을 근간으로 하였고, 시스템의 보안과 확장이 용이하도록 3-tier 구조의 클라이언트/서버 시스템이며, 프로그램 중복의 최소화과 재사용의 극대화가 이루어지도록 구현하였다.

1. 응용프로그램 구조

사용자가 요청한 서비스 제공을 위한 응용프로그램은 데이터베이스 제어 프로그램, 항목조회 프로그램, 내용조회 프로그램, 항목조회 출력프로그램, 내용조회 출력프로그램으로 구성되어 있다. 데이터베이스의 연결과 해제를 위한 데이터베이스 제어 프로그램, 조회된 결과를 정의한 양식으로 화면에 출력하기 위한 항목조회 출력프로그램과 내용조회 출력 프로그램은 프로그램의 중복을 최소화하고 재사용을 극대화하기 위해 객체화하여 공통으로 사용하고, 사용자의 정보요청을 파라메타로 전달받아 SQL 명령

을 만들어 데이터베이스를 조회하는 항목조회 프로그램과 내용조회 프로그램은 서비스 별로 프로그램을 작성하였다.

유학정보 시스템에서 미국 동부지역 뉴욕시의 대학원 정보를 조회하고자 할 경우 그림 6과 같이 유학정보 메인메뉴에서 마우스 클릭에 의해[ABR_TOP_M.html(유학정보 메인메뉴)→ABR_31_M.html(대학원안내)→ABR_31_1_M.html(미국)→ABR_31_1_1_M.html(동부지역)] 하이퍼링크를 통해 도달한 최하위 메뉴에서 특정 시(예를 들어, 뉴욕시)를 클릭하면 응용프로그램 ABR_31_A.java가 동작하게 되고, 이때 사용자의 요구사항이 파라메타(n_code='USA', r_code1=1,r_code2=01)로 전달되게 된다.

ABR_31_A.java는 데이터베이스제어 프로그램 ConnectionDB.java를 이용하여 데이터베이스 서버에 접속하고, 넘겨받은 파라메타 정보에 기초하여 SQL 명령을 만들어 데이터베이스를 조회하여 뉴욕시 내의 대학원 리스트를 항목조회 출력프로그램 ABR_ListTable.java를 이용하여 정의한 양식으로 화면에 출력한다. 사용자가 뉴욕시 내의 특정 대학(예를 들어, 뉴욕주립대)을 클릭하면, 내용조회 출력 프로그램 ABR_31_R.java가 동작하게 되고, 이때 사용자의 요구사항이 파라메타(u_code=5)로 전달되게 된다. ABR_31_R.java 역시 데이터베이스제어 프로그램을 이용하여 데이터베이스 서버에 접속하고, 넘겨받은 파라메타 정보에 기초하여 SQL 명령을 만들어 데이터베이스를 조회하여 뉴욕주립대의 상세정보를 내용조회 출력프로그램 ABR_ResultTable.java를 이용하여 정의한 양식으로 화면에 출력한다.

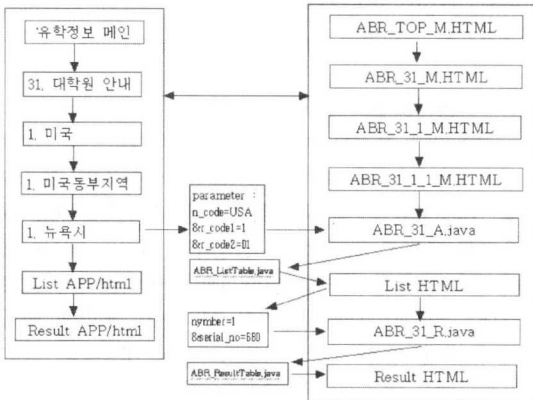


그림 5. 데이터베이스 응용프로그램의 구조 및 동작

```

public class ConnectionDB
// 데이터베이스 제어 클래스
{
static String _dburl =
    "jdbc:sybase:Tds:203.239.126.5:5001":
    // DBMS 서버의 주소

static String _user = "gong"; // 사용자 ID
static String _password = "gonggong";
    // 사용자 패스워드

static String driverName =
    "com.sybase.jdbc2.jdbc.SybDriver":
    // JDBC 드라이버 명

/* DBMS 연결을 위한 메소드 */
public void connectDB() throws
    ClassNotFoundException,
    SQLException, Exception {
    try {

/*JDBC 드라이버를 로드하고 드라이버 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
}

/* DBMS 연결해제를 위한 함수 */
public void disconnectDB() throws
    SQLException
{
    try{ stmt.close();
// SQL 명령 실행을 위한 Statement 해제

    cont.close(); // Database 접속 해제
} catch (SQLException ex)
{
    // 예외상황을 처리하기 위한 catch문
    throw ex;
}
}
}
    
```

그림 6. 데이터베이스 제어 프로그램

2. 구현 프로그램

사용자 서비스 제공을 위한 응용프로그램은 쓰레드에 기반을 둔 자바 서블릿 프로그램으로 구현하여 복수의 사용자가 동시에 서비스를 요청하더라도 빠른 응답이 이루어지도록 하였다. 향후 유사한 웹 기반의 3-tier 데이터베이스 시스템 개발에 도움이 되도록 시스템 구현을 위한 응용프로그램의 핵심 소스를 상세히 기술하였다.^[7]

1) 데이터베이스 제어 프로그램

데이터베이스제어 프로그램(ConnectionDB.java)은 데이터베이스의 연결과 해제 기능을 수행하며,

```
import java.io.*;
public class ABR_ListTable // 조회항목 출력 클래스
{
    PrintWriter out : // 출력을 위한 객체

    /* 조회항목 출력테이블의 생성자 메소드 */
    public ABR_ListTable(PrintWriter out)
    { this.out = out;

    /* 조회항목 출력테이블의 헤드출력을위한 메소드 */
    public void tableHead(String h1,String h2)
    {
        out.println("<tableheight=50
        align=center width=100% cellpadding=0
        cellspacing=0)" + "<tr bgcolor=#CCCCCCFF"
        + "<td width=14%" + h1 + "</td" +
        "<td width= * )" + h2 + "</td)" + "</tr)" );
    }

    /*조회항목 출력테이블의 내용출력을 위한 메소드*/
    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)" );
    }

    /*조회항목 출력테이블의 꼬리출력을 위한 메소드#1 */
    public void tableTail()
    {
        out.println("</table)" );
    }

    /* 조회항목 출력테이블의 꼬리출력을 위한 메소드 #2 */
    public void tableTail(int total, int current,
    String prelink, String nextlink) { . . . }
}
```

그림 7. 항목조회 출력 프로그램

이의 서블릿 프로그램은 그림 7과 같다. 데이터베이스 관리시스템(DBMS)이 변경되더라도 JDBC 드라이버만을 변경하면 응용프로그램은 수정하지 않아도 되도록 하였으며, 공통 루틴으로 객체화하여 사용자의 데이터베이스 접근 시에 재사용이 가능하도록 하였다^[8-10]

2) 항목조회 출력 프로그램

항목조회 출력프로그램(ABR_ListTable.java)은 조회된 항목을 정의한 양식으로 화면에 출력하는 기능을 수행하며, 이의 서블릿 프로그램은 그림 8과 같다. 이 역시 프로그램의 중복을 최소화하고 재사용을 극대화하기 위해 공통 루틴으로 객체화하였다.

```
import java.io.*;
public class ABR_ResultTable
        // 조회내용 출력 클래스
{
    . . .
    PrintWriter out : // 출력을 위한 객체
        /* 조회내용 출력테이블의 생성자 메소드 */

    public ABR_ResultTable(PrintWriter out)
    { this.out = out; }

    /*조회내용 출력테이블의 헤드출력을 위한 메소드*/
    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=#CCCCCCFF)
        번호</td)" + "<td width=30% bgcolor=
        #CCFFCC)" + h1 + "</td)"
        + "<td width=10% bgcolor=#CCCCCCFF)
        제목</td)" + "<td width=44%
        bgcolor=#CCFFCC)" + h2 + "</td)</tr)"
        + "<tr bgcolor=#CCCCCCFF)<td
        colspan=4>[정보]</td)</tr)" );
    }

    /* 조회내용 출력테이블의 내용출력을 위한 메소드 */
    public void resultTableContent(String c1) {
        out.println("<tr bgcolor=#FFFFFF)<td
        colspan=4 height=300)" + "<pre)" + c1 +
        "</pre)" + "</td)</tr)" );
    }

    /*조회내용 출력테이블의 꼬리출력을 위한 메소드*/
    public void resultTableTail() {
        out.println("</table)" );
    }
}
```

그림 8. 내용조회 출력 프로그램

3) 내용조회 출력 프로그램

내용조회 출력프로그램(ABR_ResultTable.java)은 출력된 항목들 중 사용자에게 의해 선택된 항목의 상세 정보를 정의한 양식으로 화면에 출력하는 기능을 수행하며, 이의 서블릿 프로그램은 그림 9와 같다. 이 역시 프로그램의 중복을 최소화하고 재사용을 극대화하기 위해 공통 루틴으로 객체화하였다.

```
public class ABR_31_A extends
    HttpServlet // 항목조회 클래스
{
    String currentProgram = "ABR_31_A";
    String nextProgram = "ABR_31_R";

    public void doGet(HttpServletRequest req,
        HttpServletResponse res)
        throws ServletException, IOException{
    printPageHead(out, "해외유학정보/국가별 학교안내/대학원 안내");
    /* 서블릿 프로그램의 입력 파라메터를 구한다. */
    n_code=req.getParameter("n_code");
    r_code1 = req.getParameter("r_code1");
    r_code2=req.getParameter("r_code2");
    conDB = new ConnectionDB();
    conDB.connectDB(); // DBMS 연결
    /*항목 조회를 위한 SQL 명령을 만들어 데이터베이스를
    조회한다.*/
    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");

    /* 항목 출력테이블의 헤드를 출력한다. */
    listTable.tableHead("번호", "학교명" + "
    (총 " + rowTotal + "건)");
    /* 조회항목을 출력테이블에 출력한다. */
    while ( rs.next() )
    {
        listTable.tableContent(number++,
            rs.getString(2), nextProgram + "?
            number="+number +
            "&serial_no=" + rs.getString(1));
    conDB.disconnectDB();// DBMS 연결해제
    /*항목출력 테이블의 꼬리를 출력과 HTML 문서 의 종료 */
    listTable.tableTail();
    printPageTail(out);
    }
    }
}
```

그림 9. 항목조회 프로그램

4) 항목조회 프로그램

항목조회 프로그램(ABR_31_A.java 등)은 사용자의 조회요청을 파라메타로 전달받아 SQL 명령을 만들어 데이터베이스로부터 항목을 조회하는 기능을 수행한다. 이는 서비스 별로 프로그램이 구현되어 있으나 SQL 명령만이 다르고 서블릿 프로그램은 그림 10과 동일하다.

```
public class ABR_31_R extends
    HttpServlet // 내용조회 클래스
{
    String currentProgram = "ABR_31_R";
    String nextProgram = "";

    public void doGet(HttpServletRequest req,
        HttpServletResponse res)
        throws ServletException, IOException
    {
    /* HTML 문서의 시작 */
    printPageHead(out, "해외유학정보/국가별 학교
    안내/대학원 안내");
    /* 서블릿 프로그램의 입력 파라메터를 구한다. */
    number = req.getParameter("number");
    // 항목 테이블 내의 순번 serial_no =
    req.getParameter("serial_no");
    // DB 내의 대학 일련번호
    conDB = new ConnectionDB();
    conDB.connectDB(); // DBMS 연결

    /* 내용 조회를 위한 SQL 명령을 만들어 데이터베이스를
    조회한다. */
    rs = conDB.stmt.executeQuery("Select
    SERIAL_NO,TEXT_LCNT,NAME,SCHOOL_DESC
    " + "from ABR_DB..SCHOOL_GRAD
    " + "where SERIAL_NO = " + serial_no);

    while( rs.next() ) {
        /* 내용 출력테이블의 헤드를 출력한다. */
        ResultTable.resultTablehead(number,rs.getString(3));
        /* 조회내용을 출력테이블에 출력한다. */
        ResultTable.resultTableContent(rs.getString(4));
    }
    conDB.disconnectDB(); // DBMS 연결해제
    /* 내용 출력테이블의 꼬리 출력과 HTML 문서의 종료
    */ResultTable.resultTableTail();
    printPageTail(out);
    }
}
```

그림 10. 내용조회 프로그램

5) 내용조회 프로그램

내용조회 프로그램(ABR_31_R.java 등)은 사용자가 선택한 항목을 파라메타로 전달받아 SQL 명령을 만들어 데이터베이스로부터 상세 내용을 조회하는 기능을 수행한다. 이 역시 서비스 별로 프로그램이 구현되어 있으나 SQL 명령문이 다르고 서블릿 프로그램은 그림 11과 동일하다.

IV. 성능평가

기존의 PC 통신서비스를 인터넷 웹서비스로의 효율적 전환을 위해 제안한 구현방법을 따라 시스템을 구현하였으며, 이의 검증은 위해 시스템 시험과 성능평가를 수행하였다.

1. 시스템 시험

기존의 PC 통신서비스를 인터넷 웹서비스로의 성공적 전환 여부를 검증하기 위해 사용자인터페이스의 호환성과 서비스의 호환성 측면에서 시스템 시험을 실시하였다.

기존의 PC 통신서비스의 사용자인터페이스는 상하위 메뉴의 이동과 전후 화면의 이동 등 모든 명령을 직접 입력해야 하고 이에 대한 서비스 역시 텍스트 형태로 제공되므로 이용의 불편함과 단조로움이 뒤따랐다. 반면, 인터넷 웹서비스의 사용자인터페이스는 하이퍼텍스트와 멀티미디어에 기반을 두고 있기에 바뀐 사용자 환경에 적응하는데 전혀 어려움이 없고 서비스 접근이 용이하고 서비스가 다양해졌다는 사용자들의 반응이었다.

또한 서비스의 호환성 여부를 검증하기 위해 각 단계별 모듈의 작동 여부와 전체 시스템의 동작을 케이스 별로 시험한 결과 모두 이상 없이 작동함을 확인할 수 있었다.

2. 성능평가

본 논문에서는 복수의 사용자가 동시에 접속하여 서비스를 요청할 경우에도 빠른 응답 서비스를 제공하도록 하기 위해 쓰레드 기반의 자바 서블릿으로 시스템을 구현하였다. 사용자 측면에서 볼 때 구현한 시스템은 하나의 블랙박스로 보여지고, 이의 프로그램 구조나 모듈별 실행시간보다는 실제 요구한 서비스가 사용자에게 얼마나 정확하게 빨리 제공되는지가 더 중요하므로, 복수 사용자의 동시 서비스 요청에 대한 서비스 응답시간을 측정하여 CGI와 서블릿으로 구현 시를 비교함으로써 이의 성능을 평가하였다.

다수의 사용자가 동시에 서비스를 요청하는 경우 각각의 사용자에게 서비스가 제공되기까지의 응답시간을 측정하였다. 항목조회 프로그램을 서블릿으로 구현한 경우 응답시간은 표 1과 같고, CGI로 구현한 경우는 표 2와 같다. 그리고 응답시간의 증가 추이를 쉽게 파악 가능하도록 그림 12에 서블릿과 CGI를 대비시켜 막대그래프로 나타내었다.

이를 분석한 결과 동시에 서비스를 요청하는 사용자가 많아지더라도 서블릿을 이용한 시스템 구현은 응답시간의 증가가 완만한데 반해 CGI를 이용한 시스템 구현은 응답시간이 급격하게 증가함을 알 수 있었다. 즉, 동시에 서비스를 요청하는 사용자가 많아질수록 서블릿을 이용한 시스템 구현이 CGI를 이용한 시스템 구현보다 훨씬 효율적이고 대략 10배나 빠르게 실행하는 것을 알게 되었으며, 수많은 사용자에 의해 동시에 서비스 요청이 이루어지는 데이터베이스 시스템의 구현방법으로 CGI보다 서블릿 더 적절함을 입증할 수 있었다.

또한 표 3에서와 같이 사용자가 처음 서비스를 요청하였을 때는 지식 프로세스의 생성으로 CGI와 서블릿 구현 모두 두 번째 서비스 요청에 대한 응답시간보다 상대적으로 길었다. 그런데 두 번째 응

표 1. 서블릿 항목조회 프로그램의 응답시간

| 작성코드 | 서블릿(항목조회 프로그램) | | | | | | | | | | | | | |
|------|----------------|---------------|------|------|-------|-------|-------|-------|-------|-------|------|--|----|------|
| | 사용자 수 | 응답 시간 (단위: 초) | | | | | | | | | | | 평균 | |
| 1명 | 1.83 | | | | | | | | | | | | | 1.83 |
| 2명 | 1.70 | 3.43 | | | | | | | | | | | | 2.57 |
| 3명 | 1.55 | 4.78 | 5.10 | | | | | | | | | | | 3.81 |
| 4명 | 1.54 | 3.39 | 4.01 | 6.22 | | | | | | | | | | 3.79 |
| 5명 | 1.83 | 3.62 | 5.22 | 5.69 | 7.73 | | | | | | | | | 4.82 |
| 6명 | 1.81 | 3.15 | 5.48 | 6.48 | 6.57 | 9.80 | | | | | | | | 5.55 |
| 7명 | 1.87 | 3.90 | 4.99 | 6.80 | 7.80 | 8.87 | 9.16 | | | | | | | 6.20 |
| 8명 | 1.35 | 3.63 | 6.90 | 7.62 | 7.71 | 9.24 | 10.07 | 11.30 | | | | | | 7.23 |
| 9명 | 3.40 | 4.29 | 5.58 | 7.42 | 9.63 | 10.14 | 10.42 | 10.74 | 12.25 | | | | | 8.21 |
| 10명 | 2.83 | 4.12 | 6.13 | 7.02 | 10.11 | 10.69 | 10.70 | 11.40 | 11.69 | 14.25 | 8.89 | | | |

표 2. CGI 항목조회 프로그램의 응답시간

| 작성코드 | 서블릿(항목조회 프로그램) | | | | | | | | | | | | | |
|------|----------------|---------------|------|------|-------|-------|-------|-------|-------|-------|------|--|----|------|
| | 사용자 수 | 응답 시간 (단위: 초) | | | | | | | | | | | 평균 | |
| 1명 | 1.83 | | | | | | | | | | | | | 1.83 |
| 2명 | 1.70 | 3.43 | | | | | | | | | | | | 2.57 |
| 3명 | 1.55 | 4.78 | 5.10 | | | | | | | | | | | 3.81 |
| 4명 | 1.54 | 3.39 | 4.01 | 6.22 | | | | | | | | | | 3.79 |
| 5명 | 1.83 | 3.62 | 5.22 | 5.69 | 7.73 | | | | | | | | | 4.82 |
| 6명 | 1.81 | 3.15 | 5.48 | 6.48 | 6.57 | 9.80 | | | | | | | | 5.55 |
| 7명 | 1.87 | 3.90 | 4.99 | 6.80 | 7.80 | 8.87 | 9.16 | | | | | | | 6.20 |
| 8명 | 1.35 | 3.63 | 6.90 | 7.62 | 7.71 | 9.24 | 10.07 | 11.30 | | | | | | 7.23 |
| 9명 | 3.40 | 4.29 | 5.58 | 7.42 | 9.63 | 10.14 | 10.42 | 10.74 | 12.25 | | | | | 8.21 |
| 10명 | 2.83 | 4.12 | 6.13 | 7.02 | 10.11 | 10.69 | 10.70 | 11.40 | 11.69 | 14.25 | 8.89 | | | |

답시각이 첫 번째 응답시간에 비해 CGI의 경우에는 약 15% 감소한데 반해, 서블릿의 경우에는 약 30% 감소하였다. 이는 사용자가 서비스를 요청할 때마다 CGI는 오버헤드가 큰 프로세스를 생성하여 서비스하는 데 반해, 서블릿은 오버헤드가 작은 쓰레드를 생성하여 서비스하는 것을 시사한다.

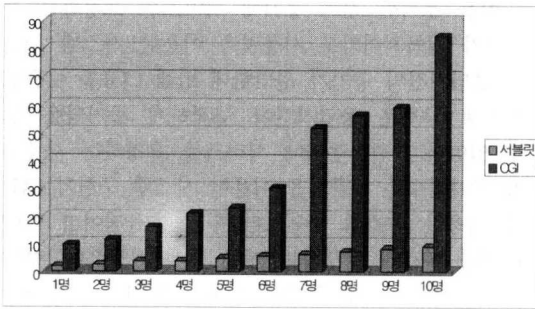


그림 11. 서블릿과 CGI 구현의 응답시간 비교

표 3. 서블릿과 CGI 구현의 응답시간 비교(단위: 초)

| 횟수 | CGI | | 서블릿 | |
|----|-----------|----------|----------|----------|
| | 첫 번째 응답시간 | 두번째 응답시간 | 첫번째 응답시간 | 두번째 응답시간 |
| 1회 | 11.03 | 9.34 | 2.42 | 1.64 |
| 2회 | 10.07 | 9.12 | 3.04 | 1.94 |
| 3회 | 12.34 | 10.34 | 3.55 | 1.66 |
| 4회 | 10.44 | 10.12 | 3.74 | 1.90 |
| 5회 | 11.56 | 9.56 | 2.98 | 1.77 |
| 평균 | 11.08 | 9.69 | 3.14 | 1.78 |

V. 결론

기존의 PC 통신 서비스가 최근 인터넷에 기반을 둔 멀티미디어 웹 서비스로 빠르게 대체됨으로 인해 과거 많은 인력과 자금을 투자하여 개발한 유용한 데이터베이스와 PC 통신 환경 상의 많은 서비스가 사용자로부터 도외시되어 사장되는 문제가 발생하고 있었다. 이에 본 연구에서는 과거 PC 통신 환경 상에서 제공되었던 텍스트 위주의 명령에 기초한 데이터베이스 서비스를 사용자의 정보 접근이 용이한 멀티미디어 웹에 기초한 데이터베이스 서비스로 효율적 전환을 위한 구현방법 및 기술에 관해 연구하였다.

본 논문에서는 현재 하이텔의 인포샵 서비스를 통해 제공되고 있는 PC 통신 기반의 해외유학정보 데이터베이스 시스템을 대상으로 하여 하이퍼텍스트와 GUI에 기반을 둔 멀티미디어 웹에 의해 서비스

의 요청과 제공이 이루어질 수 있도록 하여 사용자의 서비스 접근과 관리자의 유지·보수가 용이하도록 하는 구현방법, 사용자의 요청에 따른 서비스가 쓰레드에 의해 제공되는 서블릿 프로그램을 이용하여 시스템을 구현하여 복수의 사용자가 동시에 서비스를 요청하더라도 빠른 응답이 이루어지도록 하는 구현방법에 대해 상세히 기술하였다.

그리고 기존 PC 통신의 데이터베이스를 변경 없이 그대로 이용함으로써 새로운 데이터베이스의 재구축을 위한 비용을 절감할 수 있고 새로운 응용프로그램에 기존의 SQL 명령을 수정 없이 그대로 적용할 수 있도록 하고, DBMS 변경에 따른 응용프로그램의 수정을 최소화하기 위해 DBMS를 접근·제어는 JDBC 드라이브를 통하여 이루어지도록 하는 등의 기존 PC 통신서비스의 연속성을 유지하면서 최소 비용으로 효과적인 웹서비스로의 전환을 위한 구현방법에 대해 기술하였다.

또한 사용자는 웹브라우저 단일 인터페이스만을 이용하여 웹서버에 서비스를 요청하고, 웹서버에 의해 구동되는 미들웨어의 데이터베이스 응용프로그램이 DBMS 서버를 제어하여 분산 데이터베이스 환경에서도 서비스가 제공되어질 수 있도록 하는 3-tier 구조의 데이터베이스 시스템의 구현방법에 대해 기술하였고, DBMS 제어 등의 주요 모듈을 객체화하여 프로그램 재사용의 극대화를 위한 구현방법, 데이터베이스 응용프로그램은 SQL 명령만을 달리할 뿐이지 실제 처리절차는 거의 동일하므로 사용자의 정보요청을 파라메타로 전달받아 이를 SQL 명령으로 변환하여 DBMS 제어 객체를 통해 서비스가 제공될 수 있도록 함으로써 프로그램 중복의 최소화와 유지보수가 용이하도록 하는 구현방법에 대해서도 상세히 기술하였다.

끝으로 본 논문에서 제시한 응용프로그램의 핵심 소스는 이와 유사한 웹 기반의 3-tier 데이터베이스 시스템의 구현에 많은 도움이 되리라 생각되며, 동적인 웹 애플리케이션 제작을 위한 일반적인 방법인 CGI를 이용한 구현과 본 논문에서 제안한 쓰레드 기반의 서블릿을 이용한 구현의 성능을 비교·평가함으로써 여러 사용자가 동시에 서비스를 요청하더라도 빠른 실시간 응답이 이루어짐을 입증하였다.

참고 문헌

[1] 박재홍, 광정필, 최재원, "인터넷 웹 기반의 유학 정보 데이터베이스 시스템 구축", 한국해양정보통신

신학회, 춘계종합학술대회 논문집, 4권 1호, 2000년 5월.

[2] 서전텔콤, "INFOSHOP 접속을 위한 시스템 구성 및 운영/유지보수 지침", 서전텔콤 연구소, 기술자료 TM-96-01, 1996년 3월.

[3] 최재원, "자바 서블릿을 이용한 3-Tier 통합 데이터베이스 시스템 구축", 경성대학교 공학기술연구소 논문집, 7권 1호, 2000년 12월.

[4] 최재영, 최종명, 유재우, *프로그래머를 위한 Java 2TM*, 홍릉과학출판사, 2000년. 1월

[5] 손우상, *about JDBC*, 영진출판사, 2000년. 10월

[6] 유중명, 최재영, 안보희 공저, *사이버 강의로 배우는 웹 개발자를 위한 서블릿 JSP*, 이한디지탈, 1999년.

[7] 광재홍, 광정필, 최재원, "자바 서블릿을 이용한 인터넷 웹 기반의 3-Tier 데이터베이스 시스템 구축", 한국해양정보통신학회, 춘계종합학술대회 논문집, 4권 2호, 2000년 10월.

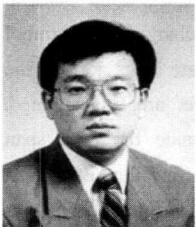
[8] Fishman, D, H., et al., "IRIS: An Object-Oriented Database Management System", *ACM Transaction in Office Information System*, Vol. 5, No 1, 1987.

[9] 유은민, 서범수, 김영국, 정석찬, "웹과 데이터베이스 연동기법 분석", 한국정보처리학회, Vol. 6, No. 1, 1999.

[10] 정기상, 김대인, 황부현 "이기종 데이터베이스에서 서블릿과 JDBC를 이용한 상품 검색", *Proceeding*, 한국정보처리학회지, Vol. 6, No. 1, 1999.

최재원(Jae Weon Choe)

정회원



1988년 2월 : 고려대학교

컴퓨터공학과 (공학사)

1990년 8월 미시간주립대학교

컴퓨터공학과 (공학석사)

1995년 8월 건국대학교

전자공학과 (공학박사)

1990년 10월~1997년 8월 : 삼성전자 정보통신연구소 선임연구원

1997년 9월~2002년 현재 : 경성대학교 전기전자·컴퓨터공학부 조교수

<주관심 분야> 정보통신망, 이동통신, 운영체제, 인터넷응용