
Java를 이용한 인터넷기반 원격시험시스템의 구현

김운호* · 박운영**

An Implementation of Remote Examination System
on Internet using Java/SWING

Yun-Ho Kim* · Un-Young Park**

요 약

본 논문에서는 Java/SWING을 이용한 인터넷 기반의 원격 시험 시스템의 설계와 구현을 제시한다. 본 시스템은 문제를 출제하는 문제 출제부와 문제를 풀이하는 문제 풀이부와 문제 풀이한 결과를 참조할 수 있는 문제 평가부로 구성된다. 문제 풀이부는 2-tier 구조를 갖는 자바 애플리케이션으로 문제 풀이부와 평가부는 3-tier 구조를 갖는 자바 애플릿으로 구현한다. 데이터베이스의 처리는 Java의 JDBC API를 이용하여 프로그래밍함으로써 특정 데이터베이스 시스템에 의존하지 않고 범용의 데이터베이스 처리가 되도록 한다. 본 논문에서의 원격 시험 시스템은 각종 성적 평가를 위하여 이용할 수 있으며, 기존의 필기 시험의 대치 또는 인터넷에 기반한 설문조사를 시행하는 시스템으로의 전환도 가능하여 다양한 분야에 기여할 수 있을 것으로 기대된다.

ABSTRACT

This paper presents an implementation of an Remote Examination System on Internet using Java and SWING GUI API. The system is consisted of three parts: generating examination part that is Java application, answering questions part that is Java Applet, reviewing results part that is Java Applet. The first part is implemented with the 2-tier client/server architecture and the last two parts is implemented the 3-tier client/server architecture. Database programming is done using Java JDBC API for database independence and portability. This system can be used the evaluation of lectures, and substituted for pen-and-pencil examinations or adopted to survey applications based on Internet.

키워드

Internet Application, Java programming, Cyber-University

* 안동대학교 전자정보산업학부

** 컴포넌트뱅크 연구원

1. 서 론

인터넷 사용자의 급속한 증가와 통신망의 고속화는 인터넷 응용 프로그램의 급속한 증가와 변화를 가져왔다. 이에 따라 최근 주목을 받는 분야 중에 하나가 인터넷 기반의 가상대학이다. 가상대학에서는 교수와 학생간의 교수와 학습 및 평가, 응시의 시행이 원격으로 이루어지게 된다. 현재 이 분야의 연구 추세는 주로 원격 강의 서비스 시스템의 구축에 초점이 맞추어져 있다. 그러나 가상대학에서 해결해야 할 또 다른 중요한 부분은 학생의 수학 능력을 평가하는 시스템이다. 현재 가상대학 시스템에서 적용된 시험 시스템에서는 시험의 출제는 문제 출제자가 직접 웹 페이지를 작성하거나 Server Side Script를 통해 생성된 HTML 폼을 사용하는 형태를 취하고 있는데, 이러한 방식은 문제의 출제에 있어서 번거로움과 불편함을 야기시키고 사용자 인터페이스 부분에서 취약한 측면을 가지고 있다.

기존의 연구에 대하여 간단히 살펴보면, CGI를 사용하여 문제 출제 시스템을 구현한 시스템[1]에서는 프로세스의 단계가 잘 구분되지 않고, 웹 서버가 클라이언트로부터 요청을 받을 때마다 새로운 프로세스가 만들어지고, 각각의 프로세스는 자신의 환경 변수를 가지게 되어 서버의 자원에 큰 부담이 된다는 단점이 있다. ASP (Active Server Pages)를 사용하여 문제 출제 시스템을 구현한 경우[2]에 있어서, 시스템의 문제점으로서 시스템이 플랫폼에 종속적인 제약이 갖게 되며, 기본적으로 스레드된 두 정보 집합인 스크립트와 HTML을 혼용해서 사용함으로써 유지보수가 어렵다는 점이다. JSP(Java Server Pages)를 이용하여 문제 출제 시스템을 구현한 경우[3]에 있어서는, 플랫폼 종속적인 단점은 극복할 수 있지만 여전히 출제되는 문제의 형태적인 다양성과 시각적으로 유려한 GUI 컴포넌트들에 기반한 사용자 인터페이스의 측면에서는 한계를 지니고 있다.

본 논문에서는 문제지의 작성과 출제된 문제에 대한 응시 그리고 응시 결과에 대한 평가를 수행하는 시스템을 객체 지향 방법론에 기반하여 설계하여, 자바 프로그래밍 언어 (Java Programming Language) [4]와 자바의 GUI API인 Swing 패키지[5]를 이용하여 구현하고자 한다. 본 논문에서 제시하는 원격 시

험 시스템은 문제 출제부, 문제 풀이부, 문제 평가부의 3부분으로 구성하였다. 문제 출제부는 2-tier 구조의 자바 애플리케이션으로 구현하였으며, 문제 풀이부와 문제 평가부는 3-tier 구조의 자바 애플릿으로 구현하였다. 문제 풀이부에서 문제 풀이자는 애플릿을 통해 문제를 풀이하고, 평가부의 서버릿에 의해 생성되는 HTML 페이지를 통해 자신의 성적을 확인할 수 있도록 하였다. 문제 출제자에게 보다 편리한 환경 하에서 문제 출제가 이루어지도록 하기 위해 Java Swing의 사용자 인터페이스를 사용하여 구현하고, 문제 출제 시스템을 통한 간단한 입력으로 멀티미디어 데이터를 입력할 수 있도록 하였다. 본 논문에서 제시하는 시스템은 문제의 출제 및 풀이, 평가에 있어서 안정되고 편리한 사용자 인터페이스를 가지고 있으며, 자바의 Enterprise Java API 접속할 수 있는 확장성을 제공하고 있다. 또한, 기존의 강의의 오프라인 평가 또는 가상대학 시스템과 병행시킬 경우 교육 효과의 상승이 기대된다.

II. 원격시험시스템의 구현

문제 출제 시스템의 구현 단계에서는 구현 모델의 구조에 기반하여 어떻게 시스템을 통합할 것인가를 정의하고, 서브시스템과 컴포넌트의 수준에서 이를 어떻게 통합할 것인가를 정의하고, 코드와 관련 산출물을 개발하고 필요에 따라 코드를 수정하는 등의 실제 구현 작업을 수행한다. 그림 1은 문제 출제 시스템의 전체 구성을 나타낸다. 문제 출제자는 문제 출제부를 통해 문제를 출제하고 출제된 문제는 Java의 데이터베이스 API인 JDBC를 이용해 데이터베이스에 대한 관리를 처리한다. JDBC를 이용하여 데이터베이스 프로그래밍을 함으로써 특정 데이터베이스에 대한 의존성을 극복할 수 있으며, 자바 표준에 준하여 구현이 가능하다[6]. 문제풀이자는 문제풀이를 위해 문제풀이 애플릿에 접속하여 문제를 풀이하고 차트 생성 서버릿에 접속하여 자신의 성적을 확인한다.

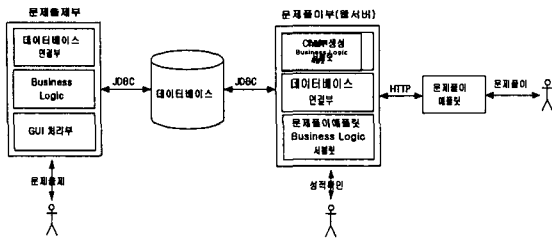


그림 1. 원격시험시스템의 전체구성도
Fig.1 Overview of Remote Examination System

2.1 문제 출제부의 구현

일반적인 2-tier 형태의 클라이언트/서버 모델로 구현된 문제출제부의 구성은 그림 2와 같다. 문제 출제 프로그램은 JDBC API를 사용하여 데이터베이스와의 연결을 설정하고, SQL문을 데이터베이스로 전송하고, 그 결과를 처리하여 문제 출제자에게 출력하게 된다. 문제 출제부는 사용자 인터페이스, 데이터베이스 연결부, 비즈니스 로직(Business Logic) 처리부분으로 구성되어 있다.

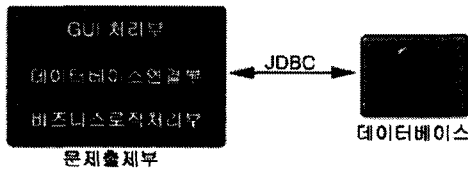


그림 2. 문제 출제부의 구성도
Fig. 2 Diagram of Question Generating Part

JDBC 드라이버에는 4가지 형태의 드라이버가 있으며[6], 본 논문에서의 시스템에서는 타입 4 형태의 드라이버를 사용한다. 타입 4 형태의 드라이버는 순수 자바로 구현되어 있으며 데이터베이스에 직접 접속하게 된다. 타입 4 형태의 드라이버는 성능면에서 타입 1이나 타입 2 형태의 드라이버보다 뛰어나고 서버나 클라이언트에 별도의 소프트웨어를 설치하지 않아도 되며 네트워크를 통해 다운로드 될 수 있다는 장점이 있다.

문제 출제부는 문제 출제자에게 문제 출제의 편리함을 주도록 하기 위해 자바 애플리케이션으로 구현되었다. 애플리케이션을 생성할 때 사용하는 javax.swing.JFrame 클래스의 내부 구조가 그림 3에 보여

져 있다. 스윙 기반 애플리케이션을 구현할 때 사용되는 컨테이너가 JFrame 클래스이다. JFrame 클래스에 어떠한 GUI 컴포넌트를 추가하고자 할 때에는 JFrame에 직접 추가하는 것이 아니라 JFrame의 JRootPane에 컴포넌트를 추가하게 된다. JFrame 클래스는 JRootPane을 프로텍티드(protected) 멤버로 포함하고 있다. JRootPane은 glassPane, layeredPane, contentPane, menuBar와 같은 GUI 컴포넌트들을 포함하고 있다.

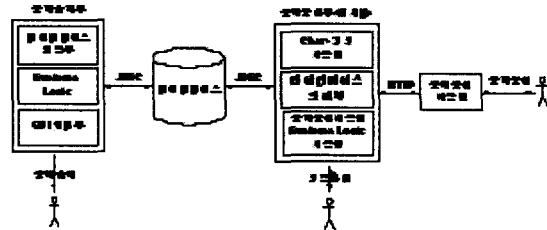


그림 . 원격시험시스템의 전체구성도
Fig.1 Overview of Remote Examination System

문제 출제부의 화면 출력을 위한 패널(Panel)을 그림 4와 같이 설계한다. 문제 출제자가 사용할 로그인, 새문제출제 등과 같은 메뉴들로 메뉴바를 구성하고 문제의 내용 입력과 문제 형태 선택, 멀티미디어 파일의 추가, 정답 입력, 객관식 문제의 경우 항목의 형과 항목의 개수 등을 입력하고 선택할 수 있는 GUI 컴포넌트들을 추가하기 위한 QuestionPanel과 사용자가 선택한 객관식 문제 개수에 따라 각 항목을 입력할 수 있는 GUI 컴포넌트를 추가하는 ItemPanel이라는 두 개의 javax.swing.JPanel 클래스로 구성한다. 각각의 패널을 구성할 때에는 각 패널마다 컴포넌트의 배치를 위해 GridBagConstraints 클래스를 사용하여 각 패널들을 구성하였다. ItemPanel에 있는 컴포넌트들은 문제의 유형과 항목의 개수의 선택에 따라 그 형태를 달리하게 되므로 GridBagConstraints를 사용하여 컴포넌트들을 구성하였다.

문제의 유형과 스윙 컴포넌트의 관계를 아래에 보인다. 문제 출제자가 선택한 형태에 따라 문제풀이자에게 보여지는 GUI 컴포넌트의 형태가 달라지게 된다.

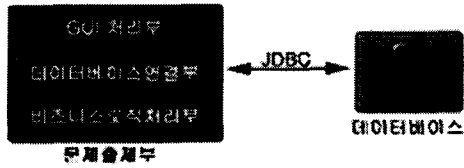


그림 4. 문제 출제부의 구성도
Fig. 2 Diagram of Question Generating Part

문제유형	객관식	기술형주관식	단답형주관식
Swing 컴포넌트	JCheckBox JComboBox JRadioButton	JTextArea	JTextField

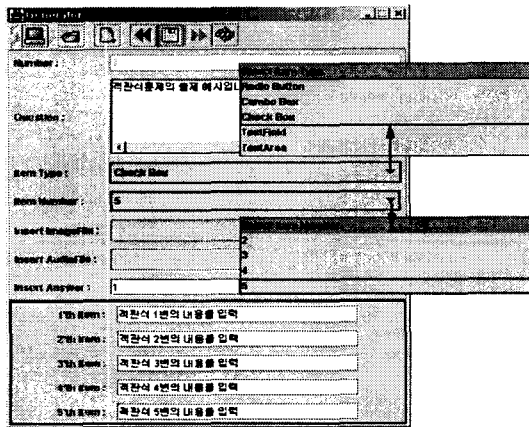


그림 5 객관식 문제의 화면표시
Fig. 5 Screen display of Choice Type Question

객관식 문제를 출제할 때의 문제출제프로그램의 GUI 컴포넌트 배치는 그림 5와 같다. 문제 출제자가 선택할 수 있는 문제의 유형에는 Radio Button, Combo Box, Check Box, TextField와 TextArea의 5가지 종류가 있으며 문제 출제자가 Radio Button, Combo Box, Check Box 중 하나를 선택하면 객관식 문제를 출제할 수 있다. 객관식 문제를 선택한 후에는 객관식 항목의 개수를 선택한다. 객관식 항목의 개수에 따라서 ItemPanel의 객관식 항목을 입력할 수 있는 입력란 개수가 변경된다. 멀티미디어 파일의

입력을 위하여 파일선택 다이얼로그를 사용하여 입력하고자 하는 파일을 선택한다.

문제의 유형 중 TextField와 TextArea를 선택하여 주관식 문제를 출제할 경우 문제출제프로그램의 GUI 컴포넌트 배치는 그림 6과 같다.

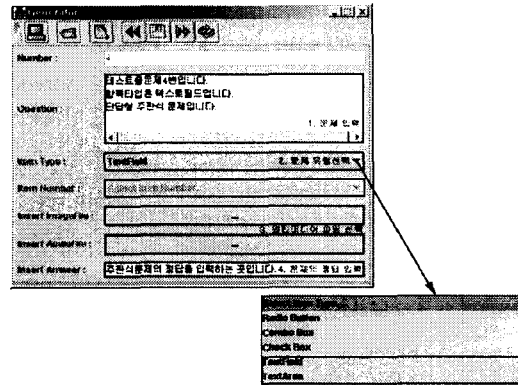


그림 6. 주관식 문제의 화면 표시
Fig. 6 Screen display of Description Type Question

주관식 문제일 경우에는 항목의 개수를 선택할 수 있는 콤보 박스가 비활성화되어 있으며 객관식 문제를 출제할 때 나타났던 ItemPanel은 사라지게 된다.

2.2 문제 풀이부의 구현

문제풀이부는 다음의 그림 7과 같이 웹 애플리케이션의 일반적인 형태인 3-tier 모델로 구성된다[7]. 웹 애플리케이션은 사용자에게서 사용자에게 데이터를 모아 웹 서버에 요청을 보내고, 요청된 서버 프로그램을 실행시켜 웹 브라우저에서 보여줄 수 있도록 데이터를 묶어 브라우저로 다시 보내 화면에 출력되도록 한다.

클라이언트측 프리젠테이션 레이어는 문제풀이 애플릿과 애플릿을 관리할 HTML 페이지로 구성하였다. 문제 풀이자는 HTML 내에 삽입된 자바 애플릿을 통해 문제풀이를 수행한다. 문제 풀이자가 HTTP 프로토콜을 사용하여 접속요청을 하게 되면 웹 서버의 애플릿은 클라이언트측의 웹 브라우저 내에서 실행되게 된다.

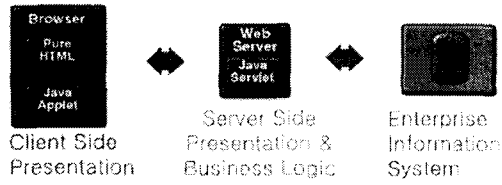


그림 7. 문제 풀이부의 구성도
Fig. 7 Diagram of Question Answering Part

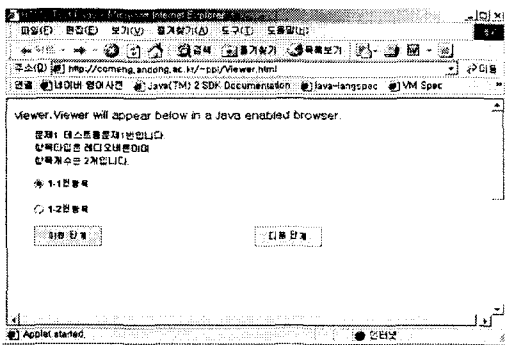


그림 8. 객관식 문제의 출력 화면
Fig. 8 Output display of Choice type Question

객관식 문제가 출제되었을 때의 출력화면은 그림 8과 같다. 객관식 문제의 경우 문제 출제자는 객관식 항목의 개수와 형태를 지정하며 문제 풀이자에게 출력될 때에는 각각의 문제 형태와 객관식 항목에 따라 출력된다.

그림 9는 주관식 문제가 출제되었을 때의 출력화면이다. 주관식 문제의 경우 문제 출제자의 선택에 따라 단답형 주관식과 서술형 주관식 출제된다.

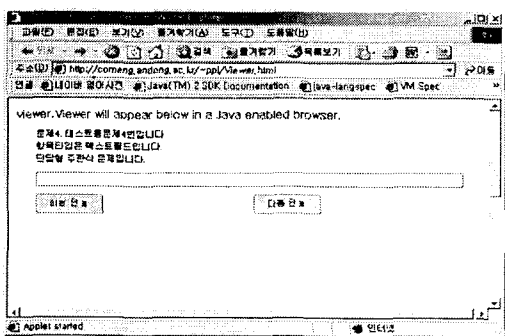


그림 9. 주관식 문제의 출력 화면
Fig. 9 Output display of Description Type Question

그림 10은 데이터베이스에 저장된 멀티미디어 데이터가 문제풀이 애플릿에 출력될 때의 모습이다. 이러한 멀티미디어 데이터를 단순한 URL을 사용해서 출력을 하는 것이 아니라 애플릿의 내부에 출력함으로써 멀티미디어 데이터의 도용을 막을 수 있다.

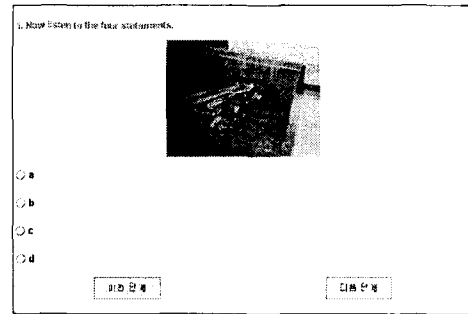


그림 10. 이미지가 포함된 문제의 출력화면
Fig. 10 Output display of Question with Image

2.3 문제 평가부의 구현

문제 평가부에서는 문제 풀이가 종료되면 그 결과를 클라이언트측 프리젠테이션을 담당하는 애플릿과 서버에서 동작하는 서블릿과의 통신을 통해서 결과를 출력하여 준다. 이러한 과정은 먼저 애플릿이 `DataOutputStream` 을 열어 서버에 HTTP 요청을 전송해서 서블릿을 호출하고 애플릿이 서버에 대해 `InputStreamReader`를 열어서 서블릿의 응답을 검색하는 방법을 취한다.

애플릿에서 서블릿으로 쿼리 문자열을 전송하는 방법은 먼저 URL 연결 객체를 만들어 서블릿에 대한 연결을 설정하면서 입력과 출력을 위한 몇가지의 플래그들을 설정하고, 캐시 사용을 제한하며, 요청의 타입을 설정하는 작업을 수행한다. `URLEncoder` 클래스의 `encode()` 메소드를 사용하고, URL 연결 객체를 통해 서버에 대한 `DataOutputStream`을 열고, 여기에 쿼리 문자열을 기록함으로써 애플릿으로부터 서블릿으로 쿼리 문자열을 전송할 수 있다. 다음은 애플릿에서 서블릿으로의 문자열 전송을 코드의 예를 보인다.

```

URL url = new URL("http://ipaddress:port_number/
BusinesslogicServlet");
    
```

```
URLConnection uc = url.openConnection();
uc.setDoOutput(true);
uc.setDoInput(true);
uc.setUseCaches(false);
uc.setRequestProperty("Content-type,
    application/x-www-form-urlencoded");
.....
String qry = URLEncoder.encode(qry) +
    "=" + URLEncoder.encode(qryString);
.....
DataOutputStream dos = new DataOutputStream(
    uc.getOutputStream());

dos.writerBytes(qry);
dos.flush();
dos.close();
```

서블릿으로부터의 결과를 검색해 애플릿에 출력하는 부분의 코드의 예를 보이면 아래와 같다.

```
TextArea taResults;
.....
InputStreamReader in = new
InputStreamReader(uc.getInputStream());
int chr = in.read();
while(chr != -1) {
    taResults.append(String.valueOf((char)chr));
    chr = in.read();
}
in.close();
```

자바 서블릿은 HTML이나 XML과 같은 텍스트 데이터만 브라우저에 보낼 수 있는 것이 아니다[8]. 서블릿을 사용해서 동적으로 만들 수 있는 데이터 형식은 매우 다양하다. 텍스트 객체가 아닌 가장 일반적인 객체는 이미지이다. 본 논문에서 개발하는 서블릿은 데이터를 HTML표현 아니라, 서블릿에 의해 동적으로 생성되는 차트를 보여준다. 자바에서 표현되는 이미지는 java.awt.Image 클래스를 사용하여 구현된다. RGB 컬러 모델로 픽셀로 된 이미지를 만드는 간단한 방법은 MemoryImageSource 객체를

사용하여, java.awt.Toolkit의 createImage() 메소드로 이미지를 만드는 방법이다. 다음은 AWT Toolkit을 사용하여 이미지를 생성시키기 위한 코드부분을 보인다.

```
init width = 100, height = 100;
int[] pixels = new int[width * height];
ImageProducer source = new MemoryImageSource
(width, height, pixels, 0, width);
Toolkit defaultToolkit = Toolkit.getDefaultToolkit();
Image img = defaultToolkit.createImage(source);
```

본 논문에서는 동적 콘텐츠를 생성하기 위하여 ChartServlet 클래스를 설계한다. 차트를 그리기 위한 차트 서블릿은 높은 수준의 그래픽 메소드들로 그릴 수 있는 이미지들을 사용해서 구현한다. ChartServlet 클래스는 문제 풀이자가 통계자료에 대한 요청을 하게 되면 그 요청을 처리한다. Stats 클래스는 데이터 베이스에 연결되어 문제 풀이자에 대한 시험 결과를 ChartServlet 클래스에 전달한다. 전달된 결과를 사용하여 ChartServlet 클래스는 화면 프레임에 차트를 그리게 되고 이렇게 생성된 차트는 GifEncoder 클래스를 통해 Gif 형식으로 변환된다. 변환된 이미지는 다시 ChartServlet 클래스를 통해 문제 풀이자에게 전송된다.

다음은 Stats 객체 내의 옵션 문자열을 생성하는 부분의 코드를 보인 것이다.

```
public Stats(ServletRequest req) throws
ServletException {
    try {
        int num =
Integer.parseInt(req.getParameter("num"));
        values = new StatValue[num];
        for(int i=0; i<values.length; i++) {
            String s = req.getParameter("v"+i);
            int sep = s.indexOf('|');
            values[i] =
new StatValue(s.substring(0, sep),
Integer.parseInt(s.substring(sep+1)));
        }
    }
```

```

    } catch(Exception e) {
        throw new ServletException("Error parsing
chart values: "+ e.toString());
    }
    process();
}

```

먼저, ServletRequest의 getParameter 메소드를 사용하여 num 옵션을 읽고 적절한 크기로 실제 데이터의 StatValue 배열을 만든다. 이 글자 이전의 것들은 모두 새로 만들어진 StatValue 객체의 이름 부분을 구성하고, 옵션 구분자 다음에 나오는 것들은 모두 java.lang.Integer 클래스의 parseInt() 메소드를 사용해서 String값에서 int값으로 변환된 값이 놓이게 된다.

ChartServlet의 doGet() 메소드에 문제 평가 내용에 대한 HTML을 생성해내는 코드는 아래와 같이 구현된다.

```

public void doGet(HttpServletRequest req,
HttpServletResponse res)
throws ServletException, IOException {
    .....
    Stats stats = new Stats();
    if(req.getParameter("num") != null)
        stats = new Stats(req);
    else stats = new Stats(req.getPathInfo());
    boolean bw =
        Boolean.valueOf(req.getParameter("bw"))
        .booleanValue();
    String img = req.getParameter("img");
    if(img != null) {
        .....
    } else {
        .....
        out.println("<HTML><HEAD><TITLE>Statistics
</TITLE></HEAD>");
        .....
        out.println("<P><TABLE border=1
cellpadding=2><TR><TH>Name</TH> +
<TH>Count</TH><TH>%</TH></TR>");

```

```

.....
out.println("</P><P><IMG
src=\""+req.getRequestURI()+
\"?img=pie&bw="+bw+"&"+statsString+"\">");
        out.println("<IMG
src=\""+req.getRequestURI()+
\"?img=bar&bw="+bw+"&"+statsString+"\"></P>");
;
out.println("</BODY></HTML>");
out.close();
}
}

```

ChartServlet의 doGet() 메소드의 결과로 출력되는 페이지는 시험의 결과를 출력해 주는 테이블과 두 개의 삽입된 이미지를 포함한다. 그림 11은 ChartServlet에 의해 생성된 테이블과 이미지의 동적 콘텐츠를 보여준다. 평가의 결과값에 상응하는 파이 차트와 바 차트가 이미지 인코더에 의해 생성되어 사용자에게 출력된다. 여기서 보는 바와 같이 서블릿은 자바 언어의 모든 부분에 접근할 수 있도록 해주기 때문에 자바 AWT 그래픽 메소드들을 사용해서 브라우저에 다운로드할 수 있는 이미지들을 만들 수 있는 장점이 있다.



그림 11. 문제 평가부의 화면출력
Fig. 11 Output display of Evaluation Part

III. 결 론

본 논문에서는 문제지의 작성과 출제된 문제에 대한 응시 그리고 응시 결과에 대한 평가를 수행하는 시스템을 Java와 Java의 GUI API인 SWING 패키지를 이용하여 구현하였다. 본 논문에서 제시하는 원격 시험 시스템은 문제 출제부, 문제 풀이부, 문제 평가부의 3부분으로 구성된다. 문제 출제부는 2-tier 구조의 자바 애플리케이션으로 구현하며, 문제 풀이부와 문제 평가부는 3-tier 구조의 자바 애플릿으로 구현하였다. 문제 풀이자는 문제 풀이부의 애플릿을 통해 문제를 풀이하고, 평가부의 서버릿에 의해 생성되는 HTML 페이지를 통해 자신의 성적을 확인할 수 있도록 하였다. 본 논문에서 제시하는 시스템은 문제의 출제 및 풀이, 평가에 있어서 안정되고 편리한 사용자 인터페이스를 가지고 있으며, 자바의 Enterprise Java API 접속할 수 있는 확장성을 제공하고 있다. 또한, 기존의 강의 평가 방법 또는 가상대학 시스템과 병행시킬 경우 교육 효과의 상승이 기대된다.

참고문헌

[1] 오민권, 윤여창, 웹을 이용한 온라인 학습 평가 시스템, 한국정보처리학회 2000 춘계학술발표논문집 제7권 제1호, pp.710-714, 2000.

[2] 박기석, 이재영, 김동환, 자바를 이용한 웹 기반의 문제 출제 시스템, 한국정보과학회 '99 가을 학술발표논문집, pp.823-826, 1999.

[3] 최돈은, 서현진, 박기석, 이재영, 동적인 문제출제 시스템의 설계 및 구현, 한국정보과학회 2000 춘계학술발표논문집, pp.345-348, 2000.

[4] J. Gosling and B. Joy, The Java Specification, Addison-Wesley, pp. 1-6, 1996.

[5] D. M. Geary, Graphic JAVA, Addison-Wesley, 3rd Ed., pp. 3-24, 1999.

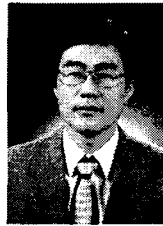
[6] White, et al., JDBC API Tutorial and Reference: Universal Data Access for the Java 2 Platform, Addison-Wesley, 2nd Ed., pp. 13-24, 1999.

[7] Y. P. Shan and R. H. Earle, Enterprise

Computing with Objects: From Client/Server Environments to the Internet, Addison-Wesley, pp. 10-13, 1998.

[8] T. Neward, Java Server-Based Java Programming, Manning, pp. 283-298, 2000.

저자소개



김윤호(Yun-Ho Kim)

1983. 2. 경북대학교 전자공학과 공학사

1993. 2. 경북대학교 대학원 컴퓨터공학과 공학석사

1997. 2. 경북대학교 대학원 컴퓨터공학과 공학박사

1997. 8.-현재 안동대학교 전자정보산업학부 조교수
 ※관심분야: VLSI 구조 및 알고리즘, 분산병렬처리, 인터넷 컴퓨팅, 객체지향 분석/설계/프로그래밍

박운영(Un-Young Park)

2000. 3. 안동대학교 컴퓨터공학과 공학사

2002. 2. 안동대학교 컴퓨터공학과 공학석사

2002. 3-현재 컴포넌트뱅크 연구원

※관심분야: 객체지향, 인터넷컴퓨팅