

文章编号: 1005-0523(1999)02-0055-06

# 基于 Web 的计算机辅助测试软件的实现

丁振凡

(华东交通大学 现代教育技术中心, 江西 南昌 330013)

**摘要:**给出了一个用 J++ 编写的计算机辅助测试软件的实现方案, 该软件在 Web 环境下使用, 分教师操作和学生操作二部分, 教师可进行试题库维护、安排测试等, 学生可进行操练和测试<sup>19</sup>。

**关键词:**Java; Web; 计算机辅助测试

**中图分类号:** TP 391.6      **文献标识码:** A

## 0 引言

目前, 大部分计算机辅助测试软件是基于单机或 Novell 网环境, 如果能将测试安排在 Internet 上进行应该是一件有意义的事情, 由于 Internet 将全世界连接在一起, 做题者可不受地理位置的限制, 应用范围得到最大的扩展, 环球信息网(WWW) 已经成为 Internet 上最具生命力的一部分, 其 Browser/Server 结构可最大程度减轻客户端的安装和配置, 将计算机辅助测试建立在基于 Web 的环境无疑是最好的选择, Java 的出现给 WWW 世界带来了活力, Java Applet 在浏览器端运行能最好地满足 WWW 世界的交互性要求, 本文中介绍的计算机辅助测试系统是一个框架系统, 所有测试内容可在实际使用中由教师不断添补和完善<sup>19</sup>。

## 1 系统的功能结构

### 1.1 教师操作部分

依应用界面的菜单结构有如下 5 项功能:

- 1) 题库维护: 包括对试题库的增、删、改、查 4 项功能和知识点的维护<sup>19</sup>。
- 2) 测试设置: 对组卷方式、组卷参数、用户认证方式、考试时间进行设定<sup>19</sup>。
  - a. 组卷方式设定

分固定试卷和临时自动组卷 2 种方式, 固定试卷方式所有学生用同一份试卷, 临时自动组卷则每个学生试卷不同, 但考核要求一样, 系统随机地从试题库抽题组卷, 固定组卷有 3 种选择, 第 1 种是从已生成的试卷库中选取一份; 第 2 种是教师从试题库中浏览选题组成一份新试卷; 第 3 种是根据考试要求自动从试题库选题产生一份试卷<sup>19</sup>。

- b. 自动组卷参数录入

**收稿日期:** 1998-11-21; **修订日期:** 1998-09-04

**作者简介:** 丁振凡(1965-), 男, 江西丰城人, 华东交通大学副教授

对要考核知识点、题量、分数、难度要求进行设定,自动组卷时将根据这些参数的要求从题库中随机选题组卷<sup>19</sup>。c. 用户认证选择

有开放式考试和限制式考试两种,开放式考试任何用户均可进入系统使用,且可以多次进入考试;限制式考试则要进行用户口令验证,且每个用户只限进入考试一次<sup>19</sup>。

#### d. 考试时间限制

让学生在规定的测试时间内完成考试<sup>19</sup>。

3) 学生管理:提供用户的增、删管理,为用户设定初始密码<sup>19</sup>。

4) 成绩查询:包括查每个学生的考试总成绩,每道题的答题得分,教师借此可了解学生的知识掌握情况<sup>19</sup>。采用临时组卷方式时,将只登记总分,不登记每题得分<sup>19</sup>。

5) 统计分析:用于固定试卷方式,根据考试成绩登记对试题的难度、区分度、信度、平均分、标准差等进行统计计算,通过这些指标评价试卷的合理性、科学性,将不合理的试题删去<sup>19</sup>。

## 1.2 学生操作部分

学生操作部分有练习、测试、口令更改共三大功能,练习时将提示学生答题的对错,在两次回答出错后,系统自动给出正确答案,系统中的试题是按知识点进行组织,在练习时学生要选择知识点,系统调出与该生水平相当的试题供其操练,在练习时学生的水平根据学生答题的正确性动态调整<sup>19</sup>。考试则不给学生任何反馈,由学生自由翻动试题去做题,只有在考试结束时才会告诉学生成绩,考试的出卷方式和安全限制取决于教师的设定<sup>19</sup>。

## 2 系统的数据访问结构

该系统是基于 Web 环境下使用,无论是教师方还是学生方,都涉及如何在浏览器上访问数据库中试题的问题<sup>19</sup>。Visual J++ 为 Microsoft 公司推出的 Java 开发环境,它针对 Microsoft 操作系统对 Java 类库做了专门的扩展,它将 Java 与 COM(组件对象模型)和支持 Java 的 Internet Explorer 3.0 浏览器相集成,这样,用户可在 Java 程序中访问任何支持 COM 的软件,对于数据库的访问,在 J++

中提供了 DAO(Data Access Object)和 RDO(Remote Data Objects) 2 种方法<sup>19</sup>。本软件中采用 RDO 对象实现对 ODBC 数据源数据的访问<sup>19</sup>。由于 Java 的安全性限制,在客户端浏览器中执行的 Java Applet 直接访问数据库有很大限制,笔者采用的方案是将应用分为 2 部分,即由 Web 服务器方的应用和程序浏览器客户方的 Java Applet 组成,所有应用软件都放在 Web 服务器上,服务器方的 Java 应用程序在 Web 服务器上运行,用户通过浏览器访问含 Applet 的页面将客户部分下载到客户方运行,客户方与服务方通过 Socket 通信交换信息,服务端的 Java 应用程序通过 RDO 访问 ODBC 数据源对应数据库的试题内容(如图 1)<sup>19</sup>。教师方和学生方都是采用该方式实现对数据库的访问操作<sup>19</sup>。

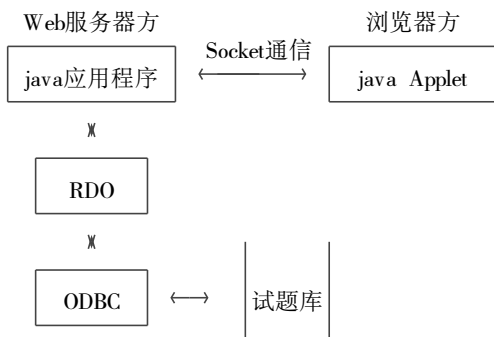


图1 系统的数据访问结构

## 3 技术实现要点

### 3.1 J++ 对数据库访问方法

首先,在 Java 程序中须加入如下两个 import 语句:

```
import msrdo20.*;
```

```
import com.ms.com.*;
```

以便能使用 RDO 对象和引用 Variant 类型的变量<sup>19</sup>.

存取数据库表格的数据内容可按如下步骤进行<sup>19</sup>.

#### 1) 建立对数据库的访问连接

通过 rdoConnection 对象 putConnect 方法规定要连接的 ODBC 数据源,并通过其 EstablishConnection 方法创建与数据源的连接<sup>19</sup>.

#### 2) 建立访问记录集

要实现数据库表格的访问是通过使用连接对象的 OpenResultSet 方法创建一个结果记录集,在该方法中要规定结果记录集的类型和加锁类型(见文献 2)<sup>19</sup>.

#### 3) 记录集的使用

##### a. 移动记录游标

提供有如下方法:MoveFirst(); MoveNext(); MovePrevious(); MoveLast() 用于移动记录游标,并结合 getBOF()、getEOF() 方法检测游标是否在文件的开始和结束位置,通过这些方法可实现对记录集的遍历访问<sup>19</sup>.

##### b. 对记录集的操作方法

对记录集的操作分为增、删、改、查四个方面,增加记录使用 AddNew() 方法结合 Update() 方法;删除记录使用 delete() 方法;修改记录使用 Edit() 方法结合 Update() 方法;查询记录仅是读取各数据项的值<sup>19</sup>在增加记录和修改记录时还涉及往记录的数据项中写入数据,对数据项的读写访问方法见 4)<sup>19</sup>.

#### 4) 对数据项的访问

##### a. 取得要访问的数据项对象

首先要通过记录集的 getrdoColumns() 方法获取栏集合,而后再从栏集合中获取指定名字的栏(或称数据项),从栏集合中取某栏要使用 getItem 方法,其中要用 Variant 类的对象作为参数,例如,我们要取得字段名为"score"的数据项对象 c,相应代码段如下:

```
rdoColumns columns = ResultSet.getrdoColumns(); // 取栏集合
```

```
Variant name1 = new Variant();
```

```
name1.putString("score");
```

```
rdoColumn c = columns.getItem(name1);
```

##### b. 读取某数据项对象的数据值

使用数据项对象的 getValue() 方法可读取数据项的值,但该值为 Variant 类型,为了在用户界面显示该值,需将 Variant 类型的值转换为数据项的真实数据类型,如转换为字符串类型用 value.toString() 方法<sup>19</sup>.

### c. 将数据写入数据项中

使用数据项对象的 putValue() 方法可将 Variant 对象的值写入数据项中,这里要注意给 Variant 类的对象 value 赋值时,所赋的数据要符合数据库中数据项的格式,如:写入日期型数据用 value.putString(“95-4-1”);写入逻辑型数据用 value.putBoolean(true);

## 3.2 浏览器客户方与 Web 服务器方的通信

Web 服务器方的 Java 应用程序与浏览器方的 Java Applet 程序之间是通过 Socket 通信交换信息,Socket 通信是一种可靠的基于连接的通信方式,Java 提供的 Socket 类与 ServerSocket 类为编写基于 Socket 的应用提供了很大的方便,通过使用封装在这两个类中的方法,简化复杂的 Socket 应用的实现,Web 服务器方的 Java 应用程序作为 Socket 通信的服务方要先运行,服务器方创建一个 ServerSocket 对象时要指定一个端口号,调用 ServerSocket 对象的 accept() 方法可进入到监听等待客户连接状态,客户方在创建 Socket 对象时要指定服务器地址和端口号以便与服务器方建立连接,在双方连通后,服务方和客户方分别有一个 Socket 对象,Java 的所有输入/输出操作是基于流的方式,Socket 通信也不例外,使用 Socket 对象的 getInputStream() 和 getOutputStream() 方法可分别取得其输入流和输出流,为了方便操作要创建过滤流对流进行过滤处理<sup>19</sup>.通常,对于字符串的输出可使用 PrintStream 流的 println 方法往流写一行数据;从流读取一行字符串可使用 DataInputStream 流的 readLine() 方法,不幸的是,它们在处理到汉字字符时会过滤掉汉字的高位字节,因此给汉字字符传送带来不便,笔者的解决办法是将字符串中的每个字符转化为整数,使用 DataOutputStream 中的 writeInt() 方法分别传送,在字符串的结尾引入了一个结束标记符以便识别处理;在接收方再用 dataInputStream 流的 readInt() 方法读取各个整数反转化为字符,并拼装为字符串<sup>19</sup>由于一道试题数据量不是很大,所以传送速度还算快<sup>19</sup>.

## 3.3 多用户连接访问处理

### 3.3.1 多线程设计

为了使该软件支持多用户的使用,服务器端在设计上采用了多线程机制(见图 2),其 Java 应用程序由一个连接监听线程和若干通信线程组成,连接监听线程负责监听新用户的 Socket 连接申请,它将为每一个连接用户创建一个通信线程,通信线程与客户端之间的信息交换是采用客户/服务器方式,每个通信线程分别与数据库建立连接并进行访问,相互间不干扰;通信线程根据客户端发送过来的访问请求去访问数据库并将访问结果通过 Socket 通道送客户方显示<sup>19</sup>.

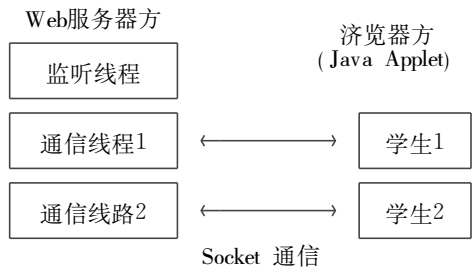


图2 多用户访问处理

### 3.3.2 访问效率问题

服务方的访问效率是软件设计的一个关键考虑之处,当用户连接数较多时,服务器将成为影响速度的访问瓶颈,为此,在软件设计上笔者采取了一些措施减轻服务端的负担<sup>19</sup>.

#### 1) 采用本地缓存减少对服务器的访问

从客户端访问数据库中的试题要经过多个环节,首先客户方往服务方发送访问消息,服务器上 Java 程序接到请求后,从数据库中取数据,而后将数据通过 Socket 通路传到客户端,事

实上,用户做题时经常可能前后翻动试题,同一道试题要多次访问,如果每次都从服务器的数据库中读取,整个系统的效率将受影响,为此本软件中采用了本地缓存,引入若干向量(也可用数组,但数组有越界问题,通用性差)分别用来存放各试题的内容、标准答案、学生解答、分数,同时引入一个标记向量记下各编号的试题是否从数据库读取过<sup>19</sup>每次要显示一道题,首先检查该序号题的标记向量值,如果未从数据库读取过则访问数据库,并将访问结果写入各向量中,否则就从各向量中取数据<sup>19</sup>该办法可减轻服务器的访问数据库负担,节省通信开销,并且客户端的试题显示速度快<sup>19</sup>。

## 2) 注意服务端的资源释放

每个用户登录后,在服务端要占有的一些资源(如 Socket 插座、通信线程),当用户退出时,必须保证这些资源的释放,由于 HTTP 的无状态性,当客户浏览器关闭时服务方不能直接知道,注意到客户关闭浏览器时,将执行 Applet 的 stop() 方法,我们可在该方法中安排一个语句发送“退出”消息通知服务方,服务器方与该用户相应的通信线程接收到该消息后再释放相关的资源,并停掉与用户对应的通信线程<sup>19</sup>。

## 3.4 学生做题界面设计

整个系统的界面含教师操作界面和学生操作界面两部分,教师操作界面设计为窗口菜单结构,具体功能较多,限于篇幅不作介绍,学生操作界面比较简单,由登录和答题界面构成<sup>19</sup>。

### 1) 登录界面

登入界面取决于教师的系统设置,对于开放式考试,只要求用户输入用户名、学号;对于限制式考试,则要求用户输入用户名、学号和口令,而后点击“登录”按钮即可进入做题<sup>19</sup>。

### 2) 测试做题界面

学生测试界面满足了学生答题时的各种选择要求,整个界面由三块面板构成,基本功能是:可自由切换题型;解答试题方便;可随意翻动试题;动态地显示考试剩余时间<sup>19</sup>。

① 题型切换面板:位于屏幕的最顶部,面板含三个按钮(分别为“判断题”、“填空题”、“结束”)和一个显示考试剩余时间的标签,点击“判断题”和“填空题”按钮可在两类试题间切换,点击“结束”按钮将弹出对话框显示学生的得分,该生的姓名、总分、每道题的得分写入服务器端的成绩登记表中<sup>19</sup>。

② 试题内容面板:位于屏幕的中部,该面板顶部为一个标签显示当前试题编号,中央为文本域显示试题内容,底部供用户录入解答,“判断题”和“填空题”在录入解答形式上不同,前者为“对”、“错”选项组,后者为文本框<sup>19</sup>。

③ 翻动试题面板:位于屏幕的最下部,该面板上有“第一题”、“前一题”、“后一题”、“最后一题”等按钮用于前后翻动试题<sup>19</sup>。

## 4 结束语

该软件可作为一个安排 Web 环境下进行辅助测试的工具,具有很大的实用性,笔者已将其用于 Java 语言的教学测试,该软件在技术上有如下特点:

1) 采用基于 Web 的应用技术,所有软件的维护、更新都集中在 Web 服务器上,对应用的管理、升级方便,客户端只要求有支持 Java 的浏览器,教师可在浏览器上对系统进行远程维护

和安排测试,学生通过浏览器进行交互式做题<sup>19</sup>。

2) 操作界面友好,学生可通过点击翻动试题的按钮前后浏览试卷内容,并可在两种类型试题间随时切换,学生的测试有时间限制,测试时运行的计时线程负责显示剩余时间,当时间用完时自动结束学生的测试<sup>19</sup>。教师管理界面也同样方便实用<sup>19</sup>。

3) 试题库内容便于维护和扩充,一般地,一个试题库的建立要经过长期的使用完善过程,以保证试题库中试题的丰富性、正确性、新颖性,该系统能满足此要求,教师可对试题库中的内容进行增、删、改操作,该软件是通过 ODBC 数据源访问数据库,可采用任何支持 ODBC 的数据库,容易利用已有的试题库<sup>19</sup>。

4) 访问速度快,服务器端采用了多线程设计,注意资源回收管理;客户端采用向量保存读过的试题,避免了对数据库的重复访问,这些措施可有效地提高整个系统的效率<sup>19</sup>。

5) 有灵活的组卷方式和用户认证选择,多种组卷形式可满足不同的组卷要求,根据考试的重要性和保密性要求的不同,系统提供有开放式和限制式两种用户认证方式供选择<sup>19</sup>。

目前系统只支持是非判断题,单项填空题(选择题可视为此类),如何支持更多的试题类型(如:多项填空、图形题等)是有待进一步考虑的问题<sup>19</sup>。

### [ 参 考 文 献 ]

[1] 美 Jamie Jaworski 著,曹康等译. Java 开发指南[M]. 北京:中国水利水电出版社,1997

[2] 谭真等. 使用 J++ 在 Web 页面中访问数据库[J]. 计算机与通信,1998,(7) 20~22

## Implementation of Computer-assisted Test Software Based on Web

DING Zhen-fan

(Modern Educational Technology Center, East China Jiaotong University, Nanchang 330013, China)

**Abstract:** This paper presents the implementation of computer-assisted test software written by J++ and used under web environment. The software falls into two parts, teachers operating part and students' operating part, while it enables teacher to manage the test questions database and arrange the tests, it enables students to do the exercises and the tests.

**Key words:** Java; web; computer-assisted test