基于Struts的在线考试系统设计与优化

周四维
(湖北大学知行学院 计算机科学系,湖北 武汉 430011)

摘 要:使用JSP技术配合Struts三层架构设计在线考试系统,包括数据库、数据库连接池、登录模块、抽题模块和考试模块。总结了当前类似在线考试系统存在的账号同时在线问题、用户键盘的误操作问题、题型灵活性设置问题,并提出了一套修改服务器端脚本、数据库关系与客户端脚本的解决方案。

关键词:Struts;数据库;数据库连接池;在线考试系统
中图分类号:TP311.52 **文献标志码**:A **文章编号**:1673-0143(2014)05-0040-06

Design and Optimization of Online Examination System Based on Struts

ZHOU Siwei
(Department of Computer Science, Zhixing College of Hubei University, Wuhan 430011, Hubei, China)

Abstract: The online examination system was designed based on Struts using JSP with three-tier, including database, database connecting pool, login on module, choosing question module and test module. The problems of current examination system such as the account existing at the same time, the user keyboard misuse, the flexibility for setting questions were summarized. The solution obtaining modified server-side scrpting, database relationships and client-side scripting was proposed.

Keywords: Struts; database; database connecting pool; online examination system

0 引言

随着计算机科学技术的飞速发展和互联网的应用范围不断扩大,以计算机为辅助手段的网络考试方式已经在社会众多领域中逐步得到应用与推广,尤其在各种培训、教育教学领域发展迅速。它使培训者、教育者从出题、组卷、组织考试、阅卷评分、试卷分析等费时费力的传统工作中解脱出来,使他们能够将主要的精力转移到利用现代化的科技手段提高教学效率和教育质量的改革中去,以更好地适应现代教育形势的发展。

Struts是Apache基金会Jakarta项目组的一个开源项目,它采用的是MVC模式,能够更好地帮助Java开发者利用J2EE[1]框架开发Web应用,能够大量地减少开发Web应用的时间,提高产品的重用度[2]。Struts主要采用Servlet与JSP相结合的技术,它把Servlet、JSP、自定义标签和信息资源整合到一个统一的框架中[3],Struts只有一个中心控制器,采用XML定制转向的URL,采用Action层来处理逻辑,开发人员利用其进行开发时无需使用自己的编程语言就能实现全套MVC模式。

1 系统设计

1.1 数据库设计

数据库设计有用户表、题目表、答案表、考卷表、成绩表。用户表记录学生与老师、管理人员的信

收稿日期:2014-05-27
作者简介:周四维(1982—),男,讲师,硕士,研究方向:知识工程与并行计算。

息，允许增删查改。题目表存放系统考试所涉及的题目与用户相关信息，允许增删查改。答案表与题目表主键关联，记录对应题目的答案。考卷表记录当前设定的考卷所包含的题目所对应的相关信息，允许增删查改。成绩表记录学生考试成绩信息。数据表关系见图1。

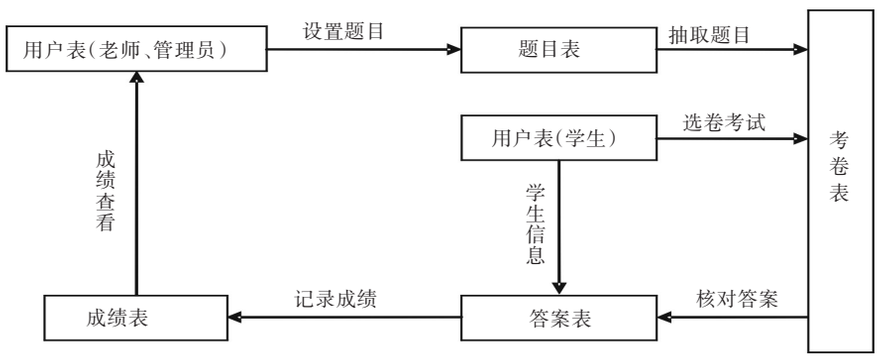

图1　数据表关系图

Fig. 1　Relation of datasheet

1.2 数据库连接池

在Struts中配置一个名为pool.xml的文件，该文件记录与之相关联的数据库信息(包括驱动文件信息、数据库文件信息、数据库用户信息及相关的最大连接数、延迟等参数信息)，它允许同时配置多条数据库连接驱动以适应在不同环境下的数据源[4-5]。以下文件为系统配置了两个数据源的连接驱动，其中一个为微软的SQL Server数据库，另一个为Oracle数据库。

```
<something-else-entirely>
    <proxool>
    <alias>ZX_onlineExam_SQLServer</alias>
    <driver - url>jdbc: microsoft: sqlserver://localhost: 1433; DatabaseName=online_exam; Select-
Method=cursor</driver-url>
    <driver-class>com.microsoft.jdbc.sqlserver.SQLServerDriver</driver-class>
    <driver-properties>
      <property name="user" value="sa"/>
      <property name="password "value="123"/>
    </driver-properties>
    <maximum-connection-count>10</maximum-connection-count>
    <minimum-connection-count>2</minimum-connection-count>
    <house-keeping-sleep-time>60000</house-keeping-sleep-time>
    <maximum-active-time>900000</maximum-active-time>
    <maximum-new-connections>1</maximum-new-connections>
    <prototype-count>1</prototype-count>
    <test-before-use>true</test-before-use>
    <house-keeping-test-sql>select 1</house-keeping-test-sql>
    </proxool>
    <proxool>
    <alias>ZX_onlineExam_Oracle</alias>
    <driver-url>jdbc:oracle:thin:@127.0.0.1:1521:crmdb</driver-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <driver-properties>
```

```
    <property name="user"value="sa"/>
    <property name="password"value="123"/>
  </driver-properties>
  <maximum-connection-count>30</maximum-connection-count>
  <minimum-connection-count>1</minimum-connection-count>
  <house-keeping-sleep-time>60000</house-keeping-sleep-time>
  <maximum-active-time>900000</maximum-active-time>
  <maximum-new-connections>1</maximum-new-connections>
  <prototype-count>1</prototype-count>
  <test-before-use>false</test-before-use>
  <house-keeping-test-sql>select CURRENT_DATE</house-keeping-test-sql>
 </proxool>
 </something-else-entirely>
```

如果有需要也可增加其他的数据源，只需在<proxool>与</proxool>间再次声明即可。

1.3 系统的Struts框架结构

本系统采用三层模式(如图2所示)，前台表示层包括用户数据与前台页面，中间层为Action层，所有的主要逻辑都集中在这里，后台层为数据库[6]。前台表示层与中间层通过Form类与Dto类进行必要的数据传输，而Struts.xml文件负责配置它们与中间层的关系[7]。中间层与数据库之间通过Dao驱动类(包含多种不同类型的数据库数据驱动)进行必要的数据交换，而Proxool.xml文件负责管理Dao驱动类使用哪些驱动[8-9]。

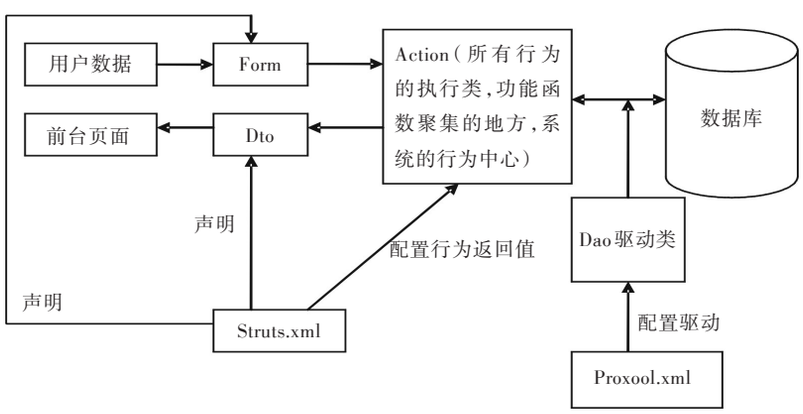

图2 系统Struts框架结构图

Fig. 2 System frame of Struts

1.4 主要功能模块设计

1.4.1 登录模块 新建UserForm与UserDto两个类，声明相关的用户属性为成员变量，设置Get()与Set()方法，配置

```
<form-beans>
        <form-bean name="USER_INFODFORM"
        type="com.exam.cms.form.USER_INFODFORM" />
</form-beans>
```

其中UserForm为封装的用户信息类，用于与网页间的传送；UserDto为Action里调用用户信息类，在Action中判断用户信息即可。

1.4.2 抽题模块 新建SubjectForm与SubjectDto两个类，声明相关的题目属性为成员变量，设置Get()与Set()方法，配置

```
<form-beans>
        <form-bean name="SUBJECT_INFODFORM"
        type="com.exam.cms.form. SUBJECT _INFODFORM" />
</form-beans>
```

其中SubjectForm为封装的题目信息类，用于与网页间的传送；SubjectDto为Action里调用题目类，负责与数据库进行数据传输，在Action中将数据库题目表的数据传给SubjectDto，然后SubjectDto传给SubjectForm并返回于页面中，老师或管理人员将选中的题目再以SubjectForm的形式传给Action，Action将SubjectForm转为SubjectDto，由SubjectDto将所选题目数据传于数据库考卷表。

1.4.3　*考试模块*　新建EXAMDEFForm与EXAMDEFDto两个类，声明相关的题目属性为成员变量，设置Get()与Set()方法。新建ANSWERForm与ANSWERDto两个类，声明相关的题目答案属性为成员变量，设置Get()与Set()方法。新建MARKForm与MARKDto两个类，声明相关的题目答案属性为成员变量，设置Get()与Set()方法，配置

```
<form-beans>
        <form-bean name="EXAMDEFForm"
        type="com. exam.cms.form.EXAMDEFForm" />
<form-bean name="ANSWERForm"
        type="com. exam.cms.form. ANSWERForm" />
<form-bean name="MARKForm"
        type="com. exam.cms.form. MARKForm "/>
</form-beans>
```

其中EXAMDEFForm为封装的试卷题目信息用于网页间的传送，ANSWERForm为封装的设置题目答案信息用于网页间传送（设置答案模块中），MARKForm为封装的成绩信息用于网页间传送。EXAMDEFDto为Action里调用试卷题目类，负责与ANSWERDto相关联判断答题是否正确，其逻辑过程为将学生完成的试卷题目通过EXAMDEFForm传给Action，Action将其转为EXAMDEFDto，并从数据库中答案表找来相关数据并赋予ANSWERDto，然后两者结合判断正误，将最后结果返回给MarkDto，MarkDto将数据返回给数据库中成绩表。

2　系统优化

2.1　当前系统存在的问题

随着在线考试系统的流行，大多数院校在大规模的实测与应用之后，发现了一些设计系统时考虑欠妥的地方，主要有：① 存在考生使用他人账号与本人账号同时在线，提交成绩后本人账号即退出考试，这样就形成了一个可以替考的漏洞。② 在线考试系统大都是以网页的形式输出的，所以很多快捷键可以直接对当前网页进行操作，例如F5刷新，一旦考生在考试时误按了，当前网页就会重置，原来做的答案就会消失，耽误考试时间，导致考试行为异常。③ 一般考试系统试题由题目和答案组成，题目不分题型统一放在一个数据库字段里，这样会造成在输出时由于字段内容大小不一造成格式混乱的现象，整体输出效果较差，最主要的是对于题目选项不能合理设置，所以一般输出的题目都是四个选项的单选，一个空的填空等比较固定的格式，这样对出题的老师存在一定的限制性。针对用户在线状态的判断、用户误操作带来的考试过程失效、系统题型的不灵活这3个问题，给出3个层面上的解决方案。

2.2　服务器端脚本的优化

服务器端脚本的优化主要针对用户在线状态的判断，一般的在线考试系统要么没有判断用户是否在线的功能，要么是基于数据库字段的方法。如没有判断用户是否在线就可能出现代考或出现一人多份成绩的情况；而基于数据库字段判断，则可能出现异常掉线的状况，那么该用户会处于无法登录的状态，只有管理员修改数据库字段值的情况下才能重新登录。

笔者建议在Struts里Web.xml文件设置一个监听器，利用HttpSessionEvent对象结合哈希表来实现

判断用户在线状态[10]，原理为：在Action里创建一个全局的哈希表，当有客户端连接服务器且通过了系统的账号认证时，首先搜索哈希表里是否有此账户的ID信息，如果有则返回该账户已在线的消息，如果没有则将此账户的ID和此客户端的SessionID存放入哈希表内，如果用户浏览器关闭或超过Session最大生命周期，则监听器会发现此Session失效，并返回一个信号给Action，Action将匹配此SessionID删除对应的账户信息，Session的最大生命周期要设为大于或等于考试最大时间。

2.3 客户端脚本的优化

客户端脚本的优化主要是针对用户键盘的误操作问题，比如键盘上有很多快捷键是与IE有关的，比如Backspace键、Power键、F5键、右键菜单、Shift+F10、浏览器的菜单栏、Alt+F4等这些操作都会改变当前页面的状态，必须屏蔽，最好的解决方法是使用JavaScript脚本来实现，以屏蔽Backspace、Shift+F10、F5刷新、退格键为例，其主要JavaScript代码如下：

```
if ((event.keyCode == 8) && (event.srcElement.type ! = "text" && event.srcElement.type ! = "textarea"&& event.srcElement.type ! = "password")|| //屏蔽 Backspace 删除键
(event.keyCode==116)|| //屏蔽 F5 刷新键
(event.ctrlKey && event.keyCode==82)) { //Ctrl + R
event.keyCode=0;
event.returnvalue=false;
}
if ((event.shiftKey)&&(event.keyCode==121)) //屏蔽 shift+F10
event.returnvalue=false;
```

2.4 数据库的优化

数据库的优化主要针对题型灵活性，一般的考试系统是将题目与答案分离，这样当题型改动较大时，可能需要重新设计表结构。这里笔者提出将题型、题目的内容、题目的选项、答案分离，见图3。

图3 题目分解图

Fig. 3 Decomposition chart of exercise problem

这样一来无论什么题型都可以划分为这几个元素，比如选择题可分解为题型(选择题)，题目内容(任意)、题目选项(选项数)、答案；填空题则可分解为题型(填空)、题目内容、题目选项(空格数)、答案。这样做虽然增加了数据的冗余度，但大大增强了题目的适应性。

2.5 优化结果

使用优化系统，若某账号已登录且尚未退出，再使用相同的账号登录时，则会出现图4所示界面，提示错误。

图4 登录优化结果

Fig. 4 Login optimization results

由于优化后题目选项数可以根据具体情况进行设置，题型设置更加灵活，图5为设置后界面。

图5　考试界面优化结果

Fig. 5　Exam interface optimization results

3　结语

本文提出一套基于Struts结构的在线考试系统的设计思想，并针对当前主流系统存在的问题，给出了解决方案。当今在线考试系统不仅限于将传统纸介转为网页形式，如何实现在线操作题，如何将非纸介题目如歌曲、图画、舞蹈转化为在线形式，如何对这些内容进行自动评分，还需要进行更深层次的研究。

参考文献(References)

[1] ECKEL B. Java编程思想[M]. 4版. 北京：机械工业出版社，2007.
[2] HALL M，BROWN L. Servlet与JSP核心编程[M]. 赵学良，译. 北京：清华大学出版社，2004.
[3] ZAKAS N C. Professional JavaScript for web developers[M]. Hoboken，NJ：Wiley Publishing，Inc，2005.
[4] 周四维. 基于Struts 2架构的云环境监测系统的研究[J]. 江汉大学学报：自然科学版，2011，39(4)：40-44.
[5] 孙卫琴. Java面向对象编程[M]. 北京：电子工业出版社，2006.
[6] ECKEL B. Thinking in Java[M]. 3rd ed. Upper Saddle River，NJ：Prentice-Hall，2002.
[7] 李刚. Struts 2.1权威指南[M]. 北京：电子工业出版社，2009.
[8] 廖雪峰. Spring2.0核心技术与最佳实践[M]. 北京：电子工业出版社，2007.
[9] 孙卫琴. 精通Hibernate：Java对象持久化技术详解[M]. 北京：电子工业出版社，2010.
[10] ELLIOTT J. Hibernate：A developer′s notebook[M]. 南京：东南大学出版社，2005.

（责任编辑：陈　旷）