

在线考试系统开发中的几个问题及解决方法^{*}

唐俊武 南理勇 左 强

(湖北工业大学信息工程学院 武汉 430068)

摘 要

简述了在线考试系统开发中网络拥塞、时间的控制、身份验证、试卷的生成、安全性、答案的存放等几个问题及其解决方法。

关键词:在线考试系统 身份验证 试卷生成

中图分类号:TP31

Several Problems and Solutions in Online Test System Software Development

Tang Junwu Nan Liyong Zuo Qiang

(Institute of Information Engineering, Hubei University of Technology, Wuhan 430068)

Abstract:The paper briefly introduces several problems and their solutions, such as network congestion, time control, authentication, generating test paper, security, saving answers and so on.

Key words:online test system, authentication, generate test paper

Class number:TP31

1 引言

Internet 技术的发展使得考试的技术手段发生了革命性的变革,它的开放性、分布性和巨大的计算能力打破了传统考试的时空界限。在线考试系统借助于遍布全球的因特网,具有灵活、客观、公正、安全、高效的特点。因此,在线考试将成为以后考试的发展趋势。现时,越来越多的企事业单位根据自身需要开发了相应的在线考试系统。本文将简要说明一个在线考试系统开发过程中的几个问题以及具体的解决方法。

2 问题及解决方法

整个在线考试系统由两个模块构成,面向管理员的系统管理模块和面向考生的考试系统模块。每个模块又由若干子模块组成。系统管理模块负责题库维护、考生管理、统计分析等。考试系统模块负责考生考试的全过程,具体包括登录界面设计和有效性验证、试卷的生成、考试界面的设计、答案

的存放等。考生考试系统模块是在线考试系统的核心部分,下面就该模块中一些问题及程序设计方案作具体说明。

2.1 考试过程中的网络拥塞

在线考试系统的设计中,首先遇到的是考试过程中网络拥塞问题。考试过程中,开始考试和交卷的瞬间,每一浏览器都向服务器发送 http 请求,造成服务器负担过重,引起长时间的等待,严重的会造成系统崩溃。整个网络负载呈极不均衡的马鞍状。

考试中的时延通常以毫秒计算,即使很短一段时间的负载分担也会带来网络拥塞的很大改善。因此,在等待考试开始时将试卷下载到本地是较好的解决方案,它实质是将网络负载分担在一段时间内。

下面给出了试卷预先下载的程序片断。

由于一段时间对同一试卷的下载次数很高,因此当用户请求下载试卷时,若试卷还未生成,调用 showpaper.asp,生成试卷 papertext¹—papertext⁶,放

入 Application(xxxxpapertexti), xxxx 为 PaperID, 以便重复下载。

```
if Application(PaperID&"str1")=null then
    str1=strGetHtmlHead()
    str1=str_change(str1)
    Application(PaperID&"str1")=str1
end if

下载试卷存放在用户浏览器中的页面 test_tools.asp 的一个变量中
```

```
<input type="hidden" value="<% = Application(session("strPaperID")&"str1")%>" name="papertext1">
```

2.2 考试时间的控制

考试过程中, 所有事务都是按时间顺序发生的, 所以对时间控制尤其重要。考虑到客户端时间的不精确, 决定以服务器时间为标准, 通过一个计时器来控制考试的进程。由于客户端机器性能不同, 计时器运行一段时间后会有一定的误差。为此, 编写一个计时器函数 clock, 先确定服务器时间与本地时间差, 通过不断读取本地时间的方法来校验当前时间。

```
function clock
    dim hours, minutes, seconds
    dim step_time: step_time=Time()
    if clocknow<>0 then
        clearTimeout(clocknow)
        clocknow=0
    end if
    if clock_on=true then
        step_time=DateAdd("s", difference, step_time)
        hours=Hour(step_time)
        minutes=Minute(step_time)
        seconds=Second(step_time)
    end if
    if hours<10 then
        hours="0"&hours
    end if
    if minutes<10 then
        minutes="0"&minutes
    end if
    if seconds<10 then
        seconds="0"&seconds
    end if
    document.tools.clock.value=hours&":"&minutes&":"&seconds
    clocknow=window.setTimeout("clock()", 1000)
else
    document.tools.clock.value=""
end if
end function
```

2.3 身份验证

根据考试用户的用户名和密码到数据库中查找相应记录, 如果库中存在该用户的记录, 则允许进入考试页面, 否则, 给出信息, 不允许进入考试页面。考虑到安全和速度我们使用了带参数的存储过程来实现身份的验证。

```
创建存储过程:
CREATE procedure usercheck
    @username varchar(50),
    @inpassword varchar(50),
    @outcheck char(3) OUTPUT
as
    if exists(select * from userinfo where username=@username and password=@inpassword)
        select @outcheck='yes'
    else
        select @outcheck='no'
ASP 程序调用存储过程:
<%
if request("ok")=1 then '表单提交标志
'建立数据库连接
Set Comm=Server.CreateObject("ADODB.Command")
Comm.ActiveConnection="DSN=localserver: UID=sa: PWD=: Database=Itembank"
'以 comm 对象建立存储过程连接, 4 代表连接类型为存储过程
Comm.CommandText="usercheck"
Comm.CommandType=4
'将第一个参数 username 追加到 p1 集合中
Set p1=Comm.CreateParameter("username", 200, 1, 50, request("username"))
Comm.Parameters.Append p1
Set p1=Comm.CreateParameter("password", 200, 1, 20, request("password"))
Comm.Parameters.Append p1
Set p1=Comm.CreateParameter("check", 129, 2, 3)
Comm.Parameters.Append p1
Comm.Execute '运行存储过程
if comm("check")="yes" then
    response.write "欢迎进入系统! 用户名:"&comm("username")
else
    response.write "对不起, 您尚未注册!"
end if
Set Comm=nothing '释放连接
else
'表单部份
%>
<html>
```

```
<head>
<title>身份验证</title>
</head>
<body>
<form method="POST" action="check.asp">
  <p><input type="text" name="username" ><br
>
  <input type="password" name="password" ></p>
>
  <p><input type="submit" value="提交" name="s"></p>
  <input type="hidden" name="ok" value="1">
</form>
</body>
</html>
<%
end if
%>
```

2.4 试卷的生成

试卷页面是动态生成的。数据库中有一个字段,标记试卷代号,可以根据用户输入的试卷代号从数据库中选取相应记录,并把它按照一定布局显示在页面上。选择题对应一个选择题表,可以从选择题表中取出相应记录,同时动态生成表单域。然后从填空题表中选取符合条件的记录,从简答题表中选择相应记录,这样就可以生成一个试卷。下面以试卷中生成判断题的代码为例说明试卷页面的生成。

```
total _ select 为从题库中抽取出来的试题数量
<% set rs=server.CreateObject("ADODB.RecordSet")
rs.open "select * from 判断题 order by id asc", conn,
1,1
mycount=rs.Recordcount '取得题库中的试题总数
rs.close
for i=1 to total _ select
randomize
sid=int((mycount +1) * rnd+1) '生成 1~题库中的
的试题总数之间的随机数
set rs=conn.execute("select * from 判断题 where id
="&sid)
while rs.eof
randomize
sid=int((mycount +1) * rnd+1)
'如果数据库找不到这条试题,就继续生成随机
数读取试题。
set rs=conn.execute("select * from 判断题 where id
="&sid)
wend
```

```
%>
'输出试题
<tr>
<td width="10%" ><%=i%>、<%=rs("题
目内容")%></td>
<td align="center" width="10%" ><select
name="cate"<%=i%>">
<option selected value=True>对</option>
<option value=False>错</option></select>
</td>
</tr>
<%next
rs.close%>
```

这种方法实现了随机抽取试题,但是在题库中试题数量不多的情况下,容易选取重复的试题。因此,可以将题库中试题总数平均分为很多个小的范围,让每个小的范围产生一个随机数,这样就有效地避免了随机数重复。当然,这种方法也存在一个问题,就是题库中的每一道试题出现的概率不相同。对于随机抽取试题的算法还需要改进。

2.5 安全性考虑

安全性包括许多方面,在这里只讨论加密方面。对于一些重要敏感信息我们必须考虑其安全性。比如用户的口令、重要的数据等可以对其进行加密。一些关键主要模块的程序代码也必须进行加密,有的甚至要做成动态链接库。加密解密的方法有很多种,常见的有字节异或加密、相邻字符换位、md5、sha1 等。

在实际的系统中,对用户的口令信息采取了字节异或加密的方法,因为该方法效率高,加密函数和解密函数相同。加密函数 encrypt(text, key) 如下:

```
function encrypt(text, key)
dim itextChar, ikey, ii, t, icryptChar
for i=1 to len(text)
if i<len(key) then
t=i mod len(key)
else
t=i mod len(key)+1
end if
ikeyChar=Asc(mid(key, t, 1))
itextChar=Asc(mid(text, i, 1))
icryptChar=ikeyChar xor itextChar
strEncrypted=strEncrypted&Chr(icryptChar)
next
encrypt=strEncrypted
end function
```

2.6 答案的存放

先从试卷页面中收集选择题、填空题及问答题答案,并把它们分别保存在三个字符串中,然后利用 SQL 语句向数据库中存放,如果数据库已经有用户的答案记录,则用 UPDATE 语句更新,否则用 INSERT 语句插入。

3 小结

本文是对在线考试系统开发工作的一个总结,限于篇幅,仅列出了其中的几个问题。在线考试系

统开发中有许多问题,这有待于我们进一步观察、发现并提出更好的解决方法,使得系统更加完善。

参 考 文 献

[1]Johnson S. Active Server Pages 详解[M]. 北京:电子工业出版社,1999
[2]刘念伯.电子科技大学毕业论文
[3] 张江等.通用考试系统的设计与实现[J].电子科技大学学报,2001,30(2):157~161.
[4]蒋理.网络考试系统设计[J].微机发展.2000,(6):71~72

(上接第 54 页)

优值,有时甚至由决策者根据精度要求给出^[5]。而满足约简条件的 β 值可能为一个区间范围,可变精度粗糙集进一步研究了 β 值变化对分类率 γ 影响的区间性。

在不同分类率 γ 下,信息系统的 β 取值区间是不同的,因而其表示的信息存在着差异,不存在可比性。因而 β 约简必须是分段的,根据不同的分类率 γ ,进行各自的约简。

在相同分类率 γ 下,约简后 β 取值区间可能产生差异,即与约简前的属性集合比较,对于相同分类率 γ ,当前约简属性集的 β 取值范围可能不同。

实际上, β 约简问题的关键在于约简集合能否表达与原属性集完全一致的信息。在 β 约简过程中,由于 β 值区间在约简过程中可能发生变化,要保持分类率不变,正区域中元素必须具有前后一致性。

定义:给定信息系统 $S=(U, C \cup D)$, U 为论域, C 为条件属性集合, D 为决策属性集合,条件属性 C 关于决策属性 D 的 β 约简定义为 C 的一个最小属性子集 $RED(C, D, \beta_{RED}^\gamma)$, 且满足

- (1) $POS(C, D, \beta^\gamma) = POS(RED(C, D, \beta_{RED}^\gamma), D, \beta_{RED}^\gamma)$
- (2)从 $RED(C, D, \beta_{RED}^\gamma)$ 中去掉任何一个属性, (1)不成立
- (3) $(\beta_{RED}^\gamma = \bigcap_p \beta_p^\gamma) \neq \Phi$

定义描述了在可变精度粗糙集模型下,给定分类率的 β 区间约简。 $\bigcap_p \beta_p^\gamma$ 表示了属性约简中每一步 β 区间的交集合, β 约简的区间性描述了在一个

区间范围内,任意取定一个 β 值,都对应着相同分类率的约简。从而把对给定 β 的约简,扩展到 β 区间的约简。

6 结束语

可变精度粗糙集分析的数据模型表示了统计趋势而非函数依赖。大部分决策信息系统属性之间并不一定存在严格的函数依赖关系,而只是表现出近似依赖的关系。

近似依赖性是粗糙依赖性的推广,当 $\beta=1$ 时,它就变成粗糙依赖性。粗糙依赖性度量了完全准确的分类能力,而近似依赖性度量的是在包含度 β 下的分类能力,因此,它并不能解释为属性间的严格函数依赖或部分依赖,而只是表示数据之间较弱的近似依赖关系。

参 考 文 献

[1]Ziarko W. Variable precision rough set model[J]. Journal of Computer and System Science, 1993, 46:39~59
[2]A. AN, N. Shan, C. Chan, N. Cercone, W. Ziarko. Discovering rules for water demand prediction: an enhanced rough-set approach[J]. Engineering Application and Artificial Intelligence, 1996, 9(6): 645~653
[3]Ziarko W. Analysis of uncertain information in the framework of variable precision rough sets[J]. Foundations of Computing and Decision Sciences, 1993, 18:381~396
[4]Ziarko W. Decision making with probabilistic decision tables, in: N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular Soft Computing [C]. Proceedings of the Seventh International Workshop, RSFDGrC '99, Yamaguchi, Japan, 1999, 463~471
[5]Beynon M. Reducts within the variable precision rough set model: a further investigation[J]. European Journal of Operational research 2001, 134:592~605