DOI:10.13792/j.cnki.cn45-1300/z.2007.03.037

用动态代理实现网上考试系统的访问控制机制*

□雷 芸[1],刘 恒[2]

(1.广西民族大学 预科教育学院 助教,广西 南宁 530006)
(2.玉林师范学院 数学与计算机科学系 助教,广西 玉林 537000)

【摘 要】结合玉林师范学院远程教育项目的研究工作,分析了网上考试系统的架构与安全性需求,研究了该系统的访问控制,并以 JAVA 的动态代理为设计模式,给出了一套关于认证和授权的实现方法.

【关键词】网上考试;动态代理;认证;授权
【中图分类号】TP37【文献标识码】A【文章编号】1004- 4671(2007)03- 0128- 04

计算机应用与研究

Realize the Access Control of the Internet- Exam- System with Dynamic Proxy

LEI Yun[1],LIU Heng[2]

(1.Assistant, Detpartment of Preparatory Education,
Guangxi University for Nationalities, Nanning, Guangxi 530006)
(2.Assistant, Department of Mathematics and Computer Science,
Yulin Normal University, Yulin, Guangxi 537000)

Abstract:Based on the research work of distance education project of YuLin Normal University, the framework of the Internet- Exam- System and the secure requrements are analyzed,the access control of the system is researched,and with the design pattern of dynamic proxy in JAVA, a method about the realizing of authentication and authorization is presented.

Key words:Internet- Exam; dynamic proxy; authentication; authorization

网上考试系统是远程教育支持系统的一个重要子系统,是以组织客观、公正、科学合理的考试为目的的标准化考试系统.该子系统由在线考试、考试管理、题库录入三个部分组成,可实现题库管理、组卷、报名及审核、考试控制、考试监督、在线考试、评分阅卷、信息查询等功能,学员通过网上报名参加网上测验考试,评分阅卷等由系统自动完成.该子系统的建设,对于远程教育支持系统实现学生学习管理与考核的功能,将发挥重要的技术支撑作用.

1 网上考试系统的架构与安全性分析

系统采用 B/S 与 C/S 相结合的架构模式,考试报名、考试实现、及考试成绩查询可由各考试机浏览器完成;考生考试过程中下载试题到本地机,答

* 2006 年广西研究生教育创新项目.

案暂存于本地机;报名信息审核、题库管理、组卷、考试控制、系统评分等核心业务逻辑由 WEB 服务器(考试中心服务器)完成.

由于考试系统的特殊性,安全性显得格外重要.网上考试系统从考生登陆(注册)到参加考试,到查询考试成绩全面采用 256 位的数据加密技术.另外在考生考试模块的设计中,采用大量的安全技术,例如:禁止刷新,禁止查看源代码,考试结束自动交卷,不能用同一用户名多次同时登陆等等.

网上考试系统中区分管理员、教师和考试者这三种角色,管理员负责通过学生注册、分配教师帐号并维护题库,教师权限负责设定有关考试环境的各项参数并可以更新题库;而考生注册后通过才可参加考试,并且对试题只有读的权限,但在答题区有写的权限,对暂存于本地机的答案则不可访问.网上考试系统必须正确认证各用户的身份,在严格区分各用户的权限的基础上对用户进行正确的授权管理.运用动态代理来实现上述目标是一个可行的选择.

2 代理与动态代理

代理(Proxy)是 JAVA 的一种设计模式,它的定义是:为其他对象提供一种代理以控制对这个对象的访问.即在出发点到目的地之间有一道中间层,意为代理.使用 Proxy 的目的主要有两个:1.认证授权机制 不同级别的用户对同一对象拥有不同的访问权利,如在网上考试系统中,有管理员、教师和考试者这三类人,就可以使用 Proxy 进行授权机制控制,控制这三种用户对论坛的访问权限.2.某个客户端不能直接操作到某个对象,但又必须和那个对象有所互动,如在网上考试系统中,考生不能直接访问题库、卷库,而是通过访问考试过程中所使用的 WEB 数据库服务器来与题库互动.

通过代理模式对类进行权限控制访问.这是代理模式的一个主要用处,要为每个类实现一个 Proxy 类,这样带来了很多 Proxy 类,非常琐碎,从 Java 1.3 开始,Sun 提供了 Dynamic Proxy API,Jdk1.3 开始支持动态代理类,这种类可以实现在运行时指定的一组接口.对代理类的方法调用,被分配给调用处理程序,而调用处理程序是在代理类生成的时候指定的.动态代理为 Java 开发人员提供了许多有意义的解决方案,采用动态代理模式,可以方便而有效地实现网上考试系统的访问控制机制.

3 网上考试系统的认证机制实现

认证是验证网上考试系统用户身份的过程,授权是根据请求用户的身份允许访问或操作某些对象的过程.这两个概念密不可分.没有授权,就无需知道用户的身份.没能认证,就不可能区分可信和不可信用户,更不可能安全地授权访问许多系统部分.利用网上考试系统进行测验考试过程中任何一个关键环节均需要对考生进行身份验证.使用动态代理方法,所有的验证逻辑对于代理的客户,都是透明的.因此,验证方案应当会非常简单,而且验证逻辑与应用程序的核心业务逻辑中完全没有耦合,还带来了可以重用、可以配置的验证代码.

3.1 调用处理程序类是处理所有数据验证逻辑的地方.下面的程序片段显示了一个调用处理程序,它没有绑定到任何具体的对象,这样就能把它用于网上考试系统中任何需要被验证的对象.

```
……
private Object delegate = null;
public BusinessObjectInvocationHandler (Object delegate) {
    this.delegate = delegate;
}
public Object invoke (Object proxy, Method method, Object[] args)
throws Throwable {
        BusinessObjectValidationService.validate (proxy, method.getName(), args);
    Object retVal = null;
    try {
        retVal = method.invoke(delegate, args);
    } catch (InvocationTargetException ite) {
            throw ite.getTargetException();
    }
    return retVal;
}
```

3.2 这一步把业务对象实现指定给调用处理程序的构造函数.由于验证逻辑是在调用处理程序中处理的,因此下面的程序片段采用不受验证限制的方式实现 User 接口.

```
……
private String username = null;
private String password = null;
public String getUsername() {
```

```
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
```

3、下面的程序片段显示了如何为 User 接口建立动态代理(用 exam 实现类),同时通过调用处理程序传递所有的方法调用.

```
public static User create() {
    return (User)Proxy.newProxyInstance (User.class.getClassLoader(),
                    new Class [] {User.class}, new BusinessObjectInvocationHandler(new Exam()));
}
```

动态代理类给了开发者一种以统一方式方便地处理任何方法上的验证途径,因而适用于网上考试系统中对考生严格的身份验证.

4 网上考试系统的授权机制实现

当一个用户经过上一节所描述的认证过程并 login 后,系统就要在内存中为其建立相应的授权访问机制,使用动态 Proxy 与 java.security.acl 的 ACL 相结合的机制,可以灵活地实现粗粒度和细粒度的双重授权控,从而建立一个安全授权系统.授权步骤如下:

4.1 Java acl 开始第一步是建立一个主体 Principal,设网上考试系统某用户 Student 是主体的拥有者:

```
private static final Principal _Student = new PrincipalImpl("Student");
```

4.2 网上考试系统程序从数据库中取出其权限数据(即取出用户和被访问对象之间的权限关系,这种权限关系可能不只一个,将其打包在 Hasbtable 中)建立 Permission,这里使用 Feature 继承了 Permission,在 Feature 中定义了有关权限的细节数据(如读或写).

```
Hashtable features = loadFeaturesForUser(sApplicationName, sUserID);
```

4.3 创建一个用户对象

```
User user = new UserImpl (sUserID, new Hashtable() );
```

4.4 为这个用户创建一个活动的 acl entry,并遍历 Hashtable features,将其中多种权限加入:

```
AclEntry newAclEntry = new AclEntryImpl( user);
……
feature = (Feature) hFeatures.get(keyName);
newAclEntry.addPermission( feature );
....
```

最后加入主体拥有者 Student,一个 ACL 安全体系就已经建立完成.当系统要检验某个用户是否拥有某个权限,如读的权限时,只要使用 acl.checkPermission(user, feature)即可

有了 ACL 机制后,就可以在网上考试系统中使用动态 Proxy 模式来对具体对象或方法进行控制,比如,对某次考试的考卷中的题目,考生可以读,教师和管理员可以进行改动(这些权限已经在上面 ACL 里部署完成),使用动态 proxy+acl 就可以实现很好的细粒度控制.

为了使动态 Proxy 能够工作,首先必须有一个 Proxy 接口,还要有一个继承 InvocationHandler 的 Proxy 类.

```
public class ExamProxy implements InvocationHandler
{
  private Exam exam;
public static Object newInstance (Exam exam,Class[] interfaces)
{
return Proxy.newProxyInstance (exam.getClass().getClassLoader(), interfaces,new ExamProxy(exam));
}
  public ExamProxy(Exam exam)
{
this.exam = exam;
}
public Object invoke (Object proxy, Method m, Object[] args) throws Throwable
{
Object result;
String methodName = m.getName();
```

```
if (methodName.startsWith("get"))
{
if (!acl.checkPermission(user, "read")) return null;
String name = methodName.substring(
methodName.indexOf("get")+3);
return exam.get(name);
}
else if (methodName.startsWith("set"))
{
if (!acl.checkPermission(user, "write")) return null;
String name = methodName.substring(
methodName.indexOf("set")+3);
exam.put(name, args[0]);
return null;
}
else if (methodName.startsWith("is"))
{
if (!acl.checkPermission(user, "read")) return null;
String name = methodName.substring(
methodName.indexOf("is")+2);
return(exam.get(name));
}
return null;
}
}
```

上例只是对信息的读、写进行简单的权限控制,除此之外,信息或程序的执行、拷贝、打印、查询、修改、新增、删除方面的授权都可以用动态代理方便地实现,例如考生只能在某时间以前运行答题的模块功能(考试结束不能再答题),某教师只能修改题库中的填空题或选择题,某管理员可以修改发布某些考试信息,等等.

5 总结

动态代理比代理模式更加简洁和抽象,使用动态代理将使得调用者无需指定被调用者的代理类,这是动态代理区别代理模式的本质.动态代理这一优势,又可以体现在另外一句话语上:动态代理拦截了调用者对被调用者的调用.使用动态代理实现网上考试系统中认证、授权等访问控制功能,带来了更大的灵活性和安全性. ■

【参考文献】

[1] Harvey M.deitel,Paul J.Deitel,Sean E.Santry 著.钱方,梅皓,周璐,吴志英等译.高级 Java2 大学教程[M].北京.电子工业出版社.2003.

[2] 黄嘉辉. Java 网络程序设计[M].北京:清华大学出版社,2002.

[3] 冯登国.网络安全原理与技术[M].北京:科学出版社,2003.

【收稿日期　2007-03-18】
【责任编辑　谢明俊】