第三方物流系统中权限管理的研究与实现①

王　艳　毕　利　刘树军　(宁夏大学 数学计算机学院 宁夏 银川 750021)

摘　要： 首先结合第三方物流系统的特点，分析了传统访问控制方法的不足与目前比较流行的-基于角色的访问控制(RBAC　Role Based Access Control)方法的优势，然后总结了 Struts 和 EJB 的特点，最后阐述了一个基于角色的以 Struts+EJB 为架构的第三方物流系统中权限管理的设计和实现过程。实验证明：该权限管理安全性高、灵活性好不仅为第三方物流系统提供了良好的权限控制，而且可以被重复利用加入到其它具有权限管理的系统中。

关键词： 第三方物流系统;权限;RBAC; EJB;Struts

Research and Implementation of Rights Management in Third Party Logistics System

WANG Yan, BI Li, LIU Shu-Jun

(Mathematics and Computer College, Ningxia University, Ningxia,Yinchuan750021, China)

Abstract: The paper first combines the characteristics of a third-party logistics system, analyzes the shortcomings of traditional access control methods, and discusses the advantages of role-based access control(RBAC Role Based Access Control)methods, which is currently popular. The paper then summarizes the characteristics of EJB and Struts, and finally introduces the design and implementation process of right management in the third-party logistics with a role-based and Struts+EJB architecture. The result shows that the system has high security, flexibility, and not only provides a good access control for the third-party logistics system, but it also can be reused by adding to the other system with the right management.

Keywords: third-party logistics system; rights; RBAC; EJB; struts

近年来，越来越多的生产厂家为了减少成本，都将仓储和配送运输部分外包给了第三方物流企业，于是第三方物流企业在我国迅速的发展起来。第三方物流管理系统是第三方物流企业进行物流业务信息化管理的平台，平台建立得好坏将直接影响企业能否降低营运成本、减少流动资金的占用和能否提高利润，从而决定企业是否具有竞争力[1]。由于第三方物流信息系统越来越向着多用户的方向不断发展(仓库管理员、结算管理员、货主、经销商等)，于是对系统安全性和灵活性方面也就提出了更高的要求。因此一个好的第三方物流系统必须建立一个良好的权限控制系统，以保证系统的安全性。

1　RBAC与传统访问控制方法的比较

传统的访问控制方法主要包括自主访问控制(DAC)和强制访问控制(MAC)。DAC 的基本思想是：用户可以按自己的意愿决定哪些用户可以访问他们的文件，并可以将自己所拥有的权限不受任何限制地授予其他用户或收回授予的权限[2]。DAC 的优点是它到访问控制是自主的，这种自主性为用户提供了很大的灵活性。缺点是安全性不高。MAC 是相对于 DAC 而言的，只是比后者要求更严。MAC 是“强加”给访问主体的，即系统强制主体服从访问控制政策，系统根据主体被信任的程度和客体所包含的信息的敏感程度来决定主体对客体的访问权，这种控制往往可以通过

① 基金项目:宁夏科技攻关项目(08-02-008); 宁夏自然科学基金(NZ0920)
收稿时间:2010-01-19;收到修改稿时间:2010-03-20

给主、客体赋以安全标记来实现。MAC 的优点是，存取控制比较严格；缺点是配置的粒度大，灵活性差[3]。

本系统采用的是目前比较流行的基于角色的访问控制方法(RBAC)，RBAC 在用户和权限之间引入了角色这个中介，通过给用户分配适当的角色以及对角色分配适当的权限，来实现对用户授予权限。在一个多管理员的系统中，当用户职能发生变化时，只要将该用户从一个角色移到另一个角色,而不需要对用户重新赋予权限，降低了权限管理和系统维护的复杂性。

2 系统采用整合Struts+EJB3.0的思路

本系统选择了 Srtuts 框架和 EJB3.0 组建技术。这是因为：Struts 是一个开放源码的、基于 MVC 模式的应用框架，它具有丰富的标签库，能大大提高开发效率，并且对其他技术和框架都有很好的融合与支持。EJB 是一个可重用、可移植的 J2EE 标准的核心组件，Sun 公司对 EJB 的定义是：EJB 的结构是开发和配置基于组件的分布式商务应用程序的一种组件结构，用 EJB 结构开发的应用程序是可伸缩的、事务性的、多用户安全的，这些应用程序可能只需编写一次，就可以在支持 EJB 规范的服务器平台上进行配置[4]。

但是由于 EJB2.0 以前的版本实现起来比较复杂，在一定程度上会影响了系统的开发效率，EJB3.0 对 EJB2.0 做了很多的简化，其中最大的改变就是处理持久化的方式。EJB3.0 用 JPA 替代了 EJB2.0 的实体 Bean,直接持久化保存到数据库，而 EJB2.0 中还需要在模型类和实体 Bean 中进行一次转换[5]。

Struts 与 EJB3.0 结合能充分发挥 Struts 清晰的 MVC 结构，以及 EJB3.0 基于组件分布式计算结构的优势，从而大大提高系统的开发效率、数据的安全性和可维护性[6]。

在 Struts+EJB3.0 架构中， EJB3.0 用来实现 Struts 中 MVC 的模型层，让 Session Bean 来封装业务逻辑，充当框架的业务逻辑层。让 JPA 来完成数据持久化，充当框架的持久化层。Struts 用来实现控制器和显示逻辑，充当框架的表示层。形成表示层、业务逻辑层，持久化层三层结构，在很大程度上保证了系统清晰的层次结构及良好的性能。其体系结构如下图 1 所示：

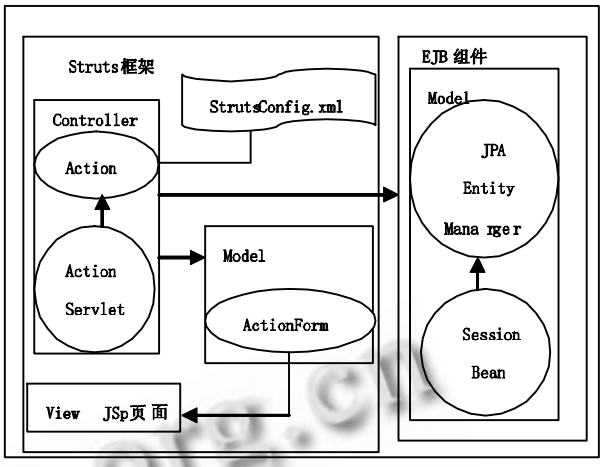

图 1 Struts+EJB3.0 体系结构图

3 系统的设计与实现

3.1 角色的分类

角色是由超级管理员按管理员职能来划分的。根据第三方物流企业的性质和管理结构的特点，将此系统的角色分为三大类：超级管理员、仓库管理员和结算管理员。

超级管理员：主要是对普通管理员进行角色及操作点的分配，以及对角色进行权限的分配等管理。

仓库管理员：主要是经超级管理员授权后对所在操作点的特定模块进行管理和操作，比如：入库，出库，配送等。

结算管理员：主要是经超级管理员授权后对所在操作点所产生的费用进行结算的权限。

3.2 权限的分类

本系统实现的权限可分为：操作点权限和模块权限。

操作点权限：由于本系统不是针对一个地方来使用的，它的使用范围是宁夏一些县市，因此本系统将这些县市分成了不同的操作点，如：银川操作点，石嘴山操作点，固原操作点等，超级管理员要对仓库管理员和结算管理员分配不同的操作点，不同操作点的管理员只能拥有相应操作点的管理权限，未经授权不能跨操作点操作。

模块权限：本系统主要包括许多模块，每个模块中又包含许多的功能(如添加、删除、修改等)，超级管理员首先对角色分配权限，即分配模块及功能，再对不同的用户分配不同的角色，使不同的用户只能拥有对特定模块中特定功能操作的权限，未经授权的用户

不能对其它模块进行管理和操作。

具体的实现流程是这样的：首先超级管理员添加系统中用到的所有的功能，即将所有 action 中的方法通过页面添加到 function 实体中，然后添加系统中的模块地址到 module 实体中，module 地址为以(*.do)结尾的 action 的名称，功能及模块添加完以后，再将不同的功能分配给模块，这样一个模块就对应上了一个或多个功能，形成了权限。接着新建用户及角色，为用户分配不同的角色，然后为角色分配所在的操作点及相应的权限，这样就使得处于不同操作点的角色具有不同的权限，比如：超级管理员对 A 仓库管理员分配了银川操作点和配送运输模块中修改配送订单的权限之后，那么这个 A 仓库管理员就具有了修改银川操作点配送运输模块中修改配送订单的权限，未经再次授权不具有其它任何权限，从而保证了第三方物流整个系统的安全性。

其实现过程如下图 2：

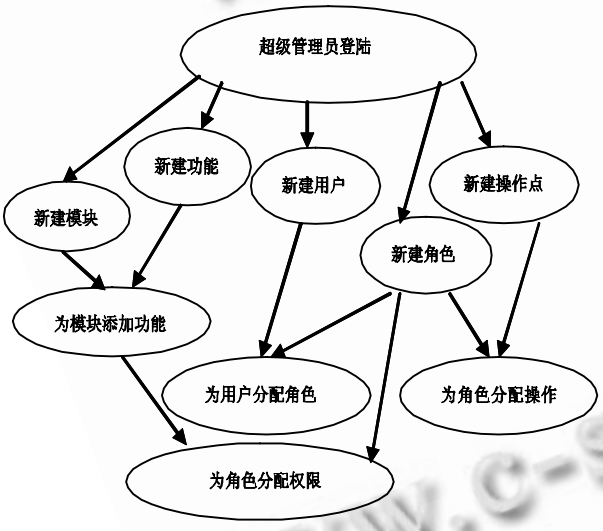

图 2　超级管理员实现权限流程图

3.3 数据库设计

实体和系统中的数据库表是一一对应的。用户表中主要存放了用户的用户名和密码，角色表中主要存放了角色的名称和描述，用户表和角色表是多对多的关系，即：一个角色可以包含多个用户，一个用户也可以属于多个角色，用户表和角色表通过用户-角色这个中间表关联。用户-角色表中存放了用户表的主键和角色表的主键。操作点表主要存放了操作点的编号和名称，操作点表和角色表是多对多的关系，它们通过角色-操作点表关联。角色-操作点表存放了角色表的主键和操作点表的主键。模块表主要存放了模块的名称和地址(以*do 结尾的 action 的名称)，功能表主要存放了功能的名称(action 中的方法名)和描述，模块表与功能表之间是多对多的关系，即一个模块可以包含多个功能，一种功能可以属于多个模块(比如：配送管理模块可以包含添加，删除，修改，查询等多个功能，而添加功能也存在于基础数据管理、车辆管理、费用管理等其它多个模块)，模块表和功能表通过权限表关联，权限表中存放了模块的主键和功能的主键。权限表和角色表通过权限-角色表关联，权限-角色表中主要存放了权限表的主键和角色表的主键，权限表和角色表之间也是多对多的关系。

实体间的关系如图 3 所示：

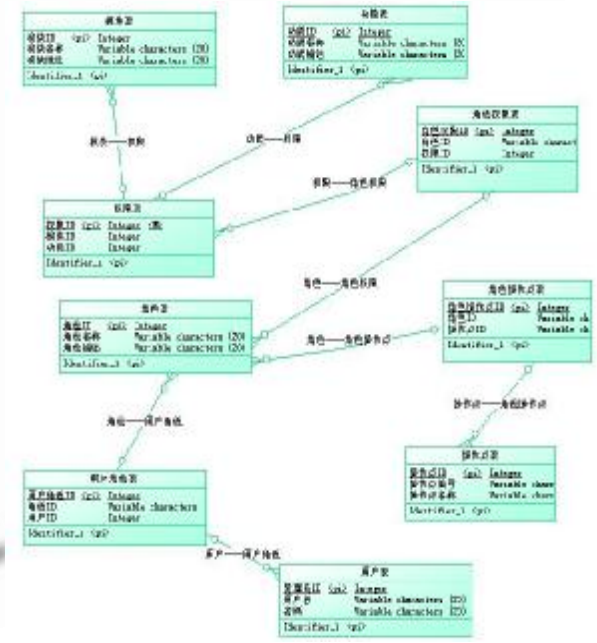

图 3　实体间的关系

3.4 系统实现

本系统中客户端采用 Struts 架构，服务器端采用 EJB3.0 组建技术。

3.4.1 服务器端

(1) Entity

本系统通过 JPA 把实体自动持久化到关系数据库中，实体中的属性与数据库中表的字段是一一对应，实体与数据库表之间以及实体与实体之间的的映射关系都是通过注解来实现的。下面以角色实体即

role.java 中的部分代码为例说明如下：

```
@Entity        // Role 所对应的类是实体类型
//实体 Role 所对应对的数据库的表名为 “role”
@Table(name="role")
public class Role {
//主键值的生成方式（用持久化驱动生成数据库表必须使用该注解）
@GeneratedValue(strategy=GenerationType.AUTO)
//该属性映射到数据库表中的 roleId
@Column(name="roleId")
@Id     //此属性为该实体的主键
public Integer getRoleId()
//指定了 role 实体和 staffrole 实体是一对多的关系
@OneToMany(mappedBy="role",targetEntity=Staffrole.class,cascade={CascadeType.REFRESH},fetch=FetchType.LAZY)
public Set getUserrole()
public void setStaffrole(Set Userrole)
```

(2) Session Bean

EntityManager(持久化管理器)作为 JPA 中非常重要的一部分，它是 EJB3.0 最新引入的。提供了对实体与数据库之间的映射管理，主要包括添加、更新、删除、查询实体等，下面首先介绍创建 EntityManager 接口，以 RoleDAO 为例：

```
public void save(Role role)     //添加角色
public List getUser(int roleno)//为用户分配角色
```

下面介绍创建 EntityManager 实例，以 Role DAOBean 为例：

```
public void save(Role role) //添加角色
{           em.persist(role); }
public List getUser(int roleno) {
                    //为用户分配角色
List l = new ArrayList();
Role s = findById(roleno);
Set you = s.getUser();
Iterator iter = you.iterator();
while (iter.hasNext()) {
Userrole px = (Userrole)iter.next();
User r = px.getUser ();
l.add(r); } return l; }
```

3.4.2 客户端

当客户端发出请求后，请求首先被控制器 Action Servlet 截获，ActionServlet 在 Struts-config.xml 配置文件中查找相应的映射，然后将相应的 ActionMapping 和 ActionForm 转发给 Action，在 Action 中通过 JNDI 查找到 EJB 的接口(Remote 接口或 Local 接口)完成对 EJB 方法的调用，最后通过 ActionForward 将 Action 类的处理结果发送至相应的 JSP 页面。下面以 RoleAction.java 中对部分代码举例说明。

```
//调用 EJB 远程接口，其中 RoleDAOBean 是这个 SessionBean 的 JNDI 名称
RoleDAOdao=(RoleDAO)ctx.lookup("RoleDAOBean/remote")
//从 Form 表单中得到用户输入的值并将其存入实体中
Role.setRolename(RoleForm.getRolename().trim());
//调用 sessionBean 中的 save 方法，实现添加角色
dao.save(role);
//通过 Struts-config.xml 查找到 addRole 所对应的 jsp 页面并转到此页面
return mapping.findForward("addRole");
```

以上只是简单举例说明了采用 EJB3.0+Struts 实现权限管理的流程，可以看出结合 Struts 与 EJB3.0 能充分发挥 Struts 清晰的业务逻辑与表示逻辑分离的特点，以及 EJB 基于组件分布式计算结构的优势，而在用户和权限之间引入角色这个中介，使得权限管理更加清晰，维护更为方便。

4 总结

本文通过对 DAC、MAC、RBAC 及 EJB、Struts 的研究与分析，阐述了基于角色的以 Struts+EJB 为框架的第三方物流系统中权限管理的设计与实现过程。RBAC 实现了用户和访问权限的逻辑分离，这样使得系统管理员对系统管理具有非常好的灵活性和易操作性；在 Web 客户端采用 Struts 框架，在服务器端采用 EJB 组件技术，不仅实现了分布式开发，有效地提

高了系统开发效率，还使得系统具有很高的重用性、扩展性和安全性。

参考文献

1 厉小军,朱鸿斌,胡上序.基于 B/S 结构的第三方物流系统设计与实现.计算机工程, 2003,29(1):256－258.

2 陈丹丹.基于 RBAC 的权限管理组件的设计与实现[硕士学位论文].武汉:武汉理工大学, 2008.

3 蔡世伟.基于 Struts+Hibernate 的权限管理系统的设计与实现[硕士学位论文].北京:北京邮电大学,2008.

4 朱俊成,李有军,王俊伟.EJB3.0 从入门到精通.北京:电子工业出版社,2009.24－26.

5 王博,陈莉君.EJB3.0 中实体 Bean 映射的深入研究.西安邮电学院学报, 2008,13(5):116－119.

6 商慧波.基于 Struts 和 EJB 的 B/S 开发框架研究与应用[硕士学位论文].大连:大连海事大学, 2005.