B/S结构软件系统权限管理的精粒度实现*

倪泰山[1]，周永刚[2]

（1. 云南省红河州地震局，云南 蒙自 661100；2. 北京市延庆县地震局，北京 102100）

摘要：介绍了 B/S结构软件设计中的精粒度权限管理的思路及方法，运用 Java高级程序设计语言，采用 Struts2＋Hibernate＋Spring框架，结合 DWR技术实现了软件系统中的权限管理，并将该技术应用到云南省的台站管理系统的后台管理中。

关键词：B/S结构；数据库；权限管理；访问控制列表

中图分类号：TP302.1 文献标识码：A 文章编号：1000－0666（2010）03－0349－05

0 引言

在基于 B/S结构或 C/S结构的软件系统中，权限管理是一个基础的而且非常重要的模块。权限管理设计的好坏，直接影响到软件系统的安全性、可维护性和可扩展性。权限管理有多种实现技术，最简单的是基于用户的权限管理（倪泰山等，2009），另一种是基于角色、组、用户的权限认证技术，这种技术将具有相同权限的用户划归同一角色或同一个组，然后将权限赋予角色，实现精细的权限控制。

在“地震台站综合管理系统”2.0版的设计中，设计人员采用开源框架 Struts2＋Hibernate＋Spring技术进行代码重构，使用 DWR框架实现后台与前台交互，权限管理采用基于“角色＋用户”的认证技术，从而实现了对用户权限的精细控制。

1 应用系统的权限管理设计

1.1 权限管理的粒度

粒度问题是数据仓库设计的一个最重要方面。粒度是指数据仓库的数据单位中保存数据的细化或综合程度的级别。

（1）用户（或角色）粒度。它可以精细到角色或具体的用户，由系统管理员将某一资源的访问权限授权给角色或用户，从而实现对数据库的 CRUD（Create, Read, Update, Delete）操作。这里的角色可以理解为实际工作中的岗位、职位或者分工。角色和用户组看起来很相似，但却有着本质的区别，最主要的区别在于：用户组是一个用户的集合，并不涉及它的授权许可；而角色则既是一个用户的集合，又是一个授权许可的集合。

（2）操作对象粒度。对数据库的查询、增加、更新、删除等操作，是由具体的模块实现的（刘敦敏，2004）。对于新注册的用户，一般可以自动赋予查询权限，对没有特别限制的模块中实现的功能，都可以进行数据浏览。增加、更新、删除等功能则由系统管理员在授权页面对角色或某一具体用户进行模块的 CRUD授权，用一个正整数的低 4位的取值表示 CRUD权限，位的取值为“1”表示允许，取值为“0”表示不允许。0001为十进制的 1，拥有 C权限；0010为十进制的 2，拥有 R权限；0100为十进制的 4，拥有 U权限；1000为十进制的 8，拥有 D权限。它们之间进行“与”运算则构成组合权限。例如，某角色对某一资源的权限为 1101（十进制 13），表示该角色对该资源具有 CRD权限，但没有 U（更新）权限。

1.2 权限管理数据库表结构设计

权限管理作为 B/S结构软件系统的基础，采用 5张数据库表来维持。模块数据表（T_Modules）记录每个模块的名称（前台主菜单名称）及子模块名称（前台子菜单名称）、入口地址等信

* 收稿日期：2009－10－31.

息，它是前台显示菜单的主要依据。用户信息记录在用户数据表（T_Users）中，角色信息记录在角色数据表（T_Roles）中。用户与角色是“多对多”关系，靠用户角色数据表（T_UserRoles）通过外键方式来维护。用户或角色对某一资源的授权信息则记录在ACL数据表（T_ACL）中。它们之间的关系及物理模型见图1。

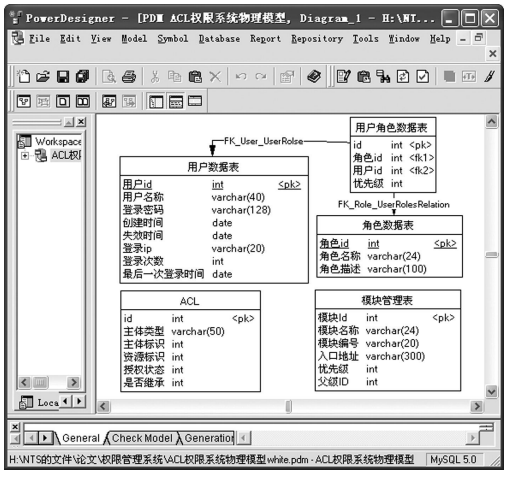

图1 权限管理数据表及关系的物理模型

Fig.1 Rights Management database tables and relations between the physical model

在ACL数据表中，主体类型一般是指用户或角色。当主体类型为用户时主体标识为用户id，当主体类型为角色时主体标识为角色id，资源标识就是指模块id，授权状态是一个正整数的低4位二进制表示的权限组合。

2 权限管理的实现

权限管理包括授权和认证。授权是指将某一资源的CRUD操作授权给用户或角色。认证是确认用户对某一资源是否具有相应权限的过程。

2.1 授权实现

授权包括角色授权和用户授权。角色授权就是先建立某一角色，然后对这个角色能访问的资源进行CRUD权限分配，再将角色分配给具有相同权限的用户。角色授权的优点是可以批量授权，可以极大地提高管理效率。用户授权，就是将资源的访问权限直接授权给某个具体的用户。

要实现权限的精粒度控制，就必须允许用户自行定义角色，然后对角色进行授权，而不是由软件设计者在写程序时就预先固定角色。而且这种精粒度的权限管理必须达到对菜单级的精细控制以及对WEB页面上的按钮级的精细控制。在B/S结构的软件系统中，对数据的增加、读取、修改、删除等操作都是通过WEB页面上的相应按钮或超链接的方式实现的。用户点击按钮将触发相应的请求，服务器端收到请求后调用业务逻辑完成相应的CRUD操作。例如，用户点击“增加”按钮，一般会出现增加数据的WEB页面，用户填写好相关的表单，单击“提交”按钮后，将触发“添加”（Create）请求并发送给服务器端，服务器端收到请求后在后台调用添加数据的业务逻辑，将数据持久化到数据库中，完成对数据的添加操作。也就是说，实现精细控制的权限管理后，如果用户对某一资源具有相应权限（例如删除权限），则在客户端的软件界面上“删除”按钮对该用户是可见的，否则就是屏蔽的。

在具体的程序实现过程中，授权就是将该角色能访问的资源标识（模块id）和角色标识（角色id）以及它对该资源具有的CRUD权限（一个正整数的低4位对应的二进制组合）写入ACL数据表，完成授权数据的持久化操作。

2.2 认证实现

认证就是用户登录后，系统根据他所拥有的权限，列出相应的菜单。具有不同权限的用户登录后，前台展现的菜单是不同的。

在具体实现程序时，认证就是从ACL数据表中检索出登录到系统的用户所能访问的资源标识（即模块id），形成前台显示菜单需要的数据格式，供前台显示菜单使用。同时，将用户所能访问的资源标识及用户对该资源具有的CRUD权限（一个正整数的低4位对应的二进制组合），写入session中，然后在视图（一般是jsp页面）上从session中取出用户的ACL控制列表，根据用户拥有的权限，在jsp页面上进行判断，然后根据判断结果分别显示“读取”、“增加”、“删除”、“更新”等按钮，实现精粒度的权限控制。

在编程过程中，一个用户可能拥有多个角色，同时也可能拥有单独的用户授权。此时，要先从数据库的T_ACL表中查询出该用户拥有的角色的

权限列表，再查询出该用户拥有的单独权限列表，对它们进行“与”运算，最后形成该用户的权限列表。在这一过程中，角色所拥有的权限可能会与用户单独拥有的权限重复，但通过“与”运算后，就可以剔除重复的权限。

2.3 编程实现

B/S结构的软件系统中，一般具有导航菜单，每一个导航菜单实现相应的功能。例如在“地震台站管理系统”中，导航菜单具有“台站管理”、“仪器管理”、“人员管理”、“用户管理”、“角色管理”等一级菜单，这些导航菜单对应软件中的模块，每个模块完成相应的功能。每一个导航菜单都具有菜单名称、入口地址（类似于网址）等信息，它们存放在数据库的模块管理数据表中。在读写数据库时，采用 Hibernate与数据库“打交道”，权限管理中所用的实体类（也叫 VO类、Poj类）类图及其关系见图 2。

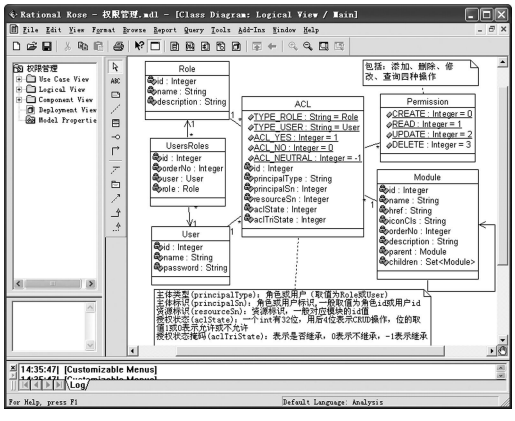

图 2　权限管理中的实体类类图及其关系（北京尚学堂科技有限公司，2008）

Fig. 2　Rights Management in the entity class diagram and their relations (Beijing Shangxue Tang Technology Co., Ltd., 2008)

在程序设计过程中，笔者采用 Struts2＋Hibernate＋Spring架构，授权部分使用 DWR框架。由于每个实体都要进行写入、读取、更新、删除、分页等操作，因此笔者使用 Java的泛型设计了一个 GenericDao基类，该基类继承自 Spring提供的 HibernateDaoSupport类，可以完成基本的 CRUD、分页等操作。对 ACL、User、Role等的 Dao操作则设计继承于 GenericDao的子类，只要传入相应的实体即可。同时，为了方便软件的维护，笔者设计了 Service层，在 Service层完成业务逻辑，并调用 Dao层完成对数据库的操作。前台与后台的“沟通”由 Struts2的 Action完成，在 Action中调用 Service层的业务逻辑来处理业务需求，然后将处理结果转发至前台 jsp页面。各种 Dao、Service、Action则统一由 Spring容器进行管理。

（1）授权过程：建立角色→对角色进行授权（资源与角色关联）→将角色分配给用户（角色与用户关联）。建立角色的过程十分简单，采用 Struts2的 Domain模型传递参数，Struts2在 Action中会自动根据前台传入的角色数据实例化一个 Role类，只需要在 Action中调用 roleService（完成对角色数据的 CRUD、分页等操作）的添加方法将其持久化到数据库中即可。而授权包含对角色的授权和单独对用户授权，当前台用户点击复选框（图 3）将其选中时，后台会根据主体标识（角色或用户）作出判断，当主体标识为“角色”时将主体标识（Role）、资源 id、角色 id、对该资源的 CRUD权限（一个正整数的低 4位）写入数据库；当主体标识为“用户”时将主体标识（User）、资源 id、用户 id、用户对该资源的 CRUD权限写入数据库。当取消复选框时，则应从数据库中删除以上信息。在实现这一操作时，笔者使用 DWR框架完成授权页面与后台业务逻辑 Service层的交互，在不刷新页面的情况下就可以完成以上操作。

图 3　精细权限管理在地震台站管理系统后台管理中的应用

Fig. 3　Fine right management in the Seismic Station Management System

（2）认证过程：验证用户登录→根据用户 id查找其拥有的角色，从 T_Ac数据表中查找角色

拥有的权限→根据角色的权限查询模块资源（菜单）列表，同时将权限列表写入 session→前台显示菜单，完成认证过程。在认证过程中，首先检查用户登录是否成功，成功后才查找用户的权限。

验证用户的操作是在 Struts2的 loginAction中完成的，如果验证成功，则调用业务逻辑 AclService，从数据库中检索出用户拥有的权限，根据权限从模块数据表中检索出模块信息放入 moduleList，同时将用户权限放入一个 acList，然后写入 session中。最后 loginAction转发到前台的主页面，主页面上用 Struts2的<s:iterator>标签遍历 moduleList，列出主菜单。需要注意的是，只要用户对某一资源（模块）具有 CRUD中的任何一个权限，主菜单就要列出相应的菜单，而用户是拥有 C、D、U还是 R权限，则需在具体的 WEB交互页面上读取 session中用户的权限列表，决定是否显示添加、删除、更新、编辑等按钮，这就是精细的权限控制。

在实际应用中，前台主菜单一般都需要美化处理。主界面有左侧树形菜单导航，也有顶部下拉菜单导航，具体采取哪种导航菜单，需根据项目的实际需要决定。而前台需要什么样的菜单数据，只需要在 Struts2的用户验证的 Action中构造就行了。在“地震台站管理系统”中，前台菜单是用 ExtJS框架展示的树形菜单，所以菜单资源返回的列表数据要转化成 json字符串。

3 权限管理在地震业务系统中的应用

3.1 应用实例

云南是地震多发地区，通过多年的建设，已形成了国家级、省级及市（县）级三级地震监测台站体系。为了提高台站管理效率，摸清全省的台站数目、观测项目、观测仪器、人员配置等资源，在云南省地震局青年基金的资助下，我们于2006年设计了基于 B/S的云南“地震台站管理系统”。该系统的建成，极大地提高了工作效率。在系统运行过程中，因实际工作需要，要求系统增加相应的功能。这时，我们发现原来的权限管理设计不利于管理新增加的需求的权限控制，需要修改的代码较多。在这种情况下，我们对“地震台站管理系统”中的权限管理重新进行评估，并按本文所述的思路及方法重构了原有的代码，对“地震台站管理系统”中的各种资源实现了精细的权限管理，并且实现了台站人员只能删除或修改自己添加的资料，其余资料只能查看的更为精细的权限控制（图 3）。

3.2 精粒度权限管理的优点

采用精粒度的权限管理后，提供给客户的软件产品可以由客户定制角色，而不是由程序员将角色及相应权限写死。对具体用户的权限也由系统管理员决定。精粒度权限管理还可以实现 Web页面上的“增加”、“删除”、“更新”、“读取”的按钮级的控制，在新的权限管理体系下，“地震台站管理系统”若需再增加新的需求，只需要编写新功能的实现代码，然后将新功能的菜单名称和入口地址加入到模块数据表中（具体由菜单管理模块实现），由管理员对新增的功能菜单进行授权即可，对原有的代码不作任何修改，这极大地提高了系统的维护效率，节约了系统的维护成本。

4 结语

本文讨论了精细的权限管理的实现方法，权限管理在 B/S结构的软件系统中具有举足轻重的地位。这种权限管理技术在办公自动化（OA）系统、CMS内容管理系统等各种 B/S的业务处理系统中得到广泛应用，而采用 Struts2+Hibernate+Spring框架则是当前 Java阵营的主流技术。采用这种精细的权限控制后，由于系统的主页面的菜单存储在数据库中，以后要扩展业务应用时，只需要设计相应的业务逻辑，然后将其入口地址等写入到数据库的模块数据表，不需要修改原来的业务应用就可以达到扩展的目的，真正实现了软件设计的积木效果，大大提高了软件的维护效率。因此，这种精细的权限管理技术可以作为 B/S软件系统的一个基本组件，只要是 B/S的软件，都可以在用它来“建造”，使软件的应用范围更加广泛。

参考文献：

北京尚学堂科技有限公司．2008 OA视频项目教程权限管理部分[EB/OL]．(2008－02－19)[2009－06－12]．http://www.very

cd. com/groups/@ u2857352/237347. topic.
蔡雪熹. 2006. Hibernate开发及整合应用大全[M]. 北京: 清华大学出版社.
刘敦敏. 2004. B/S模式的烟草信息管理系统权限管理的实现[EB/OL]. (2004－08－19)[2009－05－15]. http://www.tobacco.org.cn/news/dspNews.jsp?id=47993.
倪泰山, 阙云彩, 李祥, 等. 2009. 基于 B/S结构的地震台站综合信息管理系统的设计与实现[J]. 地震研究, 32(1): 89—93.
徐会生, 何启伟, 康爱媛. 2009. 深入浅出 Ext JS[M]. 北京: 人民邮电出版社.

Realization of the Precise Size of the Managenment of Rights of Software System Based on B/S

NI Tai-shan[1], ZHOU Yong-gang[2]
(1. Earthquake Administration of Honghe Prefecture of Yunnan Province, Mengzi 661100, Yunnan, China)
(2. Earthquake Administration of Yanqing County of Beijing, Beijing 102100, China)

Abstract

Firstly, we introduce the fine granularity of right management ideas and methods in the B/S structure software design. Then using Java language, adopting struts2 + hibernate + spring framework combined with technology we realize the right management software system, and applied the technology to the background management of the management system of seismic station of the Earthquake Administration of Yunnan Province.

Key words: B/S structure; database; rights management; access control list