第 33 卷 第 1 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.1 Computer Engineering January 2007
·开发研究与设计技术· 文章编号：1000—3428(2007)01—0271—03 文献标识码：A 中图分类号：TP319

通用自适应权限管理系统的研究与实现

宣明付，夏榆滨
(北京航空航天大学计算机学院，北京 100083)

摘 要：在分析现有权限管理系统缺陷的基础上，提出了运用 XML 来描述系统的功能、实体类和操作等相关信息，并以此作为权限管理系统的基础设计和实现了通用自适应的权限管理系统，以应对系统在开发、应用和升级过程中出现的各种变化。
关键词：权限管理；XML；自适应

Research and Implementation of Universal and Self-adaptive Authority Management System

XUAN Mingfu, XIA Yubin
(College of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】On the basis of analyzing existing authority management system, this paper presents an universal and self-adaptive authority management system, takes advantage of using XML to describe function, entity class and operation message. This system can handle any changes which appear in the processes of system development, application and upgrade.
【Key words】Authority management; XML; Self-adaptive

1 概述

1.1 简况

随着应用的普及，权限管理的模型和对象日趋复杂；同时，随着应用系统功能的改变和实际业务的变化，权限管理系统也需要作相应的调整，这些都对权限管理系统提出了越来越高的要求。鉴于权限管理的复杂性和多变性，实现一个通用的、适应性强的易于开发、维护和升级的权限管理系统是在开发应用系统时需要重点考虑的。

从权限管理模型上来看，目前常见的权限管理系统主要采用基于角色的权限管理模型，它的基本思想是根据企业中的职能岗位划分出不同的角色，将系统资源的访问权限附加到角色上，并为用户分配相应的角色，通过控制角色权限来间接地控制用户对系统资源的访问。

从权限管理的对象来说，通常包括功能权限设置、类权限设置、规则权限设置和属性权限设置。功能权限设置用于对系统功能进行控制；类权限设置用于控制用户对业务数据的访问；规则权限设置用于建立用户与类及其属性的约束关系，使得用户在访问系统数据时受到所设规则的限制；属性权限设置用于控制用户对实体类的具体属性的访问。

从权限管理系统所提供的功能来看，主要包括授权和访问控制两大模块，分别实现权限的授予和权限的验证。授权模块主要用于对角色、用户和用户组进行管理，包括基本信息的维护，角色、用户和用户组之间关系的维护，以及分别为角色、用户和用户组授予各类权限。访问控制模块实现对登录用户的身份认证，并在用户进入系统执行操作时验证用户的操作权限。

1.2 现有权限管理系统的不足

通过分析现有的权限管理系统，可以看出它们通常是针对某一个应用系统单独开发，与所在应用系统联系紧密，要考虑和解决的是系统当前的问题。这类权限管理系统一般能够满足应用系统当前的需求，但是随着应用系统开发的不断深入、实际需求的不断变化、系统应用环境的改变以及系统升级和集成的需要，都会或多或少地暴露以下两个主要问题。

(1)适应性差

权限管理系统在进行授权和访问控制时需要读取整个应用系统的功能、实体类和操作等相关信息，一般系统会从数据库读取这些信息，也有将这些信息直接编入代码的，不管采用哪种方式，都会使系统过于僵化，难以适应在系统开发、应用、升级等过程中遇到的各种变化。

在系统的开发过程中，由于各子系统的功能、实体类和操作等在开发过程中都是不断改变，逐渐增加的，因此权限管理系统很难实现与各子系统并行开发。

在系统的应用过程中，企业的业务会不断地发生变化，业务的变化会导致功能和数据的变化，这些要求权限管理也要作相应的改变，而固化了功能和实体类等信息的权限管理系统无法适应这样的改变。

系统每一次的升级，系统的功能、实体类和操作等都有改变。权限管理系统应能适应这样的改变，并且能够保证系统升级后，原有的权限设置仍然被继承下来。然而，按照现有模式开发的权限管理无法满足这种需求。

(2)紧耦合，难以复用

权限管理的两大功能，即授权和访问控制。授权时权限管理系统需要应用系统的功能、实体类和操作等信息。进行访问控制时应用系统会调用权限管理系统提供的相应功能。

基金项目：国家“863”计划“CIMS 主题”基金资助项目(2001AA412030)
作者简介：宣明付(1982－)，男，硕士生，主研方向：Web Services，XML，企业信息系统集成，企业信息管理软件设计开发；夏榆滨，副教授
收稿日期：2006-01-27 **E-mail：**xmf@cse.buaa.edu.cn

现有的权限管理系统通常都是为某个应用系统专门开发的，在设计时权限管理系统和相应的应用系统在授权和访问控制处联系紧密，应用系统的功能、实体类和操作信息一旦改变，权限管理系统就必须作调整，或权限管理系统的控制方式改变了，相应的应用系统也必须作调整，这样会使权限管理系统与相应的应用系统紧密地联系在一起，各自的修改都会影响到对方，导致权限管理系统很难独立出来应用到其它系统上。

2 通用自适应权限管理系统

2.1 基本思想

XML（eXtensible Markup Language）是一套定义语义标记的规则，这些标记将文档分成许多部件并对这些部件加以标识。它也是元标记语言，即定义了用于定义其它与特定领域有关的、语义的、结构化的标记语言的句法语言。利用 XML 可以用来建立包含结构化格式数据的文档，XML 具有自描述、可扩展的特性，并且得到了广泛的支持和应用。

基于 XML 的自适应权限管理系统的基本设计思想是使用 XML 定义系统的功能、实体类和操作等信息，权限管理系统运用这 3 类 XML 文档，动态地获得系统的功能、实体类和操作信息进行权限设置，并对已设置的权限进行修改。功能权限设置使用定义功能的 XML 文档，类权限设置、规则权限设置和属性权限设置使用定义实体类的 XML 文档。另外，类权限设置还有用到定义操作的 XML 文档。

2.2 通用性

权限管理系统通用性的目标是要在不作修改的情况下将权限管理系统应用到不同的系统上，实现对不同的应用系统的权限控制。而实现通用性的途径将权限管理系统的两大功能，即授权和访问控制与应用系统解耦。

授权时，权限管理与应用系统的联系体现在权限管理系统需要应用系统的功能、实体类和操作的相关信息，以设置相应的权限。通过 XML 文档来描述这些信息，只要提供的信息满足权限管理系统的需求，则权限管理系统不需要访问应用系统也可获得所需信息。因为权限管理系统对 XML 文档的结构有自己的要求，而各个应用系统可能已经存在相应的 XML 文档，内容满足需求，但结构不一样，为了实现对已有 XML 文档的利用而不增加额外的工作量，可以运用 XSLT（eXtensible Stylesheet Language Transformation）对 XML 文档进行转换。XSLT 是一种用来转换 XML 文档结构的语言，通过它可以很方便地改变 XML 文档的结构，甚至将 XML 转换为 HTML 或其它文本格式。

在访问控制时，权限管理系统与应用系统的联系在于应用系统需要调用权限管理系统提供的方法进行访问控制，如用户身份验证、操作权限验证等。通常这种耦合是必需的，但它会带来开发的困难，因为实现业务逻辑的代码中充斥着进行权限验证的代码，即影响调试，也容易出错。所以，为了减少这种耦合，加快系统的开发和提高系统的可维护性，可以将应用系统对权限管理系统的方法调用从业务逻辑代码中剥离出来，并且集中在应用系统的局部代码中。这主要体现在访问控制的 3 个方面，即用户身份验证、功能和操作权限验证和数据访问权限验证。用户身份验证只局限在用户登录模块中；对于功能和操作权限验证，可以在用户登录系统时调用相应方法，根据用户所拥有的功能和操作权限构造相应的系统菜单，此后不用再进行调用。对于数据访问权限，可以集中于数据访问层进行调用，对于使用了对象关系映射的系统来说，将其集成到 ORM 工具中会更加方便。

以上的解决办法适用于一个新开发的系统，如果是将一个现有应用系统的权限控制部分转移到本权限管理系统上来，以上方法显然不合适，因为应用系统原来的权限管理系统提供的访问控制接口与本权限管理系统通常不一致。解决办法之一是应用适配器，适配器意在“将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。”图 1 显示原权限访问控制类结构，其中 Client 表示应用程序，OldAccessControl 表示原权限管理系统提供的访问控制类。Client 通过调用 OldAccessControl 提供的方法实现用户的登录、退出、操作权限验证、数据访问权限验证等。

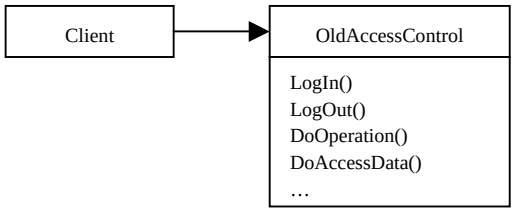

图 1 原权限访问控制类结构

图 2 显示的是将一个现有应用系统的权限控制部分转移到本权限管理系统上来后的访问控制类结构，原访问控制类 OldAccessControl 与本权限管理系统提供的访问控制类 NewAccessControl 接口不一致，系统中将 OldAccessControl 用一个抽象类或接口 Adapter 代替。Adapter 的接口与 OldAccessControl 的一致， Client 调用 Adapter 的方法，而 Adapter 具体则由 NewAccessControl 来实现，例如 Adapter 的 Login()方法的具体实现是 NewAccessControl 的 LogOn()。如此，则可将现有应用系统的权限控制平滑地移到本权限管理系统上来。

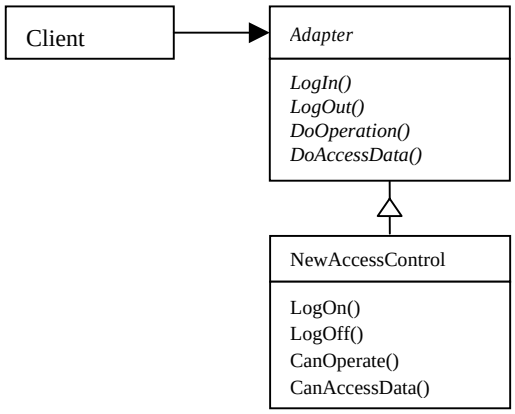

图 2 新的权限访问控制类结构

2.3 自适应性

应用系统都有它一定的生命周期，在其整个生命周期过程中由于各种原因需要作出改变。提高系统的适应性，减少相应的修改通常是软件设计人员和开发人员考虑的重点。很好的适应性对于权限管理系统尤为重要，因为影响它改变的因素更多更复杂，既有业务的需求，更有相关系统的需求。

运用 XML 定义系统的功能、实体类和操作信息，随着系统的开发、升级或者实际应用需求的改变，对 XML 文件作相应的修改，并对已有角色、用户和用户组的权限重新进行运算，保留已有的权限设置，去除多余的权限设置，对新

添加的功能、实体类的操作权限设为默认值，便可以使权限管理系统自动地适应系统中各个子系统的开发。

在系统升级时不必考虑权限系统的升级，只要在升级后运用权限系统提供的工具在新的 XML 文件的基础上对已有的权限重新进行运算即可。系统的用户也可以根据实际业务的发展，对 XML 文件作适当地修改，即可适应新的业务需要，完全不需要开发人员的参与。

3 自适应权限管理系统的设计

3.1 XML 文档结构

本权限管理系统主要用到 3 类 XML 文档，分别用于定义系统的功能、实体类和操作。下面以笔者参与设计和开发的 ERP 系统为例，分别对这 3 类文档的内容及结构作简要说明。

(1)功能 XML 文档：功能 XML 文档定义了系统的所有功能及其结构，根据实际情况，整个文档可以先按子系统划分，子系统中划分出模块，模块中包括具体的功能。下文是一段简化的功能 XML 文档的示例，其中每个节点的 CN 属性表示相应子系统、模块或功能的中文名称。

```
<?xml version="1.0" encoding="utf-8"?>
<Function>
<!—子系统-->
<SystemMage CN="系统管理">
    <!—模块-->
    <BasicDataMata CN="基础数据维护">
      <!—功能->
      <AreaMata CN="地区代码维护">
      </AreaMata>
      <PayMannerMata CN="付款方式维护">
      </PayMannerMata>
    </BasicDataMata>
    <!—其他模块和功能-->
    …
  </SystemMage>
  <!—其他子系统-->
  …
</Function>
```

(2)实体类 XML 文档：实体类 XML 文档包含了应用系统中所有的实体类及其属性信息，实体类可以按子系统组织，或将子系统的信息作为属性添加到类结点中。下文所示的是包含一个名为 WorkShop 和 Area 实体类的简化 XML 文档。

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityClass>
  <!—车间实体类-->
  <WorkShop CN="车间" SubSystem="车间管理">
    <ID CN="编号" />
    <Name CN="名称" />
    <Description CN="说明" />
  </WorkShop>
<Area CN="地区" SubSystem="系统管理">
    <ID CN="编号" />
    <Name CN="名称" />
<UpperAreaID CN="上级地区编号" />
    <Description CN="说明" />
  </Area>
  <!—其他实体类-->
  …
</ EntityClass >
```

(3)操作 XML 文档：不同的应用系统，可以对实体类进行的操作也是不同的。并且，当实体类处于不同状态时，所能施加的操作也不尽相同。另外，操作之间还有一定的逻辑关系，例如拒绝查看的同时要拒绝修改等。操作 XML 文档即是用来定义系统中实体类在不同状态下可以施加其上的操作以及操作之间的关系的。如下文所示，操作 XML 文档首先按实体类的状态划分，状态中的每个结点表示一种操作，结点属性 PRI 表示此操作与状态中其它操作比较所具有的权限级别，如果两个或多个操作具有相同的级别，设置相同的数据即可，对于与其它操作没有关系的操作，可以设置一个特殊的级别，如 0。在拒绝级别低的操作时，自动拒绝级别高的操作，在允许级别高的操作时，自动允许级别低的操作，具体应用可以根据各个权限管理系统的规则有所不同。

```
<?xml version="1.0" encoding="utf-8"?>
<Operation>
  <!—签出状态-->
  <CheckOut CN="签出">
    <Delete CN="删除" PRI=" 0" >
    </Delete>
    <Modify CN="修改" PRI="3 ">
    </Modify>
    <Query CN="查看" PRI="1 ">
    </Query>
    <Print CN="打印" PRI="2 ">
    </Print>
    <!—其他操作-->
    …
</ CheckOut >
<!—其他状态-->
  …
</Operation>
```

上面介绍的是简化的 XML 文档，实际使用时可以根据需求添加更多的信息。随着 XML 应用的普及，很多现有系统已经大量地运用到了 XML，如运用 XML 动态构造系统的菜单，运用 XML 实现对象关系映射等。这些 XML 文档中所定义的数据与权限管理系统所需的 XML 文档类似，稍作修改后即可重用到权限管理系统中来，这样可以保证数据的唯一性、一致性和即时性。

3.2 功能设计

本权限管理系统包括了基本的角色管理、用户组管理、用户管理、在线用户管理等功能，并且根据自适应权限管理的需要实现系统同步功能和系统配置功能。系统配置功能用于对系统的功能、类和属性权限进行初始化默认设置，当新的角色、用户或用户组添加到系统中时，即可拥有默认的权限，减少了重复设置权限。系统同步功能用于在系统功能、实体类或操作发生改变，即相关 XML 文档改变后，调整系统权限的默认设置和已有角色、用户和用户组的权限设置，即保留已有的，删除多余的，对于新增的设为默认值。角色管理、用户组管理和用户管理中关键的功能则是对各种对象的权限设置，此处不再赘述。

4 结束语

相对于传统的权限管理系统，本文讨论的自适应权限管理系统能够达到加快开发进度降低开发成本和升级代价，保存历史数据，减少冗余数据目的。本权限管理系统已经在笔者参与开发的 ERP 系统中完整地实现，随着相关子系统的开发和应用，本权限管理系统也得到了充分的验证，达到了预

（下转第 276 页）