



(12) 发明专利申请

(10) 申请公布号 CN 101902402 A

(43) 申请公布日 2010.12.01

(21) 申请号 201010238752.3

(22) 申请日 2010.07.21

(71) 申请人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦法务部

(72)发明人 陈文斌

(74) 专利代理机构 北京银龙知识产权代理有限公司 11243

代理人 许静

(51) Int. Cl.

H04L 12/56 (2006.01)

H04L 29/06 (2006.01)

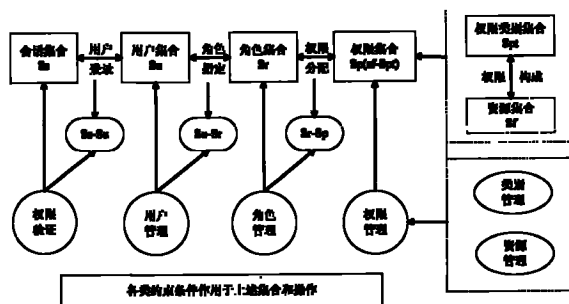
权利要求书 2 页 说明书 11 页 附图 3 页

(54) 发明名称

一种用户权限管理方法、装置

(57) 摘要

本发明提供的用户权限管理方法及装置,通过在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型中,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;确定用户访问更新后的所述权限集合的访问权限;基于确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。从而使系统的权限集合可以适应系统资源集合的动态调整,而且还可以使系统为用户授权更加灵活。



1. 一种用户权限管理方法,其特征在于,包括:

在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型中,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;

确定用户访问更新后的所述权限集合的访问权限;

基于确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。

2. 如权利要求 1 所述的方法,其特征在于,所述在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型中,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合的过程中,包括增加系统资源流程和删减系统资源流程,其中,所述增加系统资源流程包括:

将增加的系统资源添加至资源集合中,更新所述资源集合中的资源;

根据所述增加的系统资源对应的操作权限类别编码,将权限类别集合中对应的操作权限类别加载至所述增加的系统资源;

将所述更新后的资源集合与权限类别集合组合,构成更新后的权限集合;

所述删减系统资源流程包括:

将减少的系统资源从资源集合中删除,更新所述资源集合中的资源;

将所述更新后的资源集合与权限类别集合组合,构成更新后的权限集合。

3. 如权利要求 1 所述的方法,其特征在于,所述确定用户访问更新后的所述权限集合的访问权限包括:

通过会话进程与用户交互,确定用户身份;

根据确定的所述用户身份与所述用户所属角色之间的对应关系,确定所述用户所属角色;

根据确定的所述用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色对应的访问权限。

4. 如权利要求 3 所述的方法,其特征在于,若根据确定的所述用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色不具有访问所述权限集合的访问权限,则所述方法还包括:

将用户所属角色不具有访问所述权限集合的访问权限的确定结果反馈至用户。

5. 如权利要求 1 或 4 所述的方法,其特征在于,所述确定用户访问更新后的所述权限集合的访问权限包括:

根据接收的所述用户发送的特别授权请求,为所述用户授予不基于用户所属角色访问所述权限集合的访问权限。

6. 一种用户权限管理装置,其特征在于,包括:

更新模块,用于在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;

确定模块,用于确定用户访问所述更新模块更新后的权限集合的访问权限;

权限管理控制模块,用于基于所述确定模块确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。

7. 如权利要求 6 所述的装置,其特征在于,所述更新模块包括:

加载单元,用于在系统资源增加时,根据增加的系统资源对应的权限类别集合编码,将

权限类别集合中对应的操作权限类别加载至所述增加的系统资源。

8. 如权利要求 6 所述的装置,其特征在于,所述确定模块包括:

身份确定单元,用于通过会话进程与用户交互,确定用户身份;

角色确定单元,用于根据所述身份确定单元确定的用户身份与所述用户所属角色之间的对应关系,确定所述用户所属角色;

访问权限确定单元,用于根据所述角色确定单元确定的用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色对应的访问权限。

9. 如权利要求 8 所述的装置,其特征在于,所述确定模块还包括:

提示单元,用于在所述访问权限确定单元确定所述用户所属角色不具有访问所述权限集合的访问权限时,将访问权限确定单元的确定结果反馈至用户。

10. 如权利要求 6 或 8 所述的装置,其特征在于,所述确定模块还包括:

特别授权单元,用于根据接收的所述用户发送的特别授权请求,为所述用户授予不基于用户所属角色访问所述权限集合的访问权限。

一种用户权限管理方法、装置

技术领域

[0001] 本发明涉及权限管理技术领域，具体涉及一种用户权限管理方法、装置。

背景技术

[0002] 目前在企业应用系统中，普遍采用基于角色的访问控制（RBAC, Role-based Access Control）模型，对系统用户的访问权限进行管理控制。如图 1 所示。RBAC 把系统用户根据其所拥有的执行功能和安全策略分成不同的角色，然后对每个角色分配对应的权限，再通过对每个用户指定不同的角色，来实现对用户的权限管理控制。

[0003] 在现有 RBAC 模型中，通常把系统中的权限集合进行了固定的限制，即把系统的资源（功能）集合以及施加于其上的权限类别集合（即操作类别集合）进行了固化绑定，作为系统的权限集合，而现有 RBAC 模型在实际应用时，特别是在 Web 系统中，因为 Web 系统拥有复杂的页面层次和表单元素，而这些内容在开发和维护中的变化是非常频繁和不可预期的，这就导致了当系统管理的资源集合处于动态变化的时候，系统的权限集合不能与系统资源动态适应。

[0004] 可以看出，现有技术条件下的 RBAC 模型在具体实现过程中，权限集合不具备与系统的资源集合动态适应的能力，缺少灵活性。

发明内容

[0005] 本发明所要解决的技术问题是提供一种用户权限管理方法、装置。从而使 RBAC 模型中的权限集合具备与系统资源动态适应能力。

[0006] 为解决上述技术问题，本发明提供方案如下：

[0007] 本发明实施例提供了一种用户权限管理方法，包括：

[0008] 在系统资源发生变化时，更新扩展后的基于角色的访问控制 RBAC 模型中，独立于权限类别集合的资源集合中的资源，并利用更新后的资源集合更新权限集合；

[0009] 确定用户访问更新后的所述权限集合的访问权限；

[0010] 基于确定的所述用户的访问权限，对所述用户发送的访问请求进行权限管理控制。

[0011] 优选的，所述方法中，所述在系统资源发生变化时，更新扩展后的基于角色的访问控制 RBAC 模型中，独立于权限类别集合的资源集合中的资源，并利用更新后的资源集合更新权限集合的过程中，包括增加系统资源流程和删减系统资源流程，其中，所述增加系统资源流程包括：

[0012] 将增加的系统资源添加至资源集合中，更新所述资源集合中的资源；

[0013] 根据所述增加的系统资源对应的操作权限类别编码，将权限类别集合中对应的操作权限类别加载至所述增加的系统资源；

[0014] 将所述更新后的资源集合与权限类别集合组合，构成更新后的权限集合；

[0015] 所述删减系统资源流程包括：

- [0016] 将减少的系统资源从资源集合中删除,更新所述资源集合中的资源;
- [0017] 将所述更新后的资源集合与权限类别集合组合,构成更新后的权限集合。
- [0018] 优选的,所述方法中,所述确定用户访问更新后的所述权限集合的访问权限包括:
- [0019] 通过会话进程与用户交互,确定用户身份;
- [0020] 根据确定的所述用户身份与所述用户所属角色之间的对应关系,确定所述用户所属角色;
- [0021] 根据确定的所述用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色对应的访问权限。
- [0022] 优选的,所述方法中,若根据确定的所述用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色不具有访问所述权限集合的访问权限,则所述方法还包括:
- [0023] 将用户所属角色不具有访问所述权限集合的访问权限的确定结果反馈至用户。
- [0024] 优选的,所述方法中,所述确定用户访问更新后的所述权限集合的访问权限包括:
- [0025] 根据接收的所述用户发送的特别授权请求,为所述用户授予不基于用户所属角色访问所述权限集合的访问权限。
- [0026] 本发明实施例还提供了一种用户权限管理装置,包括:
- [0027] 更新模块,用于在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;
- [0028] 确定模块,用于确定用户访问所述更新模块更新后的权限集合的访问权限;
- [0029] 权限管理控制模块,用于基于所述确定模块确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。
- [0030] 优选的,所述更新模块包括:
- [0031] 加载单元,用于在系统资源增加时,根据增加的系统资源对应的权限类别集合编码,将权限类别集合中对应的操作权限类别加载至所述增加的系统资源。
- [0032] 优选的,所述确定模块包括:
- [0033] 身份确定单元,用于通过会话进程与用户交互,确定用户身份;
- [0034] 角色确定单元,用于根据所述身份确定单元确定的用户身份与所述用户所属角色之间的对应关系,确定所述用户所属角色;
- [0035] 访问权限确定单元,用于根据所述角色确定单元确定的用户所属角色与所述权限集合之间的对应关系,确定所述用户所属角色对应的访问权限。
- [0036] 优选的,所述确定模块还包括:
- [0037] 提示单元,用于在所述访问权限确定单元确定所述用户所属角色不具有访问所述权限集合的访问权限时,将访问权限确定单元的确定结果反馈至用户。
- [0038] 优选的,所述确定模块还包括:
- [0039] 特别授权单元,用于根据接收的所述用户发送的特别授权请求,为所述用户授予不基于用户所属角色访问所述权限集合的访问权限。

[0040] 从以上所述可以看出,本发明提供的用户权限管理方法及装置,通过在系统资源发生变化时,更新扩展后的基于角色的访问控制 RBAC 模型中,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;确定用户访问更新后的所述权限集合的访问权限;基于确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。从而使系统的权限集合可以适应系统资源集合的动态调整,而且还可以使系统为用户授权更加灵活。

附图说明

- [0041] 图 1 为现有技术中 RBAC 模型结构示意图;
- [0042] 图 2 为本发明实施例提供的 RBAC 模型结构示意图;
- [0043] 图 3 为本发明实施例提供的 RBAC 模型所涉及的封装格式示意图;
- [0044] 图 4 为本发明实施例提供的用户权限管理方法具体实现流程图一;
- [0045] 图 5 为本发明实施例提供的用户权限管理方法生成用户权限流程图;
- [0046] 图 6 为本发明实施例提供的用户权限管理方法具体实现流程图二;
- [0047] 图 7 为本发明实施例提供的用户权限管理装置具体实现结构示意图;
- [0048] 图 8 为本发明实施例提供的用户权限管理装置中确定模块具体实现结构示意图。

具体实施方式

[0049] 本发明中提供了一种系统资源可动态管理的扩展 RBAC 模型,具体可如附图 2 所示。

[0050] 针对现有的 RBAC 模型通常是把系统中的权限集合进行了固定的限制,即把系统的资源(功能)集合以及施加于其上的权限类别集合(即操作类别集合)进行了绑定,作为系统的权限集合,从而导致现有 RBAC 模型不具有灵活性的问题,本发明实施例提供了如附图 2 所示的扩展 RBAC 模型,该 RMAC 模型中,将权限集合分解成图 2 右侧虚线方框内的两个独立设置部分:权限类别集合与资源集合,使得权限集合由这两部分组合构成。这样,当系统的资源集合随着用户需求变化而动态调整时,权限集合也会相应的动态调整。

[0051] 具体的,当系统新增一个资源 F(F 可假设是个页面等)时,则系统可以将资源 F 添加至资源集合中,更新独立设置的资源集合中的资源。并且,系统还可以将独立设置的权限类别集合中,与资源 F 对应的操作权限类别,如增加、删除、修改等操作加载至资源 F 上,即增加对资源 F 的增加、删除、修改等操作权限。由于资源集合是独立于权限类别集合设置的,因此,资源集合的变动,并不会影响权限类别集合,这样,由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合,即利用更新后的资源集合,更新权限集合。

[0052] 由于权限类别集合中的操作权限类别存在唯一的编码,所以可根据资源的前缀编码来判断新增加的资源需要加载哪些权限类别集合中的操作权限类别。比如增加、删除、修改的操作分别对应 1、2、3 的编码,而系统新增加的资源的前缀编码为 1、2、3,则系统为新增加的资源加载增加、删除、修改操作权限类别。

[0053] 而在系统资源减少时,系统将减少的资源从资源集合中删除,更新独立设置的资源集合中的资源,然后由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合。

[0054] 通过上述描述可以看出,通过将资源集合以及权限类别集合各自独立设置,并且有资源集合与权限类别集合组合,构成权限集合,从而可以实现系统权限集合能够适应系统资源集合动态变化的要求。因为资源集合是独立设置的,所以其动态调整也不会影响系统的结构。

[0055] 下面对本发明实施例提供的扩展 RBAC 模型(如附图 2 所示)的具体组成进行详细的说明:

[0056] 资源集合 $S_f: \{F_1, F_2, \dots, F_i \dots\}$, 具体可以定义为系统中资源或功能的集合,在 Web 系统中一般以页面和功能模块的形式存在。资源集合中的元素 F_i 具体可以包括 F_{code} 、 F_{title} 、 F_{level} 、 F_{url} 、 F_{else} 等参数,其中, F_{code} 是系统资源的编码, F_{title} 是系统资源的标题, F_{level} 是系统资源的级别(在用菜单或树型结构组织资源的系统中非常重要), F_{url} 是系统资源的访问路径, F_{else} 是系统资源的其它描述信息。

[0057] 权限类别集合 $S_{pt}: \{Pt_1, Pt_2, \dots, Pt_i \dots\}$, 具体可以定义为系统中操作权限类别的集合,例如新增、修改、删除、审核等。权限类别集合中的元素 Pt_i 具体可以包括 Pt_{code} 、 Pt_{type} 、 Pt_{desc} 等参数,其中, Pt_{code} 是操作权限类别编码, Pt_{type} 是操作权限类别分类(在不同的功能流程中需要不同的权限分类方法), Pt_{desc} 是操作权限类别的描述信息。

[0058] 权限集合 $S_p: \{S_{pt} \times S_f\}$, 具体可以定义为系统的权限集合,它规定了系统中每个系统资源的所有可用操作权限类别。本发明中,权限集合是通过资源集合 S_f 与权限类别集合 S_{pt} 集合关联组合形成的,它会随着系统资源的改变而动态变化,从而避免了现有 RBAC 模型权限集合中固定资源可操作权限类别的做法。权限集合中的元素 P_i 可表示为一个二元组 $P: (F_{code}, Pt_{code})$, 其中 F_{code} 是资源集合中的资源编码, Pt_{code} 是权限类别集合中的权限类别编码。

[0059] 角色集合 $S_r: \{R_1, R_2, \dots, R_i, \dots\}$, 具体可以定义为系统角色的集合,角色集合中的元素 R_i 具体可以包括 R_{code} 、 R_{name} 、 R_{type} 等参数,其中, R_{code} 是角色编码, R_{name} 是角色名称, R_{type} 是角色类型(在系统通常可以根据职能部门、职位层次进行角色类型的区分)。

[0060] 用户集合 $S_u: \{U_1, U_2, \dots, U_i, \dots\}$, 具体可以定义为系统用户的集合,用户集合中的元素 U_i 具体可以包括 U_{code} 、 U_{name} 、 U_{else} 等参数,其中, U_{code} 是用户编码, U_{name} 是用户名称, U_{else} 代表用户的其他属性信息,包括密码、联系方式等。

[0061] 会话集合 $S_s: \{S_1, S_2, \dots, S_i, \dots\}$, 具体可以定义为系统当前的会话集合,其中每个会话都存有登录用户的相应信息和会话的上下文环境。

[0062] 本发明实施例提供的扩展 RBAC 模型中,上述集合可以采用二维关系的存储模式,具体可通过如附图 3 所示的以下步骤,按附图 2 中从右到左的顺序推导出用户权限的生成过程:

[0063] 1) 生成系统权限集合。

[0064] 首先建立系统的资源集合 S_f , 即明确系统中的哪些页面(甚至页面元素)和模块作为系统资源参与权限控制,为其指定唯一的编码 F_{code} 和其他必要属性。然后利用资源集合 S_f 与权限类别集合 S_{pt} 进行关联,得到一个对照表 (S_f-S_{pt}), 即系统权限集合 S_p 。

[0065] 2) 为角色分配权限。

[0066] 首先根据系统要求,按照部门或级别等划分原则,确定系统的角色集合 S_r 。然后针

对角色集合 S_r 中的每个元素 R_i , 在权限集合 S_p 中为其寻找应有的权限子集 P' ($P_1, P_2, \dots, P_n$), 建立角色集合与权限集合之间的对应关系, 形成角色—权限对照表 (S_r-S_p)。

[0067] 3) 为用户指定角色。

[0068] 新建用户, 加入系统用户集合 S_u 中, 然后在角色集合 S_r 中根据需要为其寻找应有的角色子集 R' ($R_1, R_2, \dots, R_n$), 建立用户与角色集合之间的对应关系, 形成用户—角色对照表 (S_u-S_r)。

[0069] 4) 为用户指定会话进程。

[0070] 由系统为用户指定其对应的会话进程 S_s , 建立会话—用户对照表 (S_s-S_u), 其中每个会话都存有登录用户的相应信息和会话的上下文环境。

[0071] 这样, 系统的权限关联机制已经形成。当某用户登录系统后, 系统会建立一个会话进程, 通过查询会话—用户对照表 (S_s-S_u), 得到该用户身份; 再通过查询用户—角色对照表 (S_u-S_r), 得到用户的角色集合; 再通过查询角色—权限对照表 (S_r-S_p), 最终可以得到用户在系统中的权限集合, 即明确了用户可以对哪些资源集合进行那些类别的访问操作。从而实现对用户访问权限的管理控制。

[0072] 在本发明实施例提供的扩展 RBAC 模型一个具体实施例中, 还可以为用户进行特别授权, 使用户可以在基于事先分配的角色, 无法访问某个权限集合时, 可以通过特别授权, 具有访问该权限集合的访问权限。这是因为现有 RBAC 模型限定了系统的所有用户的权限都依附于某个 (或某些) 角色的权限 (或权限集合), 这种限定虽然解决了用户授权的简单性, 但有时候也存在明显的不足, 例如, 当某用户由于业务要求需暂时拥有某项权限时, 而此时用户现有角色均不合适, 那么, 如果给该用户所属角色赋予此权限, 则必然同样会给予该角色的所有用户增加此权限, 这显然不合适; 如果再建一个合适的新角色, 然后赋予该用户, 但因为用户的该项权限只是暂时的, 所以权限撤销时又要删除相应角色, 显得比较繁琐。

[0073] 而在本发明实施例提供的扩展 RBAC 模型一个具体实施例中, 为了解决上述问题, 增加了为用户特别授权的过程, 使得系统对用户授权更加灵活, 以解决有些用户对暂时权限的方便授予和撤销。在具体应用是, 若用户无法根据事先分配的角色访问需要的权限集合, 则用户可以向系统发出特别授权请求, 由系统 (具体可以为管理员) 直接为用户分配访问特定权限集合的特别访问权限, 从而使用户具有不基于用户所属角色的特殊访问权限。并在用户完成访问后, 直接撤销用户的权限。这种方法解决了现有技术中, 临时权限通过 RBAC 模型中的角色授予存在的复杂性管理问题, 提高了用户授权的灵活性和可维护性。

[0074] 在本发明实施例提供的扩展 RBAC 模型一个具体实施例中, 除在用户浏览器界面进行用户权限验证之外, 还可以在系统后台程序代码中, 增加用户访问权限的验证过程, 从而实现浏览器和服务端双重验证, 保证了权限控制的安全性和正确性。本发明实施例对于如何在系统后台程序代码中增加用户访问权验证过程的表现形式并不限制。

[0075] 在本发明实施例提供的扩展 RBAC 模型一个具体实施例中, 还可以实现页面的局部更新, 从而减少了系统响应时间, 避免了占用过多带宽的情况。这是因为在现有 RBAC 模型中, 传统的权限页面应用允许用户填写表单 (form), 当提交表单时就向权限服务器发送一个请求。服务器接收并处理传来的表单, 然后返回一个新的网页。由于在前后两个页面中的大部分超文本标识语言 (HTML: Hyper Text Markup Language) 代码往往是相同的, 而用

户需要操作的只是页面中的一个局部,而页面其他部分是无需改动的,因此,现有 RBAC 模型在实现与用户交互时,存在浪费带宽的情况。而且,由于每次应用的交互都需要向服务器发送请求,应用的响应时间就依赖于服务器的响应时间。这导致了用户界面的响应比本地应用慢得多。

[0076] 下面对本发明扩展 RBAC 模型一个具体实施例,基于异步 JavaScript 和可扩展标记语言 (XML:Extensible Markup Language) 技术,即 AJAX 技术实现局部页面更新的具体实现过程为例进行说明,具体可以包括:

[0077] 1) 客户端接收用户访问请求。

[0078] 首先用户在客户端将必要信息(如操作员名称、密码等)在 WEB 页面上填写好,然后点击提交,向系统发送访问请求。

[0079] 为了便于理解,下述描述以本实施例中所涉及的扩展 RBAC 模型采用 JAVA Spring 框架进行构建为例进行说明。那么,上述访问请求即可以具体为 XML 超文本传输协议 (Http:HyperText Transfer Protocol) 请求 (Request) 对象。

[0080] 2) 系统对用户的访问请求进行重新封装后发送至系统后台。

[0081] 本发明中为了调用后台程序更加便捷,可以对 XMLHttpRequest 对象进行封装,具体的封装格式可如附图 3 所示,其中,接口名可为 callUrl(url, inparas, outparas, isSyn),即通过统一资源定位符 (URL, Uniform Resource Locator) 方式调用服务。参数 Url 是后台类路径和后台方法的结合,其中后台类路径与 JAVA 后台管理程序 spring 容器的 javabean 相对应。参数 inparas 是输入参数,具体可以为基本数据类型和对象。参数 outparas 是输出参数,其同样可以为基本数据类型和对象。参数 isSyn 是指是否异步,当需要同步时,将此配置为 true,否则为 false。

[0082] 本发明通过上述对 XMLHttpRequest 对象的封装,将用户发送的访问请求发送给系统后台。

[0083] 3) 系统后台根据用户的访问请求进行数据处理。

[0084] 在系统后台,具体可由 Web 容器接收封装后的访问请求,并将访问请求转发至业务层。

[0085] 业务层在解析了访问请求所包含的参数之后,调用持久层相关的数据库处理逻辑,把结果序列化成 XML,并把 XML 写入 HttpServletResponse (HttpServlet 响应),将 HttpServletResponse 返回至系统前台。

[0086] 4) 根据系统后台返回的数据,局部更新用户页面。

[0087] 当系统前台接收到系统后台返回的 XML 文档后,根据 XML 文档所携带的数据,用 JavaScript 操纵页面的 HTML 文档对象模型 (DOMDocument Object Model),把客户端用户界面的局部内容进行更新。

[0088] 通过上述操作,本发明实施例提供的扩展 RBAC 模型即可以实现局部页面更新操作。从而避免了系统与用户交互过程中,占用过多带宽的情况。而且交互过程中,系统处理的数据量小,因此加快了系统反映时间,也为用户提供更友好的界面。

[0089] 通过上述描述可以看出,本发明实施例提供的扩展 RBAC 模型,可以实现系统权限集合动态适应系统资源的调整变化,使扩展 RBAC 模型具有很强的灵活性。而且还可以实现用户的特别授权,方便了用户使用。另外,本发明实施例提供的扩展 RBAC 模型还可以实现

浏览器和服务端双重验证,保证了权限控制的安全性和正确性。本发明实施例提供的扩展 RBAC 模型还可以实现局部页面更新操作。从而避免了系统与用户交互过程中,占用过多带宽的情况。而且交互过程中,系统处理的数据量小,因此加快了系统反映时间,也为用户提供更友好的界面。由于本发明实施例提供的扩展 RBAC 模型并没有对 RBAC 模型进行架构上的改变,因此具有很强的通用性。

[0090] 基于本发明实施例提供的扩展 RBAC,本发明实施例提供一种用户权限管理方法,如附图 4 所示,包括:

[0091] 步骤 41,系统在系统资源发生变化时,更新独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;

[0092] 步骤 42,系统确定用户访问更新后的所述权限集合的访问权限;

[0093] 步骤 43,系统基于确定的所述用户的访问权限,对用户发送的访问请求进行权限管理控制。

[0094] 下面结合附图,对本发明提供的用户权限管理方法的一个具体实施例的实现过程进行详细的描述。

[0095] 需要说明的是,本发明实施例提供的用户权限管理方法在实现之初,需要对基本元素进行定义。

[0096] 定义配置部分具体可以包括:

[0097] 1) 对象定义:对象是指系统中各种功能模块、界面元素(包括菜单、按钮等各种界面上能控制的控件)等,它们是用用户能访问的各种对象。系统也只能控制这些事先已定义好的对象,因此,对象定义是整个系统的核心步骤直接影响后面的各个安全控制环节。对象定义的包括如下几步:

[0098] 1、功能模块定义:系统中除部分公用的界面、公用功能模块外,其它均为业务功能模块,是用户完成各自不同的业务功能的主要操作途径。有定义的功能模块对象就有可能组织权限根据用户需要完成的工作配置用户业务功能菜单,这也符合“最少特权策略”。

[0099] 2、界面元素控制:除了功能菜单要受到控制外,如要控制功能模块的界面元素其功能模块界面元素也需定义,大部分界面元素均包含有相关的业务功能操作,所以对相应操作的界面元素是进行定义是有必要的。

[0100] 2) 权限定义:在定义好对象的前提下,定义对象在不同情况的访问类型,希望对象在不同情况下具有不同的访问类型,这就需要定义对象的权限。定义权限就是定义对象访问控制和数据访问控制。为了表述方便我们对权限用一个三元组符号来表示 $P(o, t, p)$,其中 o 表示访问对象; t 表示访问类型; p 表示谓词。表示在谓词 p 为真时对于对象 o 可进行 t 类型的访问。权限定义系统安全管理基础步骤之一,只有给各种对象定义好访问的权限,才能给角色配置权限,基于角色管理才能成为可能。

[0101] 3) 角色定义:基于角色的访问控制方法的思想就是把对用户的授权分成两部份,用角色来充当用户行使权限的中介。这样,用户与角色之间以及角色与权限之间就形成了两个多对多的关系。系统提供角色定义工具允许用户根据自己的需要(职权、职位以及分担的权利和责任)定义相应的角色。角色之间有相应继承的关系,当一个角色 $r1$ 继承另一个角色 $r2$ 时, $r1$ 就自动拥有了 $r2$ 的访问权限(表示 $r1 \rightarrow r2$)。角色继承关系自然的反映了一个组织内部权利和责任的关系,为方便权限管理提供了帮助。

[0102] 4) 用户定义:系统的最终使用者是用户,因此必须建立用户的鉴别机构,登记用户的身份信息。在系统中定义可登录的用户操作系统是系统安全管理所必须步骤,也是人与系统的接口。

[0103] 另外,在本发明实施例实施之前,还可以存在用户权限的生成过程,具体可如附图 5 所示,包括:

[0104] 步骤 51,生成系统权限集合。

[0105] 首先建立系统的资源集合 Sf,即明确系统中的哪些页面(甚至页面元素)和模块作为系统资源参与权限控制,为其指定唯一的编码 Fcode 和其他必要属性。然后利用资源集合 Sf 与权限类别集合 Spt 进行关联,得到一个对照表 (Sf-Spt),即系统权限集合 Sp。

[0106] 步骤 52,为角色分配权限。

[0107] 首先根据系统要求,按照部门或级别等划分原则,确定系统的角色集合 Sr。然后针对角色集合 Sr 中的每个元素 Ri,在权限集合 Sp 中为其寻找应有的权限子集 P' (P₁, P₂, …… P_n),建立角色集合与权限集合之间的对应关系,形成角色一权限对照表 (Sr-Sp)。

[0108] 步骤 53,为用户指定角色。

[0109] 新建用户,加入系统用户集合 Su 中,然后在角色集合 Sr 中根据需要为其寻找应有的角色子集 R' (R₁, R₂, …… R_n),建立用户与角色集合之间的对应关系,形成用户一角色对照表 (Su-Sr)。

[0110] 步骤 54,为用户指定会话进程。

[0111] 由系统为用户指定其对应的会话进程 Ss,建立会话一用户对照表 (Ss-Su),其中每个会话都存有登录用户的相应信息和会话的上下文环境。

[0112] 如附图 6 所示,本发明实施例的实现过程具体可以包括:

[0113] 步骤 61,在系统资源发生变化时,更新权限集合。

[0114] 本发明实施例中,由于权限类别集合与资源集合各自独立设置,并且两者结合构成系统的权限集合,因此,当系统新增资源时,则系统可以将新增加的资源添加至资源集合中,更新独立设置的资源集合中的资源。并且,系统还可以将独立设置的权限类别集合中,与新增加的资源对应的操作权限类别,如增加、删除、修改等操作加载至该资源上,即增加对该资源的增加、删除、修改等操作权限。由于资源集合是独立于权限类别集合设置的,因此,资源集合的变动,并不会影响权限类别集合,这样,由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合,即利用更新后的资源集合,更新权限集合。

[0115] 由于权限类别集合中的操作权限类别存在唯一的编码,所以可根据资源的前缀编码来判断新增加的资源需要加载哪些权限类别集合中的操作权限类别。比如权限类别集合中的增加、删除、修改的操作权限类别分别对应 1、2、3 的编码,而系统新增加的资源的前缀编码为 1、2、3,则系统为新增加的资源加载增加、删除、修改操作权限类别。

[0116] 而在系统资源减少时,系统将减少的资源从资源集合中删除,更新独立设置的资源集合中的资源,然后由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合。

[0117] 在本发明的另一个具体实施例中,如果系统资源并没有发生变化,则无需执行步骤 61。

[0118] 步骤 62,确认用户身份。

[0119] 具体的,系统可以通过会话进程与用户交互,即可以根据已经建立的会话—用户对照表 (Ss-Su),确定用户身份。

[0120] 步骤 63,确认用户角色。

[0121] 具体的,系统可以根据已经确定的用户身份,并根据事先建立的该用户身份与用户所属角色之间的对应关系,即可以通过查找用户—角色对照表 (Su-Sr),确定该用户身份对应的用户所属角色。

[0122] 步骤 64,确认用户访问权限。

[0123] 具体的,系统可以根据已经确定的用户所属角色,并根据事先建立的该用户所属角色身份与权限集合(可以为更新后的权限集合,也可以为没有发生变化的权限集合)之间的对应关系,即可以通过查找角色—权限对照表 (Sr-Sp),确定该用户所属角色访问权限集合的访问权限。

[0124] 如果该用户所属角色具有访问权限集合的访问权限,则后续执行步骤 68,如果改用户所属角色不具有访问权限集合的访问权限,则后续可以执行步骤 65。

[0125] 步骤 65,提示用户。

[0126] 具体的,系统可以将用户所属角色不具有访问所述权限集合的访问权限的确定结果反馈至用户。

[0127] 步骤 66,用户发送特别授权请求。

[0128] 具体的,在系统提示用户其所属角色不具有访问权限集合的访问权限情况下,如果用户需要访问其所属角色不具有访问权限的权限集合,则用户向系统发送特别授权请求,希望系统授予其访问该权限集合的特别访问权限。

[0129] 在本发明的另一个具体实施例中,如果用户事先已经知道其自身所属角色不具有访问某个预定权限集合时,则用户可以无需系统提示,自身主动向系统发送特别授权请求。

[0130] 步骤 67,系统为用户特别授权。

[0131] 系统在接收到用户发送的特别授权请求时,可以为该用户授予不基于用户所属角色访问权限集合的特别访问权限。即用户可以不基于用户所属角色,向权限集合发送访问请求。

[0132] 另外,本发明实施例中也可以在用户完成访问操作或者根据需要,撤销为用户授予的特别访问权限。

[0133] 步骤 68,系统基于确定的用户访问权限,对用户发送的访问请求进行权限管理控制。

[0134] 具体的,系统可以基于步骤 64 确定的用户访问权限,也可以基于步骤 67 中授权用户的特别访问权限,对用户发送的访问请求进行权限管理控制。

[0135] 基于本发明实施例提供的扩展 RBAC,本发明实施例还提供一种用户权限管理装置 70,如附图 7 所示,包括

[0136] 更新模块 71,用于在系统资源发生变化时,更新独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;

[0137] 确定模块 72,用于确定用户访问更新模块 71 更新后的权限集合的访问权限;

[0138] 权限管理控制模块 73,用于基于确定模块 72 确定的用户访问权限,对用户发送的访问请求进行权限管理控制。

[0139] 本发明实施例中,由于权限类别集合与资源集合各自独立设置,并且两者结合构成系统的权限集合,因此,当系统新增资源时,则更新模块 71 可以将新增加的资源添加至资源集合中,更新独立设置的资源集合中的资源。并且,更新模块 71 还可以将独立设置的权限类别集合中,与新增加的资源对应的操作权限类别,如增加、删除、修改等操作加载至该资源上,即增加对该资源的增加、删除、修改等操作权限。由于资源集合是独立于权限类别集合设置的,因此,资源集合的变动,并不会影响权限类别集合,这样,由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合,即利用更新后的资源集合,更新权限集合。

[0140] 由于权限类别集合中的操作权限类别存在唯一的编码,所以更新模块 71 可根据资源的前缀编码来判断新增加的资源需要加载哪些权限类别集合中的操作权限类别。比如权限类别集合中的增加、删除、修改的操作权限类别分别对应 1、2、3 的编码,而新增加的资源的前缀编码为 1、2、3,则更新模块 71 为新增加的资源加载增加、删除、修改操作权限类别。

[0141] 而在系统资源减少时,更新模块 71 将减少的资源从资源集合中删除,更新独立设置的资源集合中的资源,然后由更新后的资源集合与权限类别集合组合,构成更新后的系统权限集合。

[0142] 在一个优选的实施例中,本发明实施例所涉及的更新模块 71,还可以包括:

[0143] 加载单元 711,用于在系统资源增加时,根据增加的系统资源对应的权限类别集合编码,将对应的权限类别集合加载至所述增加的系统资源。

[0144] 在一个优选的实施例中,如附图 8 所示,本发明实施例提供的确定模块 72 具体可以包括:

[0145] 身份确定单元 721,用于通过会话进程与用户交互,确定用户身份。

[0146] 具体的,身份确定单元 721 可以通过会话进程与用户交互,即可以根据已经建立的会话—用户对照表 (Ss—Su),确定用户身份。

[0147] 需要说明的是,本发明实施例中所涉及的各种对应关系,以及对照表均可以事先建立。具体建立过程可如上所述,在此不再赘述。

[0148] 角色确定单元 722,用于根据身份确定单元 721 确定的用户身份与用户所属角色之间的对应关系,确定所述用户所属角色。

[0149] 具体的,角色确定单元 722 可以根据已经确定的用户身份,并根据事先建立的该用户身份与用户所属角色之间的对应关系,即可以通过查找用户—角色对照表 (Su—Sr),确定该用户身份对应的用户所属角色。

[0150] 访问权限确定单元 723,用于根据角色确定单元 722 确定的用户所属角色与权限集合之间的对应关系,确定用户所属角色对应的访问权限。

[0151] 具体的,访问权限确定单元 723 可以根据已经确定的用户所属角色,并根据事先建立的该用户所属角色身份与权限集合(可以为更新模块 71 更新后的权限集合,也可以为其他权限集合)之间的对应关系,即可以通过查找角色—权限对照表 (Sr—Sp),确定该用户所属角色访问权限集合的访问权限。

[0152] 在本发明实施例提供的确定模块 72 另一个具体实施例中,确定模块 72 还可以包括:

[0153] 提示单元 724,用于在访问权限确定单元 723 确定用户所属角色不具有访问权限集合的访问权限时,将访问权限确定单元 723 的确定结果反馈至用户。

[0154] 在本发明实施例提供的确定模块 72 另一个具体实施例中,确定模块 72 还可以包括:

[0155] 特别授权单元 725,用于根据接收的用户发送的特别授权请求,为用户授予不基于用户所属角色访问权限集合的特别访问权限。

[0156] 需要说明的是,本发明实施例中,用户发送的特别授权请求可以是在接收到提示单元 724 反馈的信息情况下发送,也可以是用户自身主动发送。

[0157] 另外,特别授权单元 725 也可以在用户完成访问操作或者根据需要,撤销为用户授予的特别访问权限。

[0158] 在一个优选的实施例中,本发明实施例所涉及的权限管理控制模块 73,具体的可以基于访问权限确定单元 723 基于用户所属角色确定的用户访问权限,也可以基于特别授权单元 725 授权用户的特别访问权限,对用户发送的访问请求进行权限管理控制。

[0159] 综上所述,本发明实施例提供的用户权限管理方法及装置,通过在系统资源发生变化时,更新扩展 RBAC 模型中,独立于权限类别集合的资源集合中的资源,并利用更新后的资源集合更新权限集合;确定用户访问更新后的所述权限集合的访问权限;基于确定的所述用户的访问权限,对所述用户发送的访问请求进行权限管理控制。从而使系统的权限集合可以适应系统资源集合的动态调整,而且还可以使系统为用户授权更加灵活。

[0160] 以上所述仅是本发明的实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以作出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

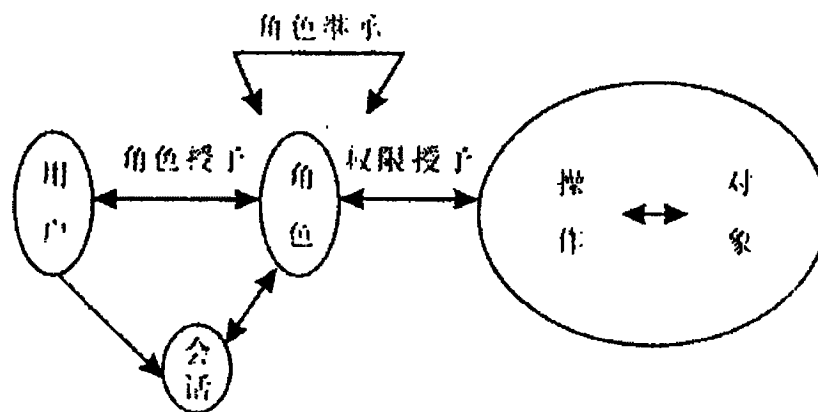


图 1

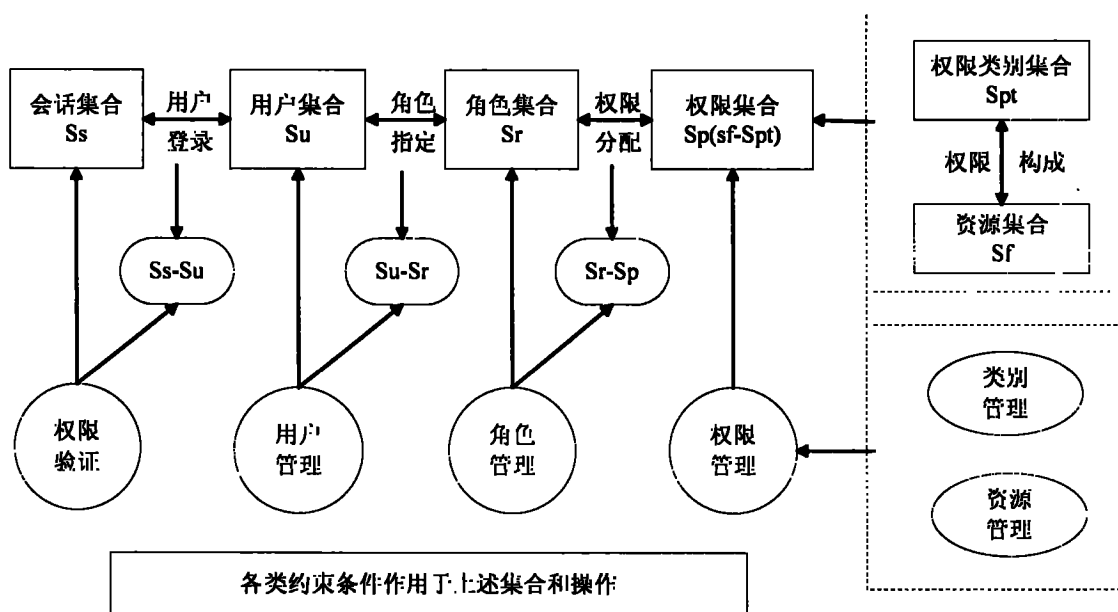


图 2

接口名	callUrl(url, inparas, outparas, isSyn);		
概述	通过 URL 方式调用服务		
返回值	无		
参数	参数	说明	
	url	String	服务地址
	inparas	入参	
	outparas	控件 ID	返回值对应的控件
	isSyn	是否同步	true 为同步, false 或不配置为异步

图 3

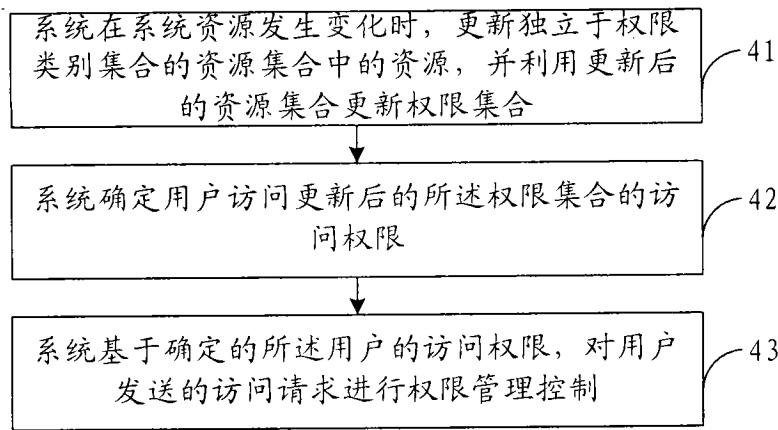


图 4

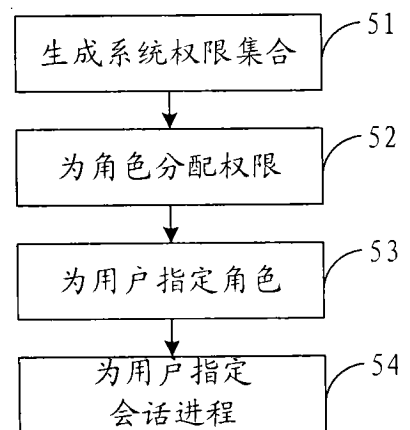


图 5

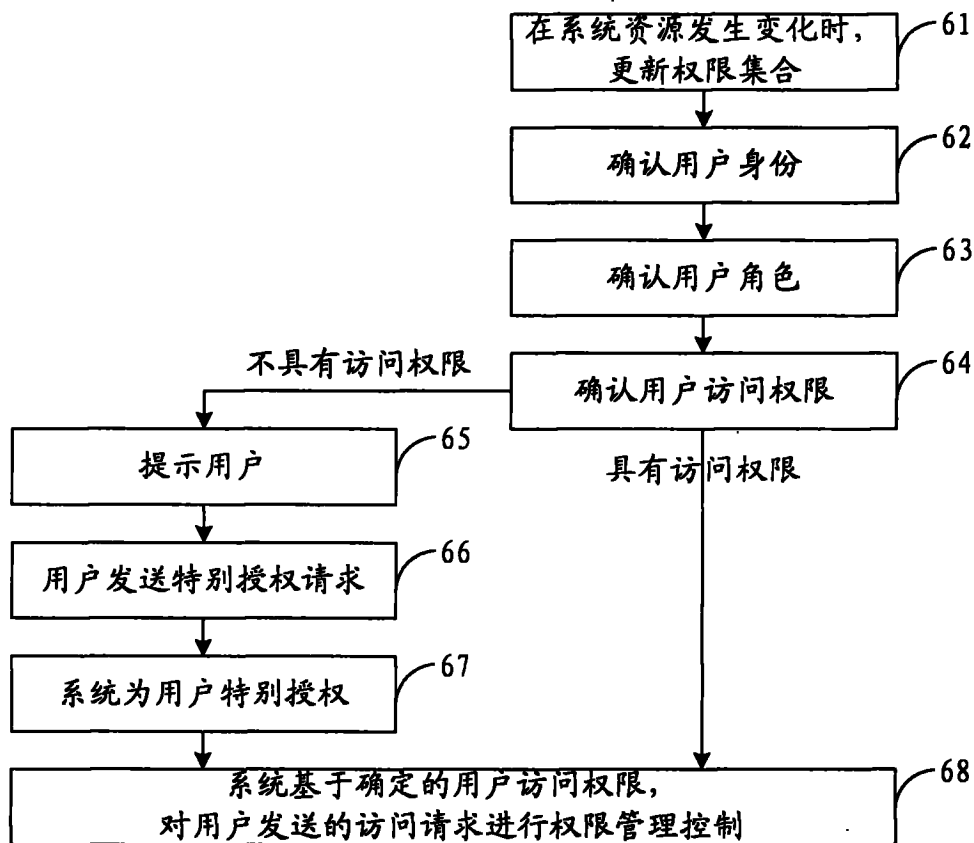


图 6

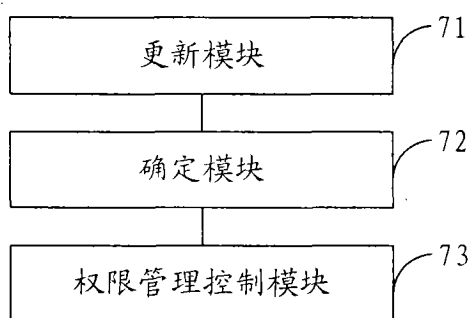


图 7

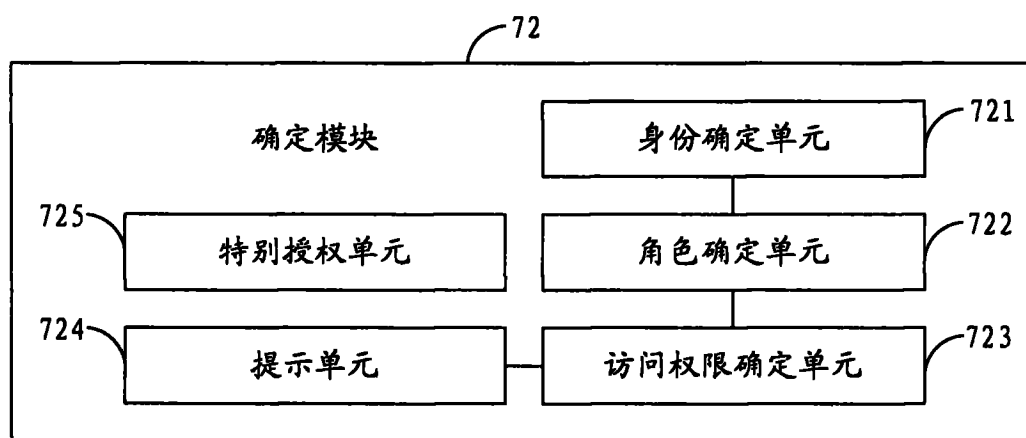


图 8