产业链协同 SaaS 平台多租户权限管理技术①

韩 敏[1,2], 杨 柳[1]

[1](西南交通大学 信息科学与技术学院, 成都 610031)
[2](成都信息工程学院 软件工程学院, 成都 610103)

摘 要: 面向汽车及零部件产业链协同 SaaS 平台的多核网状企业群协同管理需求, 在传统的 RBAC 模型的基础之上, 提出一种以龙头企业为核心的多租户多级授权模型, 运用菜单动态生成技术、URL 地址解析算法实现了权限的解析, 有效地控制了用户的访问权限. 经验证, 提出的模型和实现方法有效解决了产业链协同 SaaS 平台上多租户的用户权限管理问题.

关键词: 产业链协同; SaaS 平台; 多租户; 权限管理; URL

Authority Management of Multi-Tenants for SaaS-Based Collaborative Platform of Industrial-Chain

HAN Min[1,2], YANG Liu[1]

[1](College of Information Science and Technology, Southwest Jiao Tong University, Chengdu 610031, China)
[2](College of Software Engineering, Chengdu University of Information Technology, Chengdu 610103, China)

Abstract: To meet the collaborative management requirement of multi-core network enterprise group on SaaS-based collaborative platform of automobile industry value chain with automobile parts industry value chain, a multi-tenants and multi-level authorization model based on the core of a leading enterprise is proposed on the basis of the traditional RBAC model. The scheme effectively controls users' access rights.and achieves analysis of authority by using menu dynamic generating technology and URL address resolution algorithm. The plan and the implementation are tested on the SaaS-based collaborative platform of industrial-chain. The results show that it can solve the problem of authority management of multi-tenants for SaaS-based Collaborative Platform of industrial-chain.

Key words: cooperation of industrial-chain; SaaS-based platform; multi-tenants; authority management; URL

"十二五"期间, 国家首次设立了"现代服务领域"、"先进制造技术领域"两大领域的科学研究, 并且设立了"制造业信息化科技工程专项计划". 产业链协同 SaaS 平台在此科技浪潮的推动下迅速发展起来了. 该平台推进了服务业和制造业的协同发展、促进了信息化与工业化的深度联合、充分利用网络资源共享特性, 为中小型企业信息化提供了可靠保障. 目前, 平台已经服务于数十家联盟, 近万家企业用户. 如何有效地控制用户权限已经成为制约平台发展的问题之一.

现在绝大多数 SaaS 平台采用的都是基于角色的访问控制(RBAC)模型, 为了达到更好的效果, 国内外学者也不断的对 RBAC 进行改进, 使之能更好的使用于既有系统. 文献[1]建立了面向产业链协同SaaS平台动态联盟的四级授权模型, 对联盟、企业、部门及用户等授权继承关系进行了研究, 提出了权限控制算法, 保证了各级权限的从属关系, 但是权限控制的多次判断增加了系统的开销. 针对文献[1]角色继承引发的问题, 文献[2]禁用了角色的继承关系, 建立动态访问控制模型, 减小了权限分配时系统开销, 但缺少对全局角色和租户私有角色权限的研究, 会产生冲突. 文献[3]分析了 SaaS 平台多租户、多角色的特点, 提出了一种拓展角色的访问控制模型, 在原有 RBAC 的基础上增加了租户和租户的角色, 提高了访问控制的可管理性, 但是并没考虑到租户角色动态变化的特点. 鉴于

① 基金项目:国家科技支撑计划项目(2015BAF32B05);四川省科技支撑计划(2015GZ0076)
收稿时间:2015-04-10;收到修改稿时间:2015-05-18

此, 文献[4]针对平台多用户及权限动态变化的特点, 增加了临时角色满足用户权限动态变化的需求,同时按照树形结构方式引入用户组策略实现了部门的整体授权, 但该模型只是简单的把权限和资源进行了简单的分组, 权限控制的实质内容没变. 文献[5]和文献[6]分别针对 SaaS 模式提出并设计了一种基于 RBAC 的访问控制模型, 并通过相关系统实例说明了模型的合理性及便利性, 但是, 不适用于产业链协同 SaaS 平台.

本文在以上研究成果的基础上, 针对一平台多租户、一租户多用户以及业务数据高安全性需求的特点, 提出了一种面向产业链协同 SaaS 平台的租户权限管理方案, 并应用到汽车及零部件产业链协同 SaaS 平台上, 验证表明, 能满足平台不同租户的用户权限管理需求.

1　权限管理模型概述

1.1 产业链协作 SaaS 平台用户特点分析

产业链企业间的协作是以某个产品的供需链为基础的. 随着业务数量、业务类型、业务关系的增加, 平台上租户逐步向网状化、多核化发展, 以龙头企业为核心的协作体系关系如图 1 所示, 其特点如下:

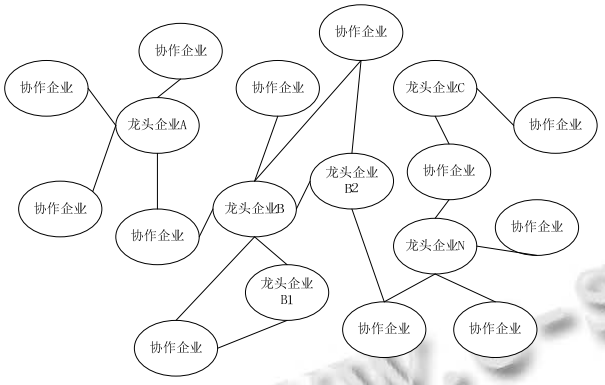

图 1　以龙头企业为核心的协作体系关系图

1)平台上存在多个租户, 各租户经营相互独立.

2)租户是围绕特定业务, 由一个龙头企业, 多个协作企业用户共同形成的协作体系, 在该体系中, 有且只有一个龙头企业.

3)由于在业务协同中承担的角色不同, 同一企业可能既是某租户中的协作企业, 也有可能是另一租户的龙头企业. 如图 1 中龙头企业 B1, 龙头企业 B2 都是龙头企业 B 的协作企业, 在其上下游有着众多二级协作企业;

4)由于企业关系多样化, 同一个协作企业用户有可能在在于多个租户中, 有些协作企业同时服务与两个一级龙头企业, 有些协作企业既服务于一级龙头企业又服务于二级龙头企业.

产业链协同 SaaS 平台上的租户有着众多分工不同的角色共同使用平台, 来完成业务协作. 平台则由平台运营商统一管理, 租户则根据角色不同, 负责各自的业务, 具体角色包括:

1)平台管理员: 负责平台的维护和管理, 包括租户账号管理、权限管理等. 平台管理员不负责具体业务的管理. 平台涉及众多租户信息的管理, 平台管理员有时也需要进行角色的划分.

2)租户管理员: 租户管理员又可细分为龙头企业管理员和协作企业管理员, 他们的主要职责是负责企业内部操作员账号管理及权限管理, 不会涉及具体业务的操作. 另外, 龙头企业管理员还需要负责协作企业账户的管理.

3)租户操作员: 租户操作员用户也可细分为龙头企业操作员用户和协作企业操作员用户, 他们的主要职责针对业务需求完成相应的业务协同.

平台上具体用户分类如图 2 所示.

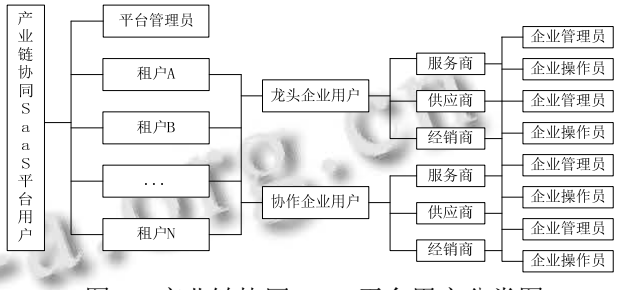

图 2　产业链协同 SaaS 平台用户分类图

1.2 以龙头企业为核心的租户管理模型

产业链协同 SaaS 平台上企业间的业务协同, 是建立在以龙头企业为核心的上下游企业间业务协作关系的基础之上的[7]. 企业间由业务往来而相互关联着, 这样的企业群称之为租户. 租户中, 不同企业用户的安全性需求是以企业间的协作关系以及用户自身业务需求为基础的. 为此, 本文引入定义 1.

定义 1. 针对多租户产业链协同 SaaS 平台租户关系的多核性、网状性特点, 以龙头企业为核心的租户管理模型 U_i 可以用一个四元组表示, $U_i=\{E_x, T_{Ui}, R(T_{Ui}), E\rightarrow R(T_{Ui})\}$表示, 其中:

① E_x={ E_0, E_1, …, E_n}表示租户包含的所有企业, n 为租户中企业的数量, E_0 表示龙头企业, 其它表示协作企业;

②T_{Ui}={ T_0, T_1, …, T_n}表示龙头企业与协作企业的协作关系类别, 例如经销商、服务商等;

③R(T_{Ui})表示按协作关系类型对企业划分的片区级, 租户可以针对各个片区的特点对片区灵活管理;

④E→R(T_{Ui})表示在某片区内租户 U_i 的协作关系集.

一个龙头企业的业务可能在多个片区同时展开, 不同片区在业务上会有一定的差异. 同一个片区的协作企业和龙头企业之间有多种协作关系.

租户使用平台, 就是向平台租用服务的过程, 平台所提供的服务用 Service 表示, Service ={ $Service_1$, $Service_2$, …, $Service_n$}. 租户根据自身业务需求向平台运营商租用服务, 并对其进行个性化配置. 再根据用户业务需求, 为其分配相应的业务功能模块. 用户的安全需求与其业务功能形成一种关联关系. 为了便于描述, 本文引入定义 2.

定义 2. 服务 Servicek 包含多项业务功能, $Service_k$ 中的业务可用五元组 Function={Function_id, Function_name, Function_des, Function→R($Service_k$), Permissions}表示, 其中:

①Function_id 表示业务功能标识, 即平台上的大类功能标识;

②Function_name 表示业务功能名称;

③Function_des 表示业务功能描述;

④Function→R(Servicek)表示业务功能服务的片区;

⑤Permissions 表示该业务功能包含的小类功能集, 即实际操作权限.

1.3 产业链协同 SaaS 平台权限管理模型

当用户登录平台后, 需要有良好的权限管理机制, 保证用户的合法访问. 目前, 应用最为广泛的访问控制模型是 RBAC 模型[8,9]. 本文在 RBAC 的基础上结合产业链协同 SaaS 平台多租户多用户多角色特点, 提出了 Smt-RBAC(SaaS multi-tena ncy Role-Based Access Control)模型. 模型的具体描述如图 3.

Smt-RBAC 模型中的元素, 及元素之间的关系如图 3 所示. 该模型中, 一个 Tenant 可拥有一个或者多个 Services, 包含多家协作企业. 对于每个服务而言, 租户可以拥有其所有的操作权限, 也可能只拥有服务中部分功能的操作权限. 每个用户可拥有多个角色, 一个角色拥有多个权限, 角色隶属于某个企业. 一个用户对应一个菜单. 平台管理员负责各租户权限和全局角色的分配但不具有实际操作能力[9]. 为了便于描述, 我们将权限定义如下.

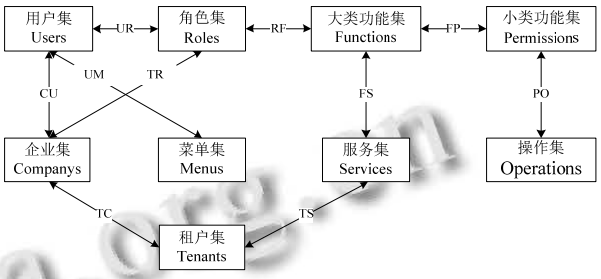

图 3 Smt-RBAC 模型

定义 3. 权限可用一个六元组 Permission={Permission_id, Permission _name, Permission_des, Permission_Operation, Service_id, Function_id}表示, 其中:

①Permission_id 表示权限标识;

②Permission_name 表示权限名称;

③Permission_des 表示链接地址, 即链接到网页的 URL 地址;

④Permission_Operation 表示该权限的操作集, 如查看、修改等;

⑤Service_id 表示该权限所属服务的标识;

⑥Function_id 表示该权限所属大类功能的标识.

在平台上, 为了便于租户的理解, 权限也称之为小类功能, 表示允许租户内用户对平台资源进行相关操作. 平台中一个功能大类包含多个小类功能. 由产业链协同 SaaS 平台“单实例多租户”的特点分析得知, 平台对租户的授权实际上就是平台上的功能权限向租户建立映射关系的过程. 对于平台上任意租户 U_i, 在协作过程中平台的授权情况可由如下定义表示.

定义 4. 租户 Ui 在平台所拥有的权限集可以由 $Permission_{Ui}$= {$Permission_{m1}$, $Permission_{m2}$, $Permission_{n2}$, $Permission_{n3}$,…}表示, $Permission_{Ui}$ 是一个有限的集合. 平台上租户可以拥有多种服务的多种功能, 具体授予的权限根据租户需求而定, 平台管理员根据租户需求完成租户授权.

在联盟中, 企业是按照协作类别管理的, 同一协作类别的企业与龙头企业交互的业务内容基本相同.

由此可见, 在同一联盟内, 同一协作类别的企业的操作权限相同[1]. 引入定义 5 来描述具有相同协作关系企业的权限集合.

定义 5. 租户 U_i 的协作类别为 T_{Ui} 的协作企业所拥有的权限集合可由 $Permission_{Ui}(T_k)$={$Permission_{m1}$, $Permission_{m2}$,…,$Permission_{n2}$,…}表示, 其中 $Permission_{Ui}(T_k) \subset Permission_{Ui}$. 协作企业的授权由龙头企业管理员完成.

由定义 4 和定义 5, 企业权限 $Permission_{Ui}(T_k)$和协作类别 T_{Ui} 之间有着一定的继承性, 一旦协作关系确定, 那么该协作体系企业的权限集合也就确定了. SaaS 平台上, 平台、租户、企业的授权模型如图 4.

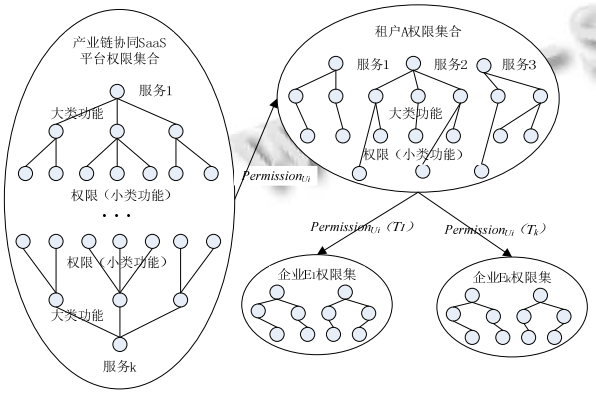

图 4　平台、租户和企业的授权模型

在企业内部, 权限按照部门职责还需要进行进一步细分, 部门用户的权限是部门权限的子集. 对于同一部门的不同操作员用户, 他们的权限可能不同, 也可能相同. 部门操作员的权限由部门管理员根据其工作岗位的岗位需求和业务要求设定. 对于企业中的部门权限, 我们对其进行如下描述.

定义 6. 在平台上, 企业中的部门权限可用 $Permission_{Ei}(Dep_t)$= {$Permission_{m1}$, $Permission_{m2}$,…}表示. 其中 $Permission_{Ei}(Dep_t) \subset Permission_{Ui}(T_k)$. 企业中部门的权限是有企业管理员设定的.

在 Smt-RBAC 模型中, 用户对平台的操作权限是由角色继承而来的. 在平台上, 企业管理员和部门主管均可以设定角色, 该角色为部门内部角色, 本文通过设置部门角色来解决水平权限管理问题. 为了便于描述, 本文对角色进行如下定义.

定义 7. 角色 Role 可用一个五元组 Role 表示, Role={Role_id, Role_name, Company_id, Dep_id, Permission_collection}, 其中:

①Role_id 表示角色标识;

②Role_name 表示角色名称, 为了解决角色命名冲突问题, 本文的角色名称由企业名称、部门名称、当前角色名称共同组成, 即 Role_name= Company_name ∪ Dep_name ∪ PosRole_name, 如五征综合部业务员;

③Company_id 表示该角色所在企业的标识, 产业链协同 SaaS 平台上存在多家企业, 各企业有着自己的内部角色, 企业标识可区分各企业同名角色;

④Dep_id 表示该角色所在部门的标识;

⑤Permission_collection 表示该角色所拥有的权限集. 对于 Dep_t 的内部角色, Permission_collection $Permission_{Ei}(Dep_t)$.

平台上的业务由操作员用户操作, 而操作员的权限由角色-权限的映射而来. 为了便于描述, 本文对用户做如下定义.

定义 8. 在面向多租户的产业链协同 SaaS 平台中, 用户 User 可用一个六元组 User={User_id, User_name, User_Role_id, User_info, User_POL, Company_id, Department_id}表示, 其中.

①User_id 表示用户标识;

②User_name 表示用户名;

③User_Role_id 表示该用户所分配的角色;

④User_info 表示用户的基本信息;

⑤User_POL 表示用户的操作等级, 若 User_POL=1 则表示该用户只有查看的操作权限.

在企业内部, 对具体用户的授权是基于角色的授权而建立的. 部门内部角色的权限是部门权限的子集, 一个用户可以被指派多个角色, 但该用户的权限也只能是部门权限的子集, 企业内部部门与角色、用户之间的授权模型如图 5 所示.

在普通的 RBAC 模型下, 系统只会验证用户是否属于角色, 但是无法进一步判断用户可以访问哪些数据及对该页面有什么操作功能.同时为了解决水平权限管理问题, 本文添加了 User_POL 字段, 建立了基于用户操作级别的页面元素控制算法, 根据 User_ POL, 在页面初始化之前对页面元素及页面数据进行控制. 权限解析如图 6 所示.

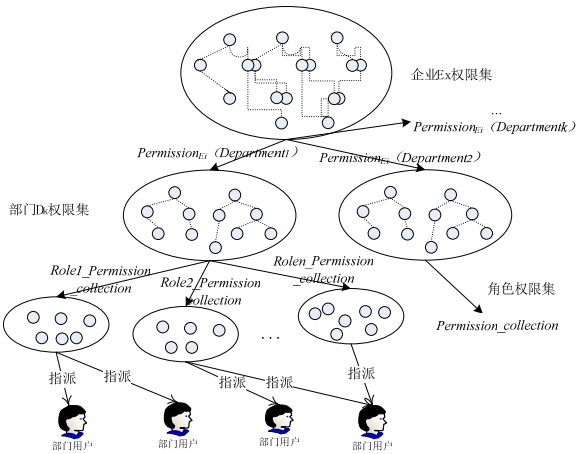

图 5 企业内部部门与角色、用户的授权模型

图 6 权限解析过程图

2 租户权限的分配及解析验证设计过程

2.1 权限的授予及修改

租户 U_i 中各类业务协同的过程就是协作企业操作员和龙头企业操作员使用各自权限进行相关业务交互、信息交互的过程. Smt-RBAC 模型选择基于角色的授权模式. 模型中的授权即为角色的授权, 授权模式是验证系统运行的基础和必备功能[10]. 其授权方式可描述如下:

Step1: 根据 $User_y$_info 获取用户所属部门 Dep_t_id;

Step2: 新建角色 $Role_m$, 加载部门权限集合 $Permission_{Ei}(Dep_t)$, 初始化部门权限到页面中, 并将其值置为空;

Step3: 为该角色分配权限 Permission_collection, 并且 Permission_collection $\subset Permission_{Ei}(Dep_t)$, 保存 Rolem;

用户权限的授予是靠角色的权限映射而来, 角色的创建、保存及权限的分配, 是本模型中的一个实时处理过程. 随着企业业务的多元化发展及业务规模的变化, 角色的权限集也是随时变化了, 为了便于角色的管理, 对角色权限集的修改的过程描述如下:

Step1: 获取角色所在部门的功能大类标识集 Function_id_collection 和权限集 $Permission_{Ei}(Dep_t)$, 并将其绑定在页面的 checkboxlist 控件上;

Step2: 获取原来角色的 Permission_collection;

Step3: 展示角色原有的权限, 将对应 checkboxlist 选项置为已选标识;

Step4: 修改角色权限: 若要新增 $Permission_x$, 则直接选择页面的该 $Permission_x$ 对应的 checkboxlist 控件, 若要去除某个 $Permission_x$, 则将该权限对应的 $checkboxlis_t$ 控件置为 False;

Step5: 保存角色, 修改数据库中 Permission_collection.

2.2 权限的解析验证

权限的解析验证是用户访问权限的控制中枢. 平台通过动态菜单生成技术来限定用户的访问路径, 使用 URL 地址解析算法来对实现对系统资源的合法访问, 采用页面控件控制算法来防止用户的非法操作. 用户成功登录平台后, 系统对权限的解析验证的实现过程描述如下:

Step1: 判断用户是否登录平台, 若登录, 加载用户权限集 User(Permission _collection);

Step2: 新建一个 StringBuilder strMenu, 将大类功能名称和小类功能名称写入到 strMenu 中, 最终显示在左侧菜单栏中;

Step3: 当用户点击菜单栏中点击或直接在浏览器地址栏中的写入 URL 地址时, 调用 URL 地址解析算法, 判断是否为用户合法权限, 若不是合法权限, 则提示错误, 返回;

Step4: 加载页面前, 首先获取用户操作级别 UserLevel=GetLevelPowerValue(UserName), 当 UserLevel 大于或等于当前控件要求时, 可做相关处理, 否则 Control.Enabled= False, Control.Visible=False.

3 方案验证

本文所建立的 Smt-RBAC 模型已经在汽车及工程机械产业链协同商务平台的设计中得到了应用. 该平台支持多租户的访问, 该模型的应用实现了对用户权限的灵活管理. 解决了以龙头企业为核心的上下游协作企业之间的业务协同, 使得业务协作更加便捷.

系统的设计采用.NET FrameWork, 开发语言为 C#, 方案的具体实现如下.

3.1 权限的授予

授权包括平台给联盟授权、联盟给企业授权、企业给部门授权、部门给用户授权. 以平台上采购系统为例, 企业给部门授权界面如图 7 所示.

当企业添加新部门时, 企业管理员只需要添加部门名称和权限集, 部门编号由系统自动生成. 部门权限集是企业权限集的子集, 部门主管给部门角色分配的权限只能是部门权限的子集, 部门角色的授权界面如图 8 所示.

图 7 部门授权界面

图 8 部门角色授权界面

界面中的权限是根据部门权限动态生成的. 在具体实现中, 用 CheckBox 控件绑定功能大类名称, CheckBoxList 控件绑定功能小类名称. 然后由两个函数完成数据相应数据的绑定: 绑定大类功能函数 DataBind_FunctionItem() 和绑定小类功能函数 DataBind_PermissionsItem().

一般情况下, 部门主管只能管理本部门内部局部角色. 为了减少角色命名冲突问题, 本系统的角色名为企业名称+部门名称+角色名称的命名方式存储在数据库中, 但这种命名方式不会在界面中展现出来. 部门角色管理的界面如图 9 所示.

图 9 部门角色管理图

3.2 权限的验证

本文涉及权限的解析主要包括以下三个方面: 菜单栏显示控制、URL 地址解析和用户操作权限解析. 通过以上功能来解析权限, 可实现对用户访问的控制.

1)菜单栏显示控制

系统的菜单是根据用户的权限动态生成的. 当用户完成身份认证后, 系统根据用户编号读取数据库中该用户权限集合, 再通过菜单的形式显示出来. 系统实现了用户与菜单的动态绑定, 原则上来说用户只能通过菜单来访问指定的页面. 另外, 企业管理员可根据自己的喜好配置菜单顺序的显示, 在数据库中, 用 MenuOrder 字段来存放菜单的显示顺序.

2)URL 地址解析

由于平台是以 B/S 模式的方式运行在网络中, 用户可以通过在浏览器中直接输入 URL 地址的方式来对访问平台的资源进行非法访问. 为了避免合法用户的越权访问, 本文设计了 URL 地址解析算法对用户的访问做了进一步的限制. 为了减少代码冗余, 本系统设计了一个类库 BasePage, 所有页面都继承 BasePage 中的 PageUI 类, 在 PageUI 类添加权限判断函数, 每次加载页面前需要对用户访问权限进行判断.

3)用户操作权限解析

在权限验证过程中添加了 URL 地址验证, 可在一定程度上的减少系统内部用户的越权访问. 但是, 部分用户也可以通过暴力手段屏蔽 URL 地址解析功能, 为此, 系统设计用户操作权限解析功能, 对页面每一个功能控件进行监听解析, 防止用户对系统资源的非法篡改. 具体做法是, 为每个用户添加操作级别字段 OperationsLevel, 只有当 OperationsLevel 达到一定级别时, 才能对当前页面进行相应的操作, 否则操作按钮不会在页面中显示, 不同 OperationsLevel 的用户进入同一功能页面的显示结果如图 10 和图 11 所示.

图 10 用户 a 的操作权限图

图 11 用户 b 的操作权限图

4 结语

本文通过对国内权限管理技术的研究现状进行分析研究, 针对平台一平台多租户、一租户多用户的特点, 提出了面向多租户的 Smt-RBAC 模型, 该模型实现了角色的分层控制, 提高了平台租户权限管理的灵活性. 上述应用实例验证了该方案的可行性. 随着产业链协同 SaaS 平台的持续发展, 大量数据在收集、存储和使用的过程中面临着诸多安全风险. 本论文的模型只研究了用户的登录和访问安全, 对用户数据的存储安全及传输安全欠缺考虑. 以后的工作中还需要对数据的安全性、完整性进行进一步的研究和探讨.

参考文献

1 王淑营.面向产业链协同商务平台的权限控制模型研究.计算机应用研究,2010,27(1):170–173.

2 Jing X, Tang JL, He DJ. Research and implementtation on access control of management-type SaaS. 2nd International Conference on Information Engineering and Computer Science. Wuhan, China. 2010. 388–392.

3 朱养鹏,张璟.SaaS 平台访问控制研究.计算机工程与应用,2011,47(24):12–16.

4 邓广胜,周国祥.SaaS 管理平台的角色访问控制研究.合肥工业大学学报(自然科学版),2013,36(12):1468–1471.

5 马立林,李红.基于 RBAC 的 SaaS 系统的权限模型.计算机应用与软件,2010,27(4):42–44.

6 申利民,刘波,邢昌元,等.SaaS 模式下可插拔访问控制框架的设计.小型微型计算机系统,2010,31(6):1107–1111.

7 王宇,王淑营.面向产业链协同 SaaS 平台的 DaaS 技术研究.计算机工程与设计,2014,35(3):1081–1087.

8 Sun Software White Papers: RBAC in the Solaris Operating Environment. 2001. 67–69.

9 金诗剑,蔡鸿明,姜丽红.面向服务的多租户访问控制模型研究.计算机应用研究,2013,30(7): 2136–2139.

10 宋万里,吴炜峰.基于改进的 RBAC 模型的系统用户权限控制研究.计算机与现代化,2014,(9):49–54.