工作流系统中一个基于双权角色的条件化RBAC访问控制模型*)

张 健 孙吉贵 李妮娅 胡成全

(吉林大学计算机科学与技术学院 长春130012) (符号计算与知识工程教育部重点实验室 长春130012)

摘 要 传统的RBAC访问控制模型已经不能表达复杂的工作流安全访问控制约束。基于传统的RBAC模型,提出了一个新的基于双权角色的条件化RBAC访问控制模型CRDWR(conditioned RBAC based on double-weighted roles)。阐述了基于动态角色分配的条件化RBAC策略,定义了基于双权角色的工作流系统访问授权新概念,并针对多个角色协同执行任务的序约束问题,给出了基于令牌的序约束算法。该模型能够表达复杂的工作流安全访问控制约束。

关键词 工作流,条件化RBAC,双权角色,令牌,访问控制

A Conditioned RBAC Model Based on Double-weighted Role of Workflow System

ZHANG Jian SUN Ji-Gui LI Ni-Ya HU Cheng-Quan

(College of Computer Science and Technology, Jilin University, Changchun 130012)

(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012)

Abstract The traditional RBAC model cannot express complicated workflow secure access control constraint. Based on the traditional RBAC model, a new conditioned RBAC model CRDWR (conditioned RBAC based on double-weighted roles) is proposed on the basis of double-weighted roles. A conditioned RBAC strategy is discussed on the basis of dynamic role assignment and a new concept of workflow access authorization is defined on the basis of double-weighted role. A sort algorithm based on token is presented in allusion to the problem of multi-role sequence constraint in the process of executing tasks. The model can express complicated workflow secure access control constraint.

Keywords Workflow, Conditioned RBAC, Double-weighted role, Token, Access control

1 引言

工作流管理系统(workflow management system, WfMS[1])着眼于流程信息的驱动和管理,通过工作流引擎的驱动使业务流程中的各个具有一定关系的业务活动在一定程度上自动运行,从而高效、灵活地达到企事业单位的业务需求,是复杂多任务协同建模的一种有效方法。为了保证任务执行的安全性,一个安全的访问控制模型是工作流管理系统中必不可少的重要组成部分。

基于角色的访问控制(RBAC[2])是由美国国家标准化和技术委员会(National Institute of Standards and Technology,NIST)的Ferraiolo等人在20世纪90年代初提出来的。在该方法中引入了角色这个中介,管理员根据应用需求定义各种角色,并给角色分配合适的访问权限,而用户根据其职务和岗位等信息再被指派为指定的角色列表。但是,传统的RBAC访问控制模型已经不能表达复杂的工作流安全访问控制约束,对角色的静态分配无法适应复杂的用户需求,特别是无法表达角色执行任务的事务完整性约束和多个角色协同执行任务的序约束,这是工作流安全研究领域尚未得到较好解决的一个重要问题。文[3]提出了一种基于带权角色的访问控制模型,这在一定程度上解决了角色执行任务的事务完整性约束,但是,文中对角色的动态分配没有涉及,对多个角色协同执行任务的序约束只给出了定义而没有给出具体的解决方案。因此,为了解决这两个问题,本文以传统的RBAC模型和一个简化的公文处理工作流为背景,以文[3]所提出的带权角色访问控制模型为基础,提出了一个工作流系统中基于双权角色的条件化RBAC访问控制模型(conditioned RBAC based on double-weighted roles,CRDWR)。

CRDWR模型通过引入工作流模型的全局变量和任务节点的输入输出变量,实现了角色的动态分配;通过引入关于角色的激活次数权值和激活序权值,实现了角色执行任务的事务完整性约束和多个角色协同执行任务的序约束。

2 RBAC模型和问题背景

在讨论CRDWR模型之前,先介绍一下传统的RBAC模型的一些基本知识。

传统的RBAC模型包括六个基本的元素集合[2]:用户集合(USERS),角色集合(ROLES),任务集合(TASKS),操作集合(OPS),权限集合(PRMS)和会话集合(SESSIONS)。角色是一个组织概念,它可以表示职务、岗位等信息;任务分配给角色而不分配给用户,一个用户如果获得了执行任务所需要的角色,则其具有执行相应任务的资格;只有当会话激活任务和用户获得了执行该任务所需要的角色时,该用户才能使用该角色包含的权限执行该任务。该模型如图1[2]所示。

*)基金项目:国家自然科学基金资助项目(60273080);符号计算与知识工程教育部重点实验室支持。张 健 博士研究生,主要研究领域为工作流技术,电子政务,决策支持系统,数据挖掘等。孙吉贵 教授,博士生导师,主要研究领域为人工智能,自动推理,约束程序设计等。李妮娅 博士研究生,主要研究领域为工作流技术,产品配置,产品数据管理等。胡成全 教授,主要研究领域为计算机安全和信息加密,逆向工程等。

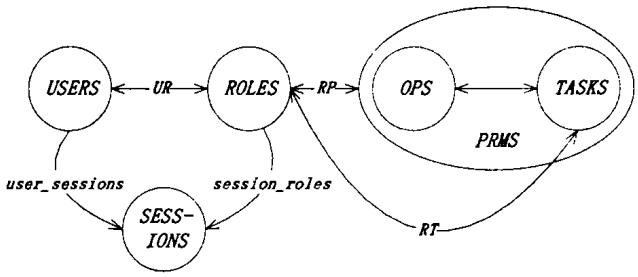

图1 传统的 RBAC 模型

在该模型的基础上，定义如下基于 RBAC 模型的工作流模型。

定义1 工作流 WF 基于角色的访问控制模型 RBAC 是一个七元组，RBAC = $\langle USERS, ROLES, TASKS, OPS, PRMS, SESSIONS, CONSTRAINTS \rangle$，其中 $CONSTRAINTS$ 是执行任务约束集合，其包含下列约束关系[2]。

1) 用户-角色关系：$UR \subseteq USERS \times ROLES$，$(u, r) \in UR$ 表示用户 u 拥有角色 r。

a) $assigned\_users(r) = \{u \in USERS \mid (u, r) \in UR\}$，表示拥有角色 r 的用户集合；

b) $assigned\_roles(u) = \{r \in ROLES \mid (u, r) \in UR\}$，表示分配给用户 u 的角色集合。

2) 角色-权限关系：$RP \subseteq PRMS \times ROLES$，$(p, r) \in RP$ 表示角色 r 具有权限 p。

$assigned\_permissions(r) = \{p \in PRMS \mid (p, r) \in RP\}$，表示角色 r 具有的权限集合。

3) 角色-任务关系：$RT \subseteq ROLES \times TASKS$，$(r, t) \in RT$ 表示授权角色 r 可以执行任务 t。

$authorized\_roles(t) = \{r \in ROLES \mid (r, t) \in RT\}$，表示可以执行任务 t 的授权角色集合。

4) $user\_sessions(u)$ 表示用户 u 对应的会话集合。

5) $session\_roles(s) \subseteq \{r \in ROLES \mid (session\_users(s), r) \in UR\}$，表示会话 s 激活的角色集合，其中 $session\_users(s)$ 表示会话 s 对应的用户集合。

6) $avail\_session\_permissions(s) = \bigcup_{r \in session\_roles(s)} assigned\_permissions(r)$，表示在会话 s 中一个用户可用的权限集合。

定义2 $RH \subseteq ROLES \times ROLES$，定义 $\preceq^R$ 是 RH 上的一个偏序关系，用来表达角色继承特性。对 $\forall r_1, r_2 \in ROLES$，$r_1 \preceq^R r_2$，当且仅当 $authorized\_permissions(r1) \subseteq authorized\_permissions(r_2)$，其中 $authorized\_permissions(r) = \{p \mid p \in PRMS, r' \in ROLES, (r', r) \in RH, p \in assigned\_permissions(r')\}$，它表达在 RBAC 模型中祖先角色继承后裔角色的权限。定义 $(r)_{RH} = \{r' \mid r' \in ROLES, r \preceq^R r'\}$，表达在角色层次关系 RH 上 r 和 r 的祖先角色的并集。例如，在图2中，$(r_3)_{RH} = \{r, r_2, r_3\}$，即存在如下关系：$r_3 \preceq^R r_2 \preceq^R r$。

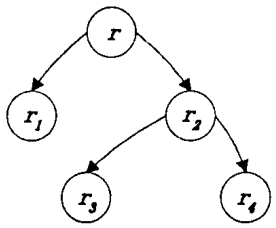

图2 一个简单的角色层次结构图

下面以一个简化的公文处理工作流为例，说明 CRDWR 模型的提出背景和基本原理，如图3所示。假设公文处理工作流由以下五个任务组成：收文签收，收文拟办，批示办理，发文和办结。其中收文签收授权收文员角色执行，收文拟办授权拟办员角色执行，批示办理授权承办处室副处长角色和承办处室处长角色协同执行，发文授权发文员角色执行，办结授权承办处室处员角色执行。授权角色与相应任务的连接弧上的数对分别表示任务的一次成功执行由该授权角色必需成功激活该任务的次数和轮到该授权角色执行该任务时所需要的序次。例如，图3中承办处室副处长角色与批示办理任务连接弧上的数对为(3, 1)，它表示批示办理任务的一次成功执行需要由承办处室副处长角色成功激活3次，并且轮到该角色第1位激活。

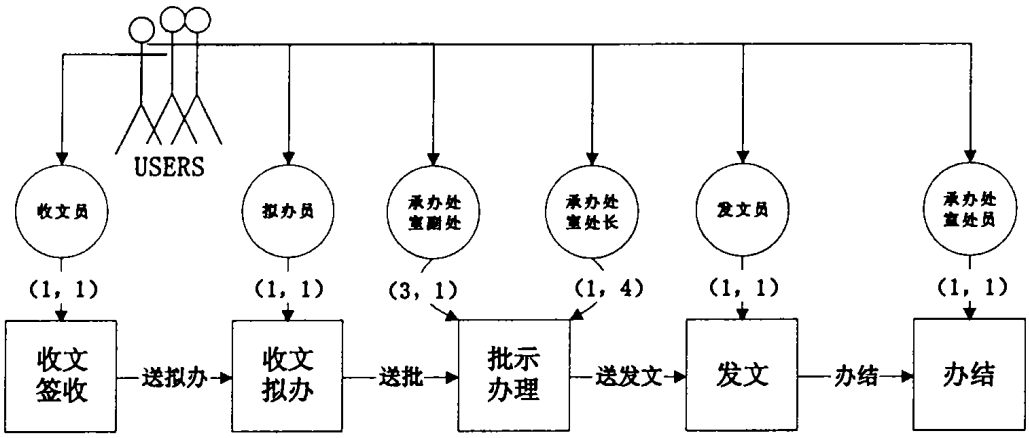

图3 一个简化的公文处理工作流

如果采用传统的 RBAC 模型对图3所示的公文处理工作流的访问授权进行建模，往往需要对任务进行分解才能实现。例如，批示办理这个任务的一次成功执行，需要承办处室副处长角色和承办处室处长角色分别成功激活3次和1次，因此在传统的 RBAC 模型中，需要把这个任务分解成4个子任务。如果有大量任务需要分解时，必然会造成工作流引擎过于庞大，当产生任务执行步骤错误时难于回滚(roll back)，而且任务分解本身往往是非常困难的[3]。

另一方面，传统的 RBAC 模型无法直接表达同一任务的不同授权角色间的执行顺序问题。例如，批示办理这个任务的执行顺序必须是先由所有的授权承办处室副处长激活任务，然后才能由授权的承办处室处长来激活任务。

此外，在工作流执行过程中，常常会遇到根据当前任务的执行状况来动态决定其后续任务的角色分配问题，例如，拥有拟办员角色的用户根据当前来文的密级和重要程度，看出该文件的密级较低，重要程度一般，那么他完全可以动态地决定

其后续任务(批示办理)可以只由承办处室副处长角色2次激活该任务即可,从而可以减少公文的处理时间,提高办文效率,这对工作流系统本身来说也可以大大减轻工作流引擎的负荷。

鉴于上述公文处理工作流的访问控制需求,可以总结出适合工作流系统的3条访问控制策略。

策略1:在工作流任务实例的一次执行中,多个角色可以分别一次或多次激活该任务[3]。

策略2:在工作流任务实例的一次执行中,不同的多个角色激活任务必须是有序的[3]。

策略3:在工作流实例的一次执行过程中,当前任务的执行者可以动态地为其后续任务分配角色和赋予相应的权值(有关权值的定义将在后续内容中作详细介绍)。

本文在传统的RBAC模型的基础上,提出了一个适合工作流系统的访问控制模型CRDWR,它不仅保持了传统RBAC模型的特性,而且还满足策略1～策略3。

3 CRDWR模型

定义3 V表示工作流过程中的全局变量集合。对于V中的每一个元素v,表示一个全局变量。全局变量的功能主要是用于存储在工作流过程中全局范围内都可见的数据信息和控制信息,以及在任务之间传递这些数据信息和控制信息。

定义4 OV表示任务的输出变量集合。对于OV集合中的每一个元素ov,都是由该任务创建的一个输出变量。

定义5 IV表示任务的输入变量集合。对于IV集合中的每一个输入变量iv,都来源于该任务所在的工作流程中的全局变量集合和其前驱任务集合$preN$的输出变量集合的并集。即假设该任务所在的工作流程中的全局变量集合为V,$preN$中共有k个元素,分别为$n_1,n_2,\cdots,n_k$,其中n_i的输出变量集合为$OV_i(i=1,2,\cdots,k)$,那么就有,$iv\in(\bigcup_{i=1}^{k}OV_i)\cup V$。引入输入变量的目的是向当前任务传递来源于全局的以及其前驱任务集合的数据信息和控制信息。

定义6 在工作流实例运行阶段,根据当前任务的IV集合中变量值的不同,动态地为该任务授权角色并赋予相应的权值,从而使任务的自动柔性授权执行更加灵活,更易满足用户的需求。这种静态角色授权和动态角色授权相结合的访问控制方式,称为条件化RBAC。

在图3所示的简化的公文处理工作流中,如果拥有拟办员角色的用户根据当前来文的密级和重要程度,看出该文件的密级较低,重要程度一般,那么他就可以为该拟办任务的输出变量x(代表密级)和y(代表重要程度)都赋值为-1。这样,当其后续的批示办理任务生成的时候,工作流引擎就可以根据其输入变量x'和y'的值(来源于拟办任务的输出变量x和y),动态地为该任务授权角色为:承办处室副处长(2,1),即可以只由承办处室副处长2次激活该任务即可,如图4所示。

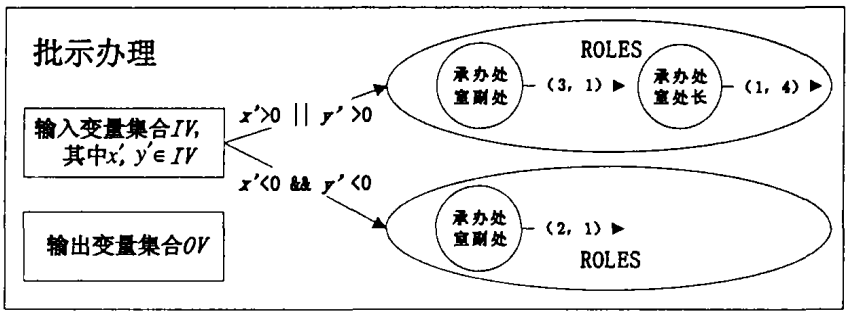

图4 条件化RBAC模型的一个实例

定义7 基于双权角色的工作流授权约束是一个四元组,$dwgrant=(task,\ authorized\_roles(task),\ weight_{num},\ weight_{sort})$,其中,$task\in TASK$。对$\forall\ r\in authorized_roles(task)$,按照角色继承和权限继承的特性,根据定义2,$r$所拥有的权限集合为:$authorized\_permissions(r)=\{p\mid p\in PRMS,\ r'\in ROLES,\ (r',r)\in RH,\ p\in assigned\_permissions(r')\}$。用$|r|$表示$authorized_permissions(r)$集合中的元素个数,按$|r|$由小到大排列使$authorized_roles(task)$成为有序集,在保持其它约束的前提下,容易选择包含权限尽可能少的角色执行任务,从而能够实现访问授权的最小权限原则[3]。

$weightnum$采用文[3]中的定义,即对$\forall\ r\ authorized_roles(task)$,$weight_{num}(r,task)$表示在一次任务执行中,角色$r$需要激活任务$task$的次数。如果$task$的一次执行总共需要激活$n$次,那么根据权限继承的特性,就一定有下列式子成立:

$$\begin{cases}\sum\limits_{r\in authorized\_roles(task)} weight_{num}(r,task)=n\\ \sum\limits_{r'\in(r)RH} weight_{num}(r',task)=weight_{num}(r,task)\end{cases}$$

$weight_{sort}(r,\ task)$表示轮到角色r激活任务$task$时所需要的序次。由$weight_{sort}$的定义可以看出,同一个角色多次激活任务的序次是不定的,即$weight_{sort}$在这里不规定同一个角色多次激活任务的序次,而只规定不同的角色激活任务的序次。例如,图3中的批示办理任务的一次成功执行,需要承办处室副处长角色成功激活3次,需要承办处室处长角色成功激活1次,且承办处室副处长角色优先于承办处室处长角色激活该任务,但是对于承办处室副处长角色的3次激活却不规定其序次。这种定义在实际应用中是有现实意义的,即具有相同角色的用户在绝大多数情况下不需要排序,一般都采用先来先操作(FIFO)的策略。

对$\forall\ r_1,r_2\in authorized_roles(task)$,如果$r_1$优先于$r_2$激活任务$task$,我们记为$r_1\prec_2$。假设$r_1,\ r_2,\ \cdots\cdots,\ r_k\in authorized_roles(task)$,且$r_1\prec r_2\prec\cdots\cdots\prec r_k$,$k=|authorized\_roles(task)|$,定义$weight_{sort}$如下:

$$\begin{cases}weight_{sort}(r_1,task)=1\\ weight_{sort}(r_{i+1},task)=\sum\limits_{j=1}^{i}weight_{num}(r_j,task)+1\end{cases}\qquad 1\leqslant i<k$$

显然,使用$dwgrant$描述的工作流系统授权模型很容易表达第2节中所提出的策略1～策略3的授权约束。

4 算法和实例

定义8 对$\forall\ r\in authorized_roles(task)$,若$r$第$k$次成功激活任务$task$,则称$k$为任务$task$在当前时刻的令牌值($token$),其中$0\leqslant k\leqslant n$,$n$为$task$的一次执行需要成功激活的总次数。

根据定义7和定义8,给出多个角色协同执行任务的序约

束算法描述。

算法1：COSC($task$, $authorized\_roles(task)$, $weight_{num}$, $weight_{sort}$, n)

1)初始化 $token=0, r_{cur}=NULL$;

2)如果 $token=n$,则算法成功结束,否则转3);

3)如果$\exists$ r,使得 $weight_{sort}(r, task)=token+1$,那么对$\forall$ $r'\in(r)_{RH}$,可以选择 r'激活任务 $task$,如果成功激活任务 $task$,则赋值 $r_{cur}=r, token=token+1$,转2),否则重复执行3);如果不存在这样的 r,则转4);

4)选择$\forall$ $r''\in(rcur)_{RH}$的角色激活任务 $task$,如果成功激活任务 $task$,则赋值 $token=token+1$,转2),否则重复执行4)。

下面以图3所示的公文处理工作流为例,使用算法1的步骤来解决批示办理任务中多个角色协同执行任务的序约束问题。为了阐述清晰,从图3中单独提出批示办理任务及其相应的授权角色和权值,如图5所示。

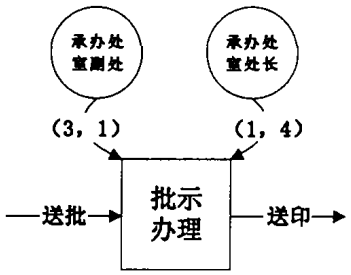

图5 应用算法1解决序约束问题的一个实例

1) 初始化 $token=0, r_{cur}=NULL$;

2) $n=4, token=0\neq n$;

3) $\exists$ $r=$承办处室副处长,使得 $weight_{sort}(r, task)=token+1=1$,假设其成功激活任务,则 $r_{cur}=$承办处室副处长,$token=token+1=1$;

2) $n=4, token=1\neq n$;

3) 不存在这样的 r,使得 $weight_{sort}(r, task)=token+1=2$;

4) 选择 $r''\in(rcur)_{RH}=$(承办处室副处长)$_{RH}$,假设 r''成功激活任务,则 $token=token+1=2$;

2) $n=4, token=2\neq n$;

3) 不存在这样的 r,使得 $weight_{sort}(r, task)=token+1=3$;

4) 选择 $r''\in(r_{cur})_{RH}=$(承办处室副处长)$_{RH}$,假设 r''成功激活任务,则 $token=token+1=3$;

2) $n=4, token=3\neq n$;

3) $\exists$ $r=$承办处室处长,使得 $weight_{sort}(r, task)=token+1=4$,假设其成功激活任务,则 $r_{cur}=$承办处室处长,$token=token+1=4$;

2) $n=4, token=4=n$,算法成功结束。

结束语 工作流与基于角色的访问控制均是目前计算机领域研究的热点问题。一个安全的访问控制模型是工作流管理系统中必不可少的重要组成部分,也是当前工作流研究的一个重要内容。由于传统的RBAC访问控制模型已经不能表达复杂的工作流安全访问控制约束,因此本文基于传统的RBAC模型,提出了一个新的基于双权角色的条件化RBAC访问控制模型CRDWR,阐述了基于动态角色分配的条件化RBAC策略,定义了基于双权角色的工作流系统访问授权新概念,并针对多个角色协同执行任务的序约束问题给出了基于令牌的序约束算法。该模型实现了角色的动态分配,实现了角色执行任务的事务完整性约束和多个角色协同执行任务的序约束,能够表达复杂的工作流安全访问控制约束。CRDWR模型还有许多需要研究和完善的地方,例如,访问授权的最小权限原则的执行算法,同一个角色多次激活任务的序约束,以及条件化RBAC访问控制模型中的角色冲突消解等问题,都有待于进一步深入研究和探讨。

参 考 文 献

1 Hollingsworth D. Workflow Management Coalition: The Workflow Reference Model [R]. Document Number WFMC-TC00-1003, Brussels, 1994

2 Ferraiolo D F, et al. Proposed NIST Standard for Role-Based Access Control [R]. ACM Transactions on Information and System Security, 2001,4(3):231～241

3 Wang X M, Zhao Z T, Hao K G. A Weighted Role and Periodic Time Access Control Model of Workflow System [J]. Journal of Software, 2003

4 Fan Y S. Foundation of Workflow Management Technology [M]. Beijing: Tsinghua University Press & Springer Press, 2001

5 Workflow Management Coalition: Workflow Management Coalition Terminology [R]. Document Number WFMC-TC-1011, Brussels, 1996

6 X. 812 ITU. Security Frameworks for Open Systems: Access Control Framework [R],1995

(上接第58页)

不会被高权重队列的分组任意挤占而无限延长,有利于减少分组延迟的向上的毛刺。利用NS2仿真模拟实验能有效验证和解释上述结论。

结束语 网络对QoS的保障需要一套完善的框架和机制,队列调度算法在各个层次上有力地支持了QoS保障的实现。队列调度算法可以在业务流聚合过程中发挥重要的作用:对于边缘路由器,业务状况复杂,通常会根据目的地或业务QoS等级进行业务流聚合,同目的地或同QoS的分组进入同一个队列接受同等级的服务,队列调度算法可以解决各个队列的分组如何汇聚成一条注入流的问题。

队列调度在QoS保障方面发挥着重要作用,同时在拥塞控制、高速路由、交换等方面也是必须考虑的问题。目前,基于GPS的调度算法仍是研究的热点之一,同时出现了基于服务曲线的算法、基于模糊逻辑的队列管理算法等多种新的调度思想。随着网络复杂性的增加和人们对网络理解的深化,队列调度的研究将更加深入,对应用的指导也将更加广泛。

参 考 文 献

1 林闯,等著.计算机网络的服务质量(QoS)[M].北京:清华大学出版社,2004

2 (美)Flannagan M,等著.尹敏,等译.Cisco Catalyst QOS—园区网中的服务质量[M].北京:人民邮电出版社,2004

3 陆慧梅,向勇,史美林. Internet QoS研究[J].小型微型计算机系统,2002,23(7):786～791

4 王重钢,隆克平,龚向阳,程时端.分组交换网络中队列调度算法的研究及其展望[J].电子学报,2001,29(4):553～59

5 姜宁康,李毓麟.NS网络仿真技术及其应用分析[J].小型微型计算机系统,2002,22(4) :415～417

6 金烨,樊骅.NS中数据流传输处理机制分析及流分析器的实现[J].计算机工程与应用,2003(22):153～155