工作流系统上下文相关访问控制模型*)

王小明 刘 丁 付争方
(陕西师范大学计算机科学学院 西安 710062)

摘 要 访问控制是提高工作流系统安全性的重要机制。基于角色的访问控制(RBAC)被绝大多数工作流系统所采用,已成为工作流领域研究的热点。但是,现有的基于角色的访问控制模型没有考虑工作流上下文对任务执行授权安全的影响,容易造成权限冗余,也不支持职责分离策略。该文提出一种工作流上下文相关访问控制模型 WfCAC,首先,定义该模型的构成要素和体系结构,然后讨论工作流职责分离和访问控制机制,并对模型性质进行分析。WfCAC模型支持用户组及其层次结构,支持最小权限授权策略和职责分离策略,实现了工作流上下文相关访问控制。

关键词 工作流系统,访问控制,上下文相关,安全策略,规则

A Context-sensitive Access Control Model for Workflow System

WANG Xiao-Ming
(College of Computer Science, Shanxi Normal University, Xi'an 710062)

Abstract Access control is an important mechanism for enhancing workflow system security, Role-based access control model (RBAC)is used in the most of workflow systems, and it has become a research topic in the area of workflow. However, in the existing role-based access control models, the influence produced by workflow context and task histories to authorization security is not token into account, redundant properties for running workflow tasks are produced easily, and the policies of separation of duties are not effectively supported. In this paper, a context-related access control model for workflow system is proposed, named as WfCAC. Firstly, the elements and architecture of this model are defined, respectively. Secondly, the mechanisms for achieving the policies of separation of duties and access control mechanism are discussed. Finally, the properties of this model are analyzed. WfCAC model supports the policy of the user group with hierarchy structure, the context- sensitive access control of workflow, the minimizing authorization policies and the policies of separation of duties, respectively.

Keywords Workflow system, Access control, Context-sensitive, Security policy, Rule

1 引言

工作流(Wf)技术是复杂多任务协同建模的一种有效方法,也是 CSCW 领域的一项关键技术,已广泛应用于分布式和网络化生产制造、软件工程、空间科学以及生物 DNA 序列研究等重要领域建模。工作流由一组协同任务及其依赖关系构成,工作流管理系统(WfMS)支持工作流结构定义、任务调度和约束实施,使多任务协同执行以便实现特定的组织目标。工作流任务由特定的用户或进程(以下统称用户)按照一定的组织策略协同执行。为了保证任务执行的安全性,任务执行必须由已授权用户完成,通过工作流的访问控制(access control)机制实现[2,3]。访问控制是系统安全最重要的核心策略之一,其任务是保证系统资源只能被授权用户访问,防止非授权用户访问系统[3,4]。

环境动态变化是现代工作流的重要特征,所以它的访问控制机制也必须对环境动态变化具有主动适应能力,这就要求对工作流任务执行实施上下文相关(context-sensitive)访问控制。事实上,在工作流系统中用户执行任务所需要的权限与工作流上下文环境以及工作流执行历史密切相关。因此,工作流访问控制的突出特点是其具有高度的动态性、上下文相关性以及访问历史相关性。本文提出一种工作流上下文相关访问控制模型 WfCAC,用于解决工作流任务执行的动态授权、上下文相关授权和基于历史的授权问题。

本文第 2 节讨论 WfCAC 模型构成要素、体系结构和访问控制策略及其实现机制;第 3 节对模型的性质进行分析;第 4 节对相关工作进行比较;最后总结全文,并提出有待进一步研究的相关问题。

2 WfCAC 模型

2.1 WfCAC 模型构成要素

目前,以 RBAC 为基础的工作流访问控制模型的基本原理是:任务和用户作为角色的成员分别分配给角色。资源及其操作偶对称为许可,分配给角色。当某任务被 WfMS 调度执行时,按该任务执行所需要的角色选择享有该角色的用户,当前享有任务执行角色的用户均有资格执行任务,工作流执行引擎根据系统当前环境选择适当用户,被选择的用户通过工作流会话激活角色从而获得角色包含的全部许可,从而执行相应的任务,其体系结构如图 1,其中 U, R, P, S, T, Σ 分别是用户集,角色集,许可集,会话集,任务集和约束规则集。URA,PRA 和 RH 分别是用户-角色赋值,许可-角色赋值

*)本课题得到国家自然科学基金项目(10571112)资助。王小明 博士,教授,CCF 高级会员,主要研究领域为系统安全的访问控制,工作流与数据库系统安全模型。

和角色层次关系，SU 和 SR 分别是会话-用户和会话-角色关系，意义与 RBAC 模型相同[4]。TRA 是任务-角色赋值关系。双箭头表示一对多或多对多关系，单箭头表示一对一关系。带箭头的虚线表示上述赋值关系和激活关系满足一定的约束。

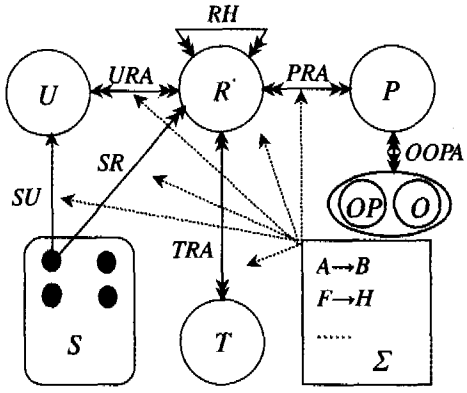

图 1　传统工作流访问控制模型

深入分析上述传统的工作流访问控制模型可以发现，任务和许可关于角色的赋值在逻辑上是互不关联的，并假设总是存在执行特定任务的最小角色。最小角色存在性假设是为了实现任务执行的最小权限授权原则。但是，这一假设使得大规模工作流访问控制的角色设计十分困难。一方面由于大量不同任务实例导致需要定义许多最小角色，从而使角色空间爆炸性增长。另一方面，许可-角色赋值和任务-角色赋值很难保持相关性和一致性。事实上，由于角色-许可赋值的静态性和任务执行的动态性，最小角色定义实际上是不可能实现的。现有的基于角色的工作系统访问控制模型的许可使能机制是只要角色被激活，分配给角色的许可全部无条件使能，因此它们只能实现角色粒度上的任务执行访问控制，无法根据任务执行的上下文环境在许可级别上进行细粒度访问控制。为了深入讨论工作流上下文相关访问控制，对工作流上下文定义如下：

定义 1　工作流访问控制上下文是用户环境和应用环境以及与访问控制决策相关的信息集，也称为工作流访问控制上下文环境。

工作流访问控制上下文由工作流引擎从任务执行上下文和任务执行历史产生，如用户标识，已激活角色，使能许可，资源状态，时间和空间位置等。当工作流实际执行时，WfMS 把每一步执行的历史数据记录在日志数据库中。因此以下讨论假定工作流访问控制上下文总是可以获得的。在此基础上，提出工作流上下文相关访问控制模型 WfCAC，它的基本构成元素是用户，用户组，角色，任务，许可和约束。用户是执行任务的主体，用户组是用户的集合，用于对组织结构进行建模。角色是用户组的职责，如部门职责等。任务由授权用户执行，许可是执行任务所需要的资源以及在资源上的具体操作，例如对文件的读或写操作等。用户分配给用户组，用户组之间可能存在层次结构关系，映射组织部门之间的上下级管理关系；用户组分配给角色，从而使用户组中的全部用户扮演相应的角色。任务分配给执行该任务所需要的角色，许可分配给任务。当任务被调度执行时，根据任务执行所需要的角色选择分配给角色的用户组，从而选择该用户组中的用户，被选择的用户激活角色从而执行相应任务的许可。约束是对上述关系的有效性限制。WfCAC 模型体系结构如图 2。

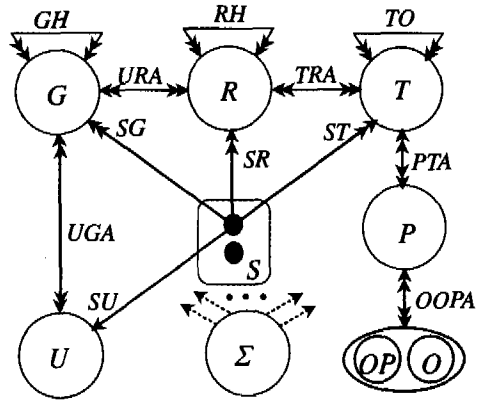

图 2　WfCAC 模型

WfCAC 模型由集合、关系和函数构成，定义如下：

定义 2(基本集合)　设 I 为自然数集，$T=\{t_i\}_{i\in I}$ 是工作流任务集，$T\_inst=\{t\_inst_i\}_{i\in I}$ 是任务实例集，$R=\{r_i\}_{i\in I}$ 是角色集，$U=\{u_i\}_{i\in I}$ 是用户集，$P=\{p_i\}_{i\in I}$ 是许可集，$S=\{s_i\}_{i\in I}$ 是工作流会话集，$O=\{o_i\}_{i\in I}$ 是资源集，$\psi=\{\varepsilon_i\}_{i\in I}$ 是资源类型集，$OP=\{op_i\}_{i\in I}$ 是资源上的原子操作集，$G=\{g_i\}i\in I$ 是用户组集，Σ 是工作流上下文约束规则集。

定义 3　工作流任务是一个三元组 $t=(\Gamma_t,\Delta_t,\Sigma_t)$，其中 Γ_t 是 t 执行所需要的许可集，$\Delta_t\square\psi$ 是 t 执行所需要的资源类型集，$\Sigma_t\square\Sigma$ 是执行 t 的约束集。

定义 4　工作流上下文相关访问控制模型是一个七元组 $WfCAC=(T, R, G, U, P, S, \Sigma)$。其中，$T, R, G, U, P, S, \Sigma$ 意义同定义 2。

定义 5(关系)

(1)任务-角色赋值关系：$TRA\square T\times R,(t, r)\in TRA$ 表示授权角色 r 可以执行任务 t；

(2)用户-用户组赋值关系：$UGA\square U\times G,(u,g)\in UGA$ 表示用户 u 是用户组 g 的成员；

(3)用户组-角色赋值关系：$GRA\square G\times R,(g,r)\in GRA$ 表示用户组 g 分配给角色 r；

(4)任务-许可赋值关系：$TPA\square T\times P,(t,p)\in TPA$ 表示任务 t 的执行需要许可 p。

(5)角色层次结构偏序关系：$RH\square R\times R,(r_i,r_j)\in RH$ 表示角色 r_i 是角色 r_j 的父角色，r_j 是 r_i 的子角色，父角色继承其子角色能够执行的任务。简记为 $r_i\square r_j$。当 $r_i\neq r_j$ 时，记为 $r_i\square r_j$。

(6)工作流会话：$S\square 2^G\times U\times 2^R\times 2^T,(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\in S$ 表示用户组 $g_1,\cdots,g_m$ 的用户 u 激活角色 $r_1,\cdots,r_n$ 从而执行任务 $t_1,\cdots,t_k$。用户的一个会话可以和多个用户组、角色和任务相关联，从而实现任务执行协同；

(7)任务执行偏序关系：$TO\square T\times T,(t_i,t_j)\in TO$ 表示成功执行 t_i 之后才能激活 t_j；

(8)用户组层次结构偏序关系：$GH\square G\times G,(g_i,g_j)\in GH$ 表示用户组 g_i 是用户组 g_j 的上级组，g_j 是 g_i 的下级组，上级组继承下级组的管理权限。简记为 $g_i\square g_j$。当 $g_i\neq g_j$ 时，记为 $g_i\square g_j$；

(9)操作-资源-许可赋值关系：$OOPA\square P\times(OP\times O)$，$(p,(op,o))\in OOPA$ 表示许可 p 能够对资源 o 实施操作 op。

定义 6(函数)

(1)用户组-用户函数：$groupuser: G\to 2^U$。

$groupuser(g)=\{u\in U|(u,g)\in UGA\}$；

(2)用户-用户组函数：$usergroup$：$U\rightarrow 2^G$。

$usergroup(u)=\{g\in G|(u,g)\in UGA\}$；

(3)用户组-角色函数：$grouprole$：$G\rightarrow 2^R$。

$grouprole(g)=\{r\in R|r_i\square r,(g,r_i)\in GRA\}$；

(4)角色-用户组函数：$rolegroup$：$R\rightarrow 2^G$。

$rolegroup(r)=\{g\in G|r_i\square r,(g,r_i)\in GRA\}$；

(5)角色-任务函数：$roletask$ $R\rightarrow 2^T$。

$roletask(r)=\{t\in T|r_i\square r,(t,r_i)\in TRA\}$。

(6)任务的前任务集函数：$pretask$：$T\rightarrow 2^T$。

$pretask(t)=\{t_i|(t_i,t)\in TO\}$；

(7)任务的后任务集函数：$postask$：$T\rightarrow 2^T$。

$postask(t)=\{t_i|(t,t_i)\in TO\}$；

(8)任务-角色函数：$taskrole$：$T\rightarrow 2^R$。

$taskrole(t)=\{r\in R|r\square r_i,(t,r_i)\in TRA\}$；

(9)角色-任务函数：$roletask$：$R\rightarrow 2^T$。

$roletask(r)=\{t\in T|ri\square r,(t,r_i)\in TRA\}$；

(10)任务-许可函数：$taskperm$：$T\rightarrow 2^P$。

$taskperm(t)=\{p\in P|(t,p)\in TPA\}$；

(11)许可-任务函数：$permtask$：$P\rightarrow 2^T$。

$permtask(p)=\{t\in T|(t,p)\in TPA\}$；

(12)会话-任务函数：$sesstask$：$S\rightarrow 2^T$。

$sesstask(s)=\{t_1,\cdots,t_k\in T|(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\in S\}$；

(13)会话-角色函数：$sessrole$：$S\rightarrow 2^R$。

$sessrole(s)=\{r_1,\cdots,r_n\in R|(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\in S\}$；

(14)会话-用户组函数：$sessgroup$：$S\rightarrow 2^G$。

$sessgroup(s)=\{g_1,\cdots,g_m\in G|(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\in S\}$；

(15)会话-用户函数：$sessuser$：$S\rightarrow U$。

$sessuser(s)=u\in U,(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\in S\}$；

(16)任务-任务实例函数：$tins$：$T\rightarrow 2^{T\_inst}$。

$tins(t)=\{t\_inst|$ t_inst 是 t 的实例$\}$。

在 WfCAC 模型中，用户组用来管理人力资源，角色用来表达特定部门的业务职能。虽然角色和用户组的语义十分相似，但二者建模的对象不同，角色是用户组和任务的集合，而用户组是用户和角色的集合。通常情况下，组织部门和责任是不同的两个范畴，前者以人和职位为中心，后者以责任和行为活动为中心对组织结构及其功能进行描述。职位和责任相分离有助于实现细粒度访问控制。在工作流中存在两种重要的活动过程，一种是用户执行任务过程，另一种是对用户执行任务过程的监控和评价过程，二者不可缺少。在 WfCAC 模型中，我们规定上级用户组的用户负责对下级用户组的用户行为进行监控、审计和评价等管理。对任意用户组 $g\in G$，对其负有管理责任的用户组集合为：

$$admingroup(g)=\{g|g_i\square g\} \quad (1)$$

对一个特定的用户组 g 的用户 u，对 u 在 g 部门的活动能够实施管理的用户全集为：

$$adminuser(u,g)=\{u(U|u\in g_i,g_i\in admingroup(g)\} \quad (2)$$

用户履行其管理职责可以和工作流任务执行同步或异步，如对实时工作流的管理可能需要与工作流任务执行同步，而非实时工作流可以在任意时刻实施管理过程。能够执行任务 $t\in T$ 的候选用户集为：

$$cando\_user(t)=\{u\in U|u\in g,g\in taskrole(t),\Sigma_t \text{ 被满足}\} \quad (3)$$

工作流引擎能够快速从上述规模比较小的用户集中选择恰当的任务执行所需要的用户。对任意 $s\in S$，s 的用户 u 在 s 中能够获得的许可集为：

$$sessperm(s)=\{p(P|p(taskperm(t),\Sigma_t \text{ 被满足}\} \quad (4)$$

Σ_t 中包含了任务-许可赋值约束，它与工作流执行历史、系统当前状态以及外部环境有关。因此，特定的用户组的用户通过特定的角色执行特定任务时所获得的许可是上下文相关的。显然，$sessperm(s)\subseteq taskperm(t)$ 成立。为最大限度支持访问控制的最小权限和职责分离策略，需要预先确定不同环境下任务执行所需要的许可子集，用任务授权模板描述，定义为：

定义 7 任务授权模板是一个六元组 $\sigma=(g,u,r,t,Z_p,C)$。其中 $u\in groupuser(g)$，$g\in rolegroup(r)$，$r\in taskrole(t)$，$t\in T$，$Z_p\square taskperm(t)$，Z_p 满足约束 $C\square\Sigma_t$ 并且 $Z_p\neq\emptyset$。

任务 t 的授权模板全集记为 $tem(t)$，所有任务授权模板全集记为 TEM。授权模板用于实时推导实际授权。执行任务的用户组、角色、资源类型和约束均在任务定义时确定，执行任务所需要的许可根据任务执行上下文动态确定。一个任务的授权模板可能有多个，当任务 t 开始执行时，由 t 的授权模板推导的许可被执行，而 t 的其它许可处于阻塞状态，从而支持上下文相关最小权限策略。为了实现细粒度任务执行动态约束，根据工作流上下文和任务执行历史从授权模板中可以进一步推导出授权信牌(token)，使用授权信牌能够实现任务实例级、许可级和资源级动态访问控制。

定义 8 授权信牌是一个九元组 $\omega=(g,u,r,t,t\_inst,p,n,U_t,\bar{U}_t)$，任务 t 的实例 t_inst 执行时，用户组 g 的用户 u 通过角色 r 成功执行 n 次 p。U_t 和 $\bar{U}_t$ 分别是 ω 的已执行用户集和禁止执行用户集。

任务实例 t_inst 的授权信牌全集记为 $tok(t\_inst)$，所有任务实例授权信牌全集记为 TOK。授权信牌确定了参与任务实例执行的用户组、角色、资源和使能许可及其成功执行次数。如果 ω 是从授权模板 $\sigma(tem(t))$ 推导而来的，则记为 $\sigma\square\omega$。ω 产生初期，$n>0$，并且 $U_t=\square$。用户每执行一次 p，n 值减 1，该用户被添加到 U_t 中。当 $n=0$ 时，授权信牌无效，意味着不允许在 t_inst 中执行 p。为描述简洁，用 $g(\sigma)$，$u(\sigma)$，$r(\sigma)$，$t(\sigma)$，$Z_p(\sigma)$，$C(\sigma)$ 表示授权模板在其分量上的投影，$g(\omega)$，$u(\omega)$，$r(\omega)$，$t(\omega)$，$t\_inst(\omega)$，$p(\omega)$，$n(\omega)$，$U_t(\omega)$，$\bar{U}_t(\omega)$ 表示授权信牌 ω 在其分量上的投影值。为使 WfCAC 模型支持许可粒度的动态访问控制策略(如 Chinese Wall 策略)，我们提出任务实例授权临界信牌集和临界许可概念。

定义 9(临界信牌集) 设 $critical(t\_inst)=\{\omega_1,\omega_2,\cdots,\omega_m\}\square tok(t\_inst)$ 是任务实例 t_inst 的授权信牌子集，当 t_inst 执行完成时，对任意 $\omega_i,\omega_j\in critical(t\_inst)$，$U_t(\omega_i)\square U_t(\omega_j)=\square$，则称 $critical(t\_inst)$ 是 t_inst 的临界信牌集，并称 $p(\omega_i)$ 和 $p(\omega_j)$ 是 t_inst 的临界许可。

任务实例 t_inst 的临界许可集为：

$$criticalperm(t\_inst)=\{p_i(P|p_i=p(\omega),\omega(critical(t\_inst))\} \quad (5)$$

在上述概念基础上，定义任务实例的否定用户概念如

下：

定义 10(否定用户) 设 $t\in T$，$t\_inst(tins(t)$，对任意 $\omega_i,\omega_j\in tok(t\_inst)$，如果用户 u 已执行 ω_i 中出现的许可 p_i，那么不允许 u 再执行 ω_j 中出现的许可 p_j。称 u 是 ω_j 的一个否定用户，即 $u\in\bar{U}_t(\omega_j)$。

对任意 $\omega(critical(t\_inst),\omega$ 的否定用户集由下列公式求得：

$$\bar{U}_t(\omega)=\{u(U|\omega'\in critical(t\_inst),\ u\in U_t(\omega')\} \quad (6)$$

由定义 8 和定义 10 可以直接得出下列结论：

定理 1 设 $t(T,\ t\_inst\in tins(t),\omega\in critical(t\_inst)$，那么 $U_t(\omega)\square\bar{U}_t(\omega)=\square$。

如果当前从 $cando\_user(t)$ 中选择的执行任务实例 t_inst 的用户 $u\in\bar{U}_t(\omega)$，则出现了用户选择冲突，此时采用“否决优先”策略，禁止 u 执行 ω 中出现的许可操作，重新选择其他可执行用户，从而使冲突得到消解。冲突消解规则定义为：

$$\forall t\in t,\forall t\_inst\in tins(t),\forall\omega\in tok(t\_inst),\forall u\in U,:$$
$$u\in cando\_user(t)\wedge u\bar{U}_t(\omega)\Rightarrow u\notin U_t(\omega) \quad (7)$$

2.2 WfCAC 模型职责分离策略

工作流的访问控制策略主要是任务执行的职责分离和最小权限授权策略[3,9,17]。前者能够有效防止用户访问欺诈行为发生，对协同系统安全十分重要；后者使用户仅仅获得执行任务所必需的最小权限集，一旦任务执行结束，权限立即失效。但是，现有的基于角色的工作流访问控制模型还不支持职责分离策略[3,9]，也不支持真正意义上的最小权限授权策略，职责分离不得不通过应用程序编码并嵌入到任务定义之中，使授权约束实施的灵活性受到极大限制。本文建立任务互斥和授权信牌互斥关系，使 WfCAC 模型不仅支持任务级静态和动态职责分离，而且支持任务实例级动态职责分离。

定义 11(任务静态互斥关系) 设 $t_i,t_j\in T$ 是两个不同的工作流任务，$r\in R$，如果 $t_i\in roletask(r)$，并且 $t_j\in roletask(r)$ 可能造成任务-角色赋值安全隐患，则称 t_i 和 t_j 之间存在静态互斥关系。记为 $(t_i,\ t_j)\in TSEX$。

WfCAC 模型的静态职责分离策略规则定义为：

$$\forall t_i,\ t_j\in T,\forall r\in R:$$
$$t_i\in roletask(r)\wedge t_j\in roletask(r)\Rightarrow(t_i,\ t_j)\notin TSEX \quad (8)$$

定义 12(任务动态互斥关系) 设 $t_i,\ t_j\in T$ 是两个不同的工作流任务，$u\in groupuser(g)$，$g\in rolegroup(r)t_i\in roletask(r)$，$t_j\in roletask(r)$，如果用户组 g 的用户 u 通过角色 r 激活 t_i 和 t_j 可能造成任务执行安全隐患，则称 t_i 和 t_j 之间存在动态互斥关系。记为 $(t_i,\ t_j)\in TDEX$。

WfCAC 模型的动态职责分离策略规则定义为：

$$\forall u\in U,\forall g_1,\cdots,g_m\in G,\forall r_1,\cdots,r_n\in R,\forall t_1,\cdots,t_k\in T,\forall s\in S:$$
$$s=(\{g_1,\cdots,g_m\},u,\{r_1,\cdots,r_n\},\{t_1,\cdots,t_k\})\Rightarrow(t_i,\ t_j)\in\{\{t_1,\cdots,t_k\}\times\{t_1,\cdots,t_k\}\}\wedge(t_i,\ t_j)\in TDEX \quad (9)$$

由定义 11 和定义 12 可以得出下列任务互斥一致性定理。

定理 2 设 $t_i,\ t_j\in T$ 是两个不同的工作流任务，那么 $\forall t_i,\ t_j\in T:(t_i,\ t_j)\in TSEX\Rightarrow(t_i,\ t_j)\in TDEX$。

任务级静态和动态职责分离在工作流中普遍存在。例如，支票签发工作流中，任一用户不得以任何身份(职位或权限)既执行空白现金支票填写任务又执行现金支票审批任务，否则可能造成财务欺诈行为发生。但是，在有些情况下静态职责分离仍然太强，需要在任务实例级实施更细粒度的动态职责分离，从而适当放松任务级约束强度。例如，支票签发工作流中，用户可以既执行空白现金支票填写任务又执行现金支票审批任务，但不允许批准自己填写的现金支票，此类约束在工作人员紧缺状态下能够更好地提高工作效率。任务实例级职责分离通过授权信牌互斥关系实现。

定义 13(信牌互斥关系) 设 $t\in T$ 是工作流任务，$t\_inst_i,\ t\_inst_j\in tins(t)$，$u\in g$，$g\in G$，$r\in R$，存在两个不同的授权信牌 ω_1,ω_2，$tins(\omega_1)=t\_inst_i$，$tins(\omega_2)=t\_inst_j$，如果 ω_1,ω_2 均被实施可能造成安全隐患，则称 ω_1 和 ω_2 之间存在授权信牌互斥关系。记为 $(\omega_1,\omega_2)\in TOKEX$。

设 $do(\omega)$ 为执行信牌谓词，任务实例级职责分离策略规则为：

$$\forall\omega_i,\omega_j\in TOK:\ do(\omega_i)\wedge do(\omega_j)\Rightarrow(\omega_1,\omega_2)\in TOKEX \quad (10)$$

2.3 访问控制策略实施

WfCAC 模型的访问控制过程由任务(实例)驱动，用户参与，系统协调共同完成。设 $finish(t)$，$activate(t)$，$activated(t)$ 和 $permit(\omega)$ 分别是已执行任务、正在激活任务、已激活任务和执行授权信牌谓词。任务激活规则为：

$$\forall t\in T,\ \forall t_1,\cdots,t_n\in pretask(t):$$
$$activate(t)\Rightarrow(finish(t_1)\vee activated(t_1))\wedge\cdots\wedge(finish(t_n)\vee activated(t_n)) \quad (11)$$

任务被激活之后，需要实施用户访问控制，访问控制规则为：

$$\forall t\in T,\exists\sigma\in TEM,\exists\omega\in TOK,\exists s\in S:\ activate(t)\wedge t\in sesstask(s)\wedge\sigma\in tem(t)\wedge\sigma\square\omega\Rightarrow permit(\omega) \quad (12)$$

3 WfCAC 模型性质

WfCAC 模型具有下列性质：

(1)支持最小权限授权策略。在 WfCAC 模型中按照任务执行上下文环境把任务执行所需要的许可动态分配给任务，任务分配给角色。当任务被 WfMS 调度执行时，用户通过用户组获得角色从而激活任务，执行任务当前被赋值的许可。根据任务执行语义对任务进行最小许可赋值不仅是合理的，而且总是可以实现的。所以 WfCAC 模型支持真正意义上的任务执行最小权限授权策略。

(2)支持职责分离策略。建立在任务静态和动态互斥关系基础上的职责分离使 WfCAC 模型支持任务级静态和动态职责分离；授权信牌互斥关系使 WfCAC 模型支持任务实例级动态职责分离。

(3)支持动态否决授权。否决授权是访问控制的重要机制，但是由于否决授权可能产生授权冲突，因此绝大多数访问控制模型不支持否决授权。WfCAC 模型通过授权临界信牌和临界许可实现了动态否决授权，授权冲突消解方法比较简单。

(4)具有良好的任务、角色和用户组之间协同机制。用户通过一个会话可以激活多个用户组、角色和任务，从而在工作流安全策略允许范围内实现多任务、多角色和多用户组协同，能够使同一用户以不同的身份(部门)通过不同的角色获得不同的许可参与工作流任务的执行活动。

(下转第 124 页)

设备必须能支持 IP 组播。

视频控制的主要功能是对转发站点进行控制，用来建立和管理转发站点上的 IP 组播数据组的传输。控制系统要最大限度地满足完成指向需求用户的数据发送，同时密切注意视频传输的质量。具体地说就是要尽可能多地为同类请求用户发送数据，但要在允许的带宽范围之内。这个带宽是通过计算机实时控制的，计算机实时控制系统随时监控视频传输的质量，自动调整带宽；同时对网络其他各项参数也实现实时监控。可见，视频控制实质上也就是计算机的实时控制。计算机实时控制的好坏直接决定了 IP 组播的效果。

结束语 IP 组播带入了许多新的应用并减少了网络的拥塞和服务器的负担。目前 IP 组播的应用范围还不够大，但它能够降低占用带宽，减轻服务器负荷，并能改善传送数据的质量，尤其适用于需要大量带宽的多媒体应用，如音频、视频等。这项新技术已成为当前网络界的热门话题，并将从根本上改变网络的体系结构。

目前，IP 组播技术在商业应用中还面临着一些需要解决的问题，如组播服务的收费方式和方法；组播网络的监控；组播成员的身份认证；如何保证组播的 QoS；采用何种商业模式向用户推销组播服务等。但可以预见的是，人们日益认识到组播技术所带来的优点和长处，组播技术必将成为构建校园网不可缺少的网络技术之一。

参 考 文 献

1 李炜．IP 组播路由协议的研究与实现．中国工程科学，2002，14(1)

2 姚文杰，等．组播技术在数字视音频监控系统中的应用．微型电脑应用，2001，17(9)

3 王诗成，等．基于区域网通信技术的研究．电子技术，2001(12)

4 谭云兰．Internet 的组播与组播路由实现．计算机与现代化，2001(3)

5 潘继军．基于 ARM 的嵌入式系统实验分析．微计算机信息，2006(5)

(上接第 104 页)

4 相关工作比较

典型的基于角色的工作流访问控制模型是文[8]提出的基于任务的访问控制 TBAC，它使不同的任务步被不同的用户执行，实现了简单的任务执行职责分离。但是它不能有效表达任务授权的上下文相关约束。文[15]提出的工作流授权模型实现了任务授权与任务执行同步，但没有考虑工作流任务执行上下文对授权安全性的影响，不支持角色层次结构和用户组及其层次结构，也不具有职责分离机制。文[18]提出了基于团队的访问控制，其团队概念与本文的用户组概念相似，但不具有层次结构，因此无法对组织层次结构进行有效建模。上述工作流访问控制模型中均把许可赋值给角色，任务也赋值给角色，而且两类赋值是互不相关的。在任何情况下，工作流任务执行上下文对角色获得许可没有直接约束，用户通过激活角色从而获得角色包含的全部许可执行相应的任务，因此它们是以角色为粒度的上下文无关访问控制，没有真正实现最小权限授权策略。本文提出的 WfCAC 模型较好地实现了工作流上下文相关访问控制，具有良好的职责分离机制，支持否定用户授权，也支持许可粒度上的工作流访问控制。

小结 本文以基于角色的访问控制为基础，引入用户组概念，提出了工作流上下文相关访问控制模型 WfCAC。在 WfCAC 模型中，用户分配给用户组，用户组分配给角色，任务也分配给角色，根据任务执行上下文把许可分配给任务。当某工作流任务被 WfMS 调度执行时，用户组的用户激活其享有的角色并通过相应的任务获得满足当前工作流上下文约束的许可，从而完成任务执行。角色能够获得的许可根据任务执行上下文环境和任务执行历史动态确定，从而较好地实现了最小权限授权策略和基于历史的访问控制，并支持任务级和任务实例级职责分离策略。工作流访问控制上下文获取和授权信牌推导高效算法，WfCAC 模型与 WfMS 的有效集成方法等都是十分有意义的研究课题，有待进一步研究。

参 考 文 献

1 Workflow Management Coalition. The workflow reference model Tech Rep: TC00-1003, 1995. http//www. wfmc. org/standard/docs/tc003v11. dbf

2 Workflow Management Coalition. Workflow security considerratins- white paper: [technical report WFMC-TC-1019]. issuel. 1, 1998. http://www. wfmc. org

3 Bertino E, Ferrari E, et al. The Specification and enforcement of authorization constraints in workflow management system. ACM Transactions on Information and System Security, 1999, 2(1): 65～104

4 Ferraiolo D, Sandhu R, Gavrila S, Kuhn R D. Chandr- amouli R. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security, 2001, 4(3): 60～110

5 Oh S, Park S. Task-role-based access control model. Journal of Information System, 2003, 28(7): 533～562

6 Botha A R, Eloff P H. Separation of duties for access control in workflow environments. IBM Systems Journal, 2001, 40(3): 666～682

7 Thomas R K, Sandhu R. Task-based authorization control (TBAC): a family of models for active and enterprising-oriented authorization management. Database Security, XI: Status and Prospects, Lin Y T, Qian S, Editors. London: Chapman and hall Press, 1997. 166～181

8 Thomas K R, Sandhu R. Task-based authorization control (TBAC): a family of models for active and enterprise- oriented authorization management. In: Proceedings of the IFIP WG11. 3 Workshop on Database Security, 1997. 77～93

9 Wu Shengli, Amit S, John M, et al. Authorization and access control of application data in workflow systems. Journal of Intelligent Information Systems, 2002, 18(1): 71～94

10 Evans G N, Mason-Jones R, Towill D R. The scope paradigm of business process reengineering. Business Process Management Journal, 1999, 5(2): 121～156

11 Bertino E, Ferrari E, Atluri V. Specification and enforement of authorization constraints in workflow management systems. ACM Transactions on Information and System Security, 1999, 2(1): 65～104

12 Ahn J G, Sandhu R. Role-based authorization constraints specification. ACM Transactions on Information and System Security, 2000, 3(4): 43～54

13 Henricksen K, Indulska J. Modeling context information in pervasive computing systems. In: Proceedings of the First International Conference, Springer Verlag, 2002. 167～180

14 Simon T R, Zurko E M. Separation of duty in role-based environments. In: Proceedings of ACM on Computer Foundations Workshop, 1997. 43～55

15 Son H J, Oh S K, Choi H K, et al. GM-WTA: an officient workflow task allocation method in a distributed execution environment. Journal of System and Software, 2003, 67(3): 165～179

16 Nitsche U, Holbein R, Morger O, et al. Realization of a context-dependent access control mechanism on a commercial platform. In: Proceedings of the 14th International Information Security Conference, 1998. 160～170

17 Atluri V, Hong W-K. An authorization model for workflows. In: Proceedings of the 5th European Symposium on Research in Computer Security, 1996. 44～64

18 Van der Aalst W M P. A reference model for team-enabled workflow management systems. Data & Knowledge Engineering, 2001, 38(3): 335～363

19 王小明，赵宗涛，郝克刚．工作流带权角色与周期时间访问控制模型．软件学报，2003，14(11)：1841～1848

20 Liu N, Grosof B N, Feigenbaum J. A logic-based approach to distributed authorization. ACM Transactions on Information and System Security, 2003, 6(1): 128～171

21 王小明，赵宗涛，马建峰．一种新的 RBAC 角色协同关系及其 Petri 网模型[J]．电子学报，2003，31(2)：225～227