·体系结构与软件技术· 文章编号：1000—3428(2013)04—0075—03 文献标识码：A 中图分类号：TP393

工作流访问控制模型及动态用户分配策略

贠亚利，韩海晓，李雅萍

(洛阳理工学院计算机与信息工程系，河南 洛阳 471023)

摘 要：针对工作流访问控制灵活性和任务分配动态性较差的问题，提出一种新的工作流访问控制模型，使角色和用户在建模和实例两级实现基于任务的工作流权限管理，将建模阶段的权限管理方案作为模板，工作流实例继承该模板并进行个性化设置。设计工作流访问控制模型的用户分配原则，并提出一种基于繁忙因子的动态用户分配策略。分析结果表明，该模型能够根据历史执行情况进行动态用户分配，从而提高工作流的执行效率。

关键词：工作流；工作流访问控制；动态用户分配；繁忙因子；任务分配依赖；授权

Workflow Access Control Model and Dynamic User Assignment Policy

YUN Ya-li, HAN Hai-xiao, LI Ya-ping

(Department of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang 471023, China)

【Abstract】 In order to solve the problem of inflexible access control and lack of dynamic task allocation for workflow, this paper presents a new workflow access control model. It integrates both users and roles authorization on workflow tasks in two level of modeling stage and instance stage, and also designs dynamic user assignment rules for the model by introducing the load equilibrium-based user assignment strategy on the user busy factor. Analysis result shows that this model can produce dynamic user assignment according to history executive condition, and improve workflow's executive efficiency.

【Key words】 workflow; workflow access control; dynamic user assignment; busy factor; dependency of task assignment; authorization

DOI: 10.3969/j.issn.1000-3428.2013.04.018

1 概述

随着网络技术的发展，工作流[1]技术的应用日益广泛，安全问题也日益突出，访问控制是其中一个重要组成部分，已成为近年研究的热点。研究者提出了一些适合工作流的访问控制模型，如基于角色的访问控制(Role-based Access Control, RBAC)[2]模型、基于任务的访问控制模型(Task-based Access Control, TBAC)[3-4]、基于任务和角色的访问控制(Task-role-based Access Control, TRBAC)模型[5-6]、扩展的基于角色的访问控制模型(Extended Role-based Access Control, ERBAC)[7-9]等，这些模型在工作流建模一级进行任务分配，但在工作流实例执行过程中存在灵活性不够，不能满足工作流实例在执行过程中的个性化用户分配要求的问题。此外现有的工作流用户分配策略[10]以角色为授权单位，并且任务分配没有有效利用历史执行信息，因此，动态特征不足，不能根据任务历史执行情况动态调整任务分配方案。

本文在上述模型的基础上，提出一种两级工作流访问控制模型——基于任务、角色和用户的访问控制模型(Task-role-user-based Access Control, TRUBAC)，不仅可以在工作流建模阶段和工作流实例两级进行权限管理，将建模阶段的权限管理方案作为模板，工作流实例继承建模阶段的方案，并且可以进一步对其进行个性化设置，使得工作流的权限管理更加灵活。模型以任务为单位动态授权、具有可对角色和用户综合授权、两者互为补充的特点，因此，授权机制更加灵活。

2 访问控制模型

RBAC 是将用户对对象的访问权限转换成角色对对象的访问权限，并通过给用户分配不同的角色达到赋予用户不同权限的目的。这种基于角色的权限控制模型直接将企业组织结构映射到信息系统中，使得对用户权限的管理直

基金项目：洛阳理工学院青年基金资助项目(2009QZ29)

作者简介：贠亚利(1975－)，女，讲师、硕士，主研方向：工作流技术，数据库应用及安全；韩海晓，助教；李雅萍，副教授

收稿日期：2012-03-22 **修回日期**：2012-07-05 **E-mail**：397074486@qq.com

观和统一，在很大程度上简化了权限管理工作。

但是，在大型企业系统中用户众多，业务对象复杂，仅使用角色的定义在权限管理中相对复杂，由于用户的业务需求或系统功能的改变，导致权限发生变化。仅采用通过调整角色的定义来改变授权，限制了授权的灵活性。因此，有学者提出 ERBAC 模型，采用对用户和角色混合授权的方法，使得权限控制更加灵活和安全。在该模型中，除了可以对角色授权并将角色授予用户之外，还可以直接对用户授权，即角色授权作为基础手段，用户的直接授权则是在角色授权管理中上的补充和调整。

由于工作流的执行过程以任务为单位，使得安全问题从独立的计算机系统中静态的主体和客体保护转移到随着任务的执行而进行动态授权的保护上，因此工作流需要随着任务的执行而进行动态授权保护，从而产生一种称为基于任务的访问控制模型，它从工作流中的任务角度建模，可以依据任务和任务状态的不同，对权限进行动态管理。这是一种基于实例(instance-based)的访问控制模型，不仅能对不同工作流实行不同的访问控制策略，而且还能对同一工作流的不同任务实例实行不同的访问控制策略。基于任务和角色的访问控制模型 TRBAC 将角色赋予相关的任务，然后再给任务赋予相关的权限，实现权限的按需和动态分配。

本文以上述访问控制模型为基础，提出 TRUBAC 模型，它在综合几种模型特点的基础上进行扩充。不仅可以对工作流基于工作流定义进行任务分配，指定可以执行每一个任务的用户和角色，作为工作流实例的任务分配模板；并且对每一个工作流实例，可以在此分配模板的基础上调整该实例的任务分配方案。因此，对于一个工作流实例的任务，它的权限是两者的综合：来自于工作流定义的隐式权限与来自于工作流实例的显式权限的综合。权限的指派在任务开始执行时才授予具体的用户，该用户是具有执行该任务权限的成员，通过直接指派或通过角色指派之后，其权限被激活，用户对任务及数据的使用有时间和次数的限制。任务执行完成后，系统收回其权限，这样随着任务的执行而进行动态授权保护，授权步具有生命周期。TRUBAC 模型如图 1 所示。

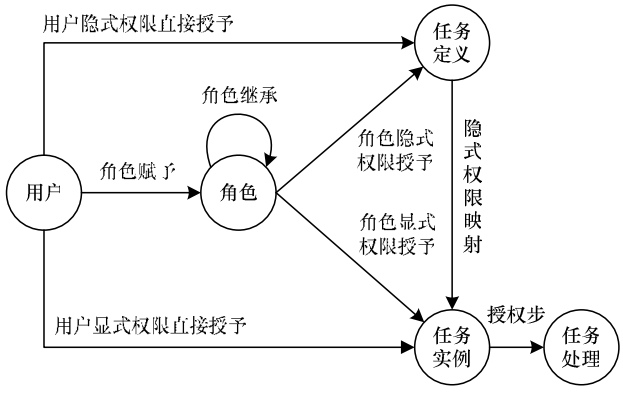

图 1 TRUBAC 模型

用户集：$U=\{u_1, u_2,\cdots, u_n\}$。

角色集：$R=\{r_1, r_2,\cdots, r_n\}$。

权限集：$P=\{p_1, p_2,\cdots, p_n\}$。

工作流实例集：$I=\{i_1, i_2,\cdots, i_n\}$。工作流的每一次执行形成一个工作流实例。

任务集：$T=\{t_1, t_2,\cdots, t_n\}$。任务集由工作流定义中的任务组成。

通过为工作流定义中的任务分配用户、角色权限，实现工作流中任务的隐式权限分配，相关的关系如下：

PTA(Permission-Task Assignment)：$PTA \subseteq P \times T$，任务权限分配关系。

TRA(Task-Role Assignment)：$TRA \subseteq T \times R$，任务角色分配关系。

URA(Use-Role Assignment)：$URA \subseteq U \times R$，用户角色分配关系。

PTRA(Permission-Task-Role Assignment)：$PTRA \subseteq P \times T \times R$，任务角色权限关系。

PTUA(Permission-Task-User Assignment)：$PTUA \subseteq P \times T \times U$，任务用户权限关系。

通过为工作流实例的任务分配用户、角色权限，实现工作流实例中任务显式权限授予，相关的关系如下：

IPTA(Instance-Permission-Task Assignment)：$IPTA \subseteq I \times P \times T$，工作流实例任务权限分配关系。

ITRA(Instance-Task-Role Assignment)：$ITRA \subseteq I \times T \times R$，工作流实例任务角色分配关系。

IURA(Instance-Use-Role Assignment)：$IURA \subseteq I \times U \times R$，工作流实例用户角色分配关系。

IPTRA(Instance-Permission-Task-Role assignment)：$IPTRA \subseteq I \times P \times T \times R$，工作流实例任务角色权限关系。

IPTUA(Instance-Permission-Task-User Assignment)：$IPTUA \subseteq I \times P \times T \times U$，工作流实例任务用户权限关系。

通过授权步，实现任务指派关系：

ITUPLTmS(Instance-Task-User-Permission-Lifecycle-Times Step)：$ITUPLTmS \subseteq I \times T \times U \times P \times L \times Tm$，工作流实例任务的用户权利指派，在授权步中，用户可以执行任务或改变任务的状态。P 是授权步所激活的权限，L 是授权步的存活期限，Tm 是授权步的执行次数。

3 TRUBAC 的任务用户指派策略

TRUBAC 模型根据工作流实例中任务的执行顺序，从 2 个途径查找能够执行下一个任务的用户：

步骤 1 根据 PTRA 和 IPTRA 选择有执行当前任务权限的角色和用户，然后通过 URA 和 IURA 查询到能够承担该角色的所有用户集 U_1。

步骤 2 根据 PTUA 和 IPTUA，选择被直接授权的用户集 U_2。

步骤 3 求 $U = U_1 \bigcup U_2$，从 U 中选择一个合适的用户

(选择策略见策略 1~策略 6)承担执行该任务。

现有模型很少考虑承担任务的用户怎么选择，如果选择不当，将会使工作流后续任务无法执行或效率低下。文献[10]提出了用户负载均衡等策略，在一定程度上提高了工作流的执行效率，但是没有利用任务执行的历史信息，也没有考虑执行任务的用户之间的差异，因此，在实际应用中存在不足。针对这一情况，根据工作流执行的特点，本文提出新的用户分配策略，在保证安全性的同时，提高工作流的执行效率。

策略 1 基于用户任务优先级的任务指派原则

为用户指定不同的任务优先级，在指派任务时，高优先级的用户优先被指派。当有多个用户都可以执行一个任务时，应该指派给优先级高的用户，这一策略保证了高优先级的用户由于被指派执行该任务的概率高，执行次数相对多，因此执行效率也会越来越高。如果相同优先级的用户，选择执行该任务历史次数多的用户，那么将该任务指派给他。

策略 2 为将来尽可能预留弹性资源的原则

如果在 2 个用户间进行选择，最好是选择那个相对只能够处理少量其他种类工作的资源。即当还有其他资源可以选择时，尽量让通用性好的资源空闲。

策略 3 用户职责分离原则

一些敏感的任务需要不同的用户执行，如发票处理流程中会计和出纳的职员必须不同。这通过授权步之间的分权依赖实现。

策略 4 基于繁忙因子的用户负载均衡原则

由于能力等因素，不同用户能同时执行的最大任务数可能会有差异，因此对于不同的人他们从事等量的任务可能繁忙程度和效率都不同。为了使工作流高效运转，在分配任务时，考虑用户的繁忙程度。

定义 1 用户正在执行的任务量与他能胜任的最大任务量的比值，称为该用户的繁忙因子。

某用户的繁忙因子 $b=n/N$，其中，n 是该用户当前的任务数；N 是该用户的最大任务数。需要指出的是，不同用户，N 值可能有所不同。

定义 2 在执行任务的过程中，每个用户的繁忙程度相对均衡，称为基于繁忙因子的用户负载均衡。

基于繁忙因子的任务分配原则：将该任务分配给能够执行该任务的各用户中繁忙因子最小的用户。

策略 5 任务分配给高效执行用户的原则

根据历史执行记录，可以计算出用户执行任务的效率，将任务分配给执行效率高的用户。

建立执行记录表($WfInstance$, $Task$, $User$, T_{begin}, T_{end})记录任务的执行情况，各属性的含义分别是：工作流实例，任务，用户，任务开始时间，任务结束时间。根据该表，计算 $T_{end}-T_{begin}$ 的差，可以知道用户完成任务的历史执行时间，不同用户通过比较这个时间，可以作为下一次相同任务选择用户的分配依据。

算法 将任务分配给执行任务效率最高的用户

步骤 1 查找任务 t 的用户。

步骤 2 在任务 t 的用户中查找空闲用户 u 的集合。即在执行记录表中没有关于 u 的相关记录，或者没有用户是 u 而结束时间非空的记录，这样的用户是空闲用户。如果有这样的 u，则执行步骤 3，否则执行步骤 4。

步骤 3 搜索执行记录表，首先查找 T_{end} 不为空的记录，计算步骤 1 中的每个用户针对该任务的最新执行时间 $T_{end}-T_{begin}$(同一用户执行一个任务，可能会执行多次，最后一次执行信息是用户执行该任务的最新记录，反映了用户最近执行该任务的效率状况，是任务分配的参考依据)。将任务分配给执行该任务需要时间最少的用户。

步骤 4 如果第 2 步中没有空闲的用户，则将任务分配给繁忙因子最低的用户，即查找执行记录表，判断步骤 1 中的用户中，哪个用户的相对任务数最少(繁忙因子最低)，将任务分配给他。

策略 6 满足用户任务分配依赖的原则

在工作流中有多个任务，实际工作中，人员可能存在不同的工作组，同一组内的人员分工协作完成一系列任务。对于这一系列任务，当将一个任务分配给某个组中的成员时，相关的一些任务也应该分配给同一组内的相关人员。因此，这些任务的用户分配存在依赖。

定义 3 如果将任务 t_1 分配给 u_1，则指定将 t_2 分配给 u_2，称任务 t_1、t_2 之间存在用户分配依赖。

建立用户分配依赖关系(T_1, U_1, T_2, U_2)，含义是：T_1、T_2 2 个任务的分配存在用户依赖，如果将 T_1 分配给 U_1 执行，则 T_2 分配给 U_2。利用用户分配依赖关系，可以实现用户以小组为单位合作完成多个任务。

满足用户任务分配依赖原则的任务分配算法：如果对任务 t_1 分配用户为 u_1，则查找用户分配依赖表，判断有无关于任务 t_1、u_1 的用户分配依赖记录，如果有，则将任务 t_2 分配给用户 u_2。如果 u_2 由于忙碌等原因不能将任务 t_2 分配给 u_2，则拒绝将 t_1 分配给 u_1。如果关于(u_1, t_1)的用户分配依赖有多个，则根据执行记录表，将 t_2 分配给执行效率高的用户。

4　结束语

本文提出了一种 TRUBAC 访问控制模型、用户和角色的双重授权机制以及基于工作流定义和工作流实例两级权限分配机制，实现灵活的任务权限分配，并通过授权步实现工作流任务的动态用户指派，同时提出若干用户分配策略，提高工作流的执行效率。此外，在执行记录表中记录了工作流任务执行历史的大量信息，利用数据挖掘技术，从中挖掘出更多的信息，进一步改进工作流的用户分配策略。

(下转第 81 页)