工作流中 TBAC 权限控制模型的扩展与 UML 描述

Extend Privilege Control Model on TBAC and Describe the Model Using UML in the Workflow

杨勇虎　刘振宇　（东北大学 东软信息学院 计算机科学与技术系 辽宁 大连 116023）

摘　要：本文首先分析了现实工作流授权所需要满足的一些特征，并提出基于任务的访问控制模型。然后通过分析，在此模型基础上进行了扩展，引入了上下文的概念，能够有效的管理和跟踪工作流的具体任务。针对模型的每一个过程进行了详细的设计和分析，给出了具体的实施办法，并且通过 UML 进行了有效的描述。

关键词：工作流 权限 任务 角色 上下文

1　引言

工作流管理系统是当前很多领域用来有效地管理业务处理过程以及实现办公自动化的一项关键技术。在构建一个安全的工作流系统时，权限的管理是一个非常重要的环节，其安全性和效率将会非常大地影响到整个信息系统，因此，一个好的权限管理模型往往是一个好的信息系统的开始。

从 20 世纪 70 年代开始，先后提出了很多的权限访问控制模型。1992 年 Ferraiolo 和 Kuhn 提出了基于角色的访问控制模型（Role – Based Access Control，RBAC），在权限和用户之间引入了角色的概念，很大程度上减轻了管理员的负担，提高了系统工作的效率。Thomas 和 Sandhu 在 1993 年提出了基于任务的访问控制（Task – Based Access Control，TBAC），该模型基于任务来定义权限，从任务的角度来建立安全模型和实现安全机制，在任务处理的过程中提供动态实时的安全管理。在这个模型中，权限分配给任务，任务再分配给角色，在任务处理过程中提供动态的权限管理，非常适用工作流状态下的权限管理。本文在 TBAC 的基础上进行扩展，建立了一个基于 TBAC 的 UML 模型，引入了上下文的概念。有效的实现了权限的动态分配和跟踪。

2　TBAC 的模型概述

新一个合适的工作流访问控制具有以下特征：

（1）一个用户只能在某个周期时间段或某个持续时间段中具有某个或某几个角色的权限，可以称之为时间约束。

（2）操作权限只有任务开始执行并处于相应的状态时才能授予用户，并且任务一旦离开，该状态操作权限应该就被收回，可以称之为状态约束。

（3）一个用户可能在同一段时间执行不同的任务，并且在不同的任务中拥有不同的权限，这些任务相互依赖并以一种协调的方式执行，可以称之为多任务约束。

（4）一个用户在执行任务的具体实例时只能对该任务所允许操作的对象或是对象的某个部分进行相应的操作。可以称之为最小特权约束。

模型针对工作流的特点，增加了任务这一项，任务具有权限，即根据具体的执行任务要求和权限约束，任务具有相应的权限，不同的任务拥有不同的权限。在动态授权中权限随着任务的流动而变动，根据工作流要求和任务约束，任务分配给合适的角色，只有当任务到达角色，角色需要执行任务时，角色才被赋予了执行任务的权限和操作工作流中文件的权

限。即角色的权限与任务紧密相连，根据任务进行动态分配和回收。

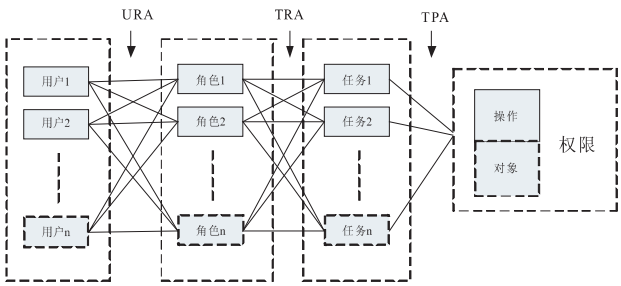

图1　基于任务的角色访问模型结构图

3　基本定义

定义1. 用户集(U)：系统中所有对资源(客体)进行访问和操作的对立主体。也可以认为是任务的参与者。用U表示全体用户的集合，u表示一个用户，$u \in U$。

定义2. 角色集(R)：在企业环境中，角色是一个组织部门中的行政职位，也可以是一个技术角色，或者是某个业务流程的工种。用R表示全体角色的集合，r表示一个角色，$r \in R$

定义3. 任务集(T)：是指系统中的所有任务，是在企业日常工作中的一个业务流程，是企业完成某项工作目标的基本单位，也可以是一个项目的一个操作部分。用T表示各个任务的集合，用t表示具体的一个任务，$t \in T$。

定义4. 权限集(P)：权限是对计算机系统中被保护的数据和资源的访问许可。主要包括对象的访问权限和数据操作的访问权限。用P表示权限集合，用p表示具体某个权限。$p \in P$

定义5. 对象访问控制(OVC)：用一个二元组来表示(控制对象、访问类型)，其中控制对象表示系统中一切需要进行访问控制的资源。我们将引入一套完整的资源表示方法来对系统中出现的各类资源进行定义和引用。访问类型是指对于相应的受控对象的访问控制，如：读取、修改、删除等。

定义6. 数据访问控制(DVC)：如果不对数据访问加以控制，系统的安全性是得不到保证的，容易发生数据泄密事件。所以在权限中必须对对象可访问的数据进行按不同的等级给予加密保护。我们同样用一个二元组来表示：(控制对象，谓词)。权限最终可以组合为：(控制对象，访问类型，谓词)

定义7. 任务权限配置(TPA)：权限和任务的关联，多对多。$TPA \subset T \times P$

定义8. 任务角色分配(TRA)：角色与任务的关联，多对多。$TRA \subset T \times R$

定义9. 角色用户授权(URA)：用户与角色的关联关系，多对多。$URA \subset U \times P$

4　扩展模型的具体实现与 UML 描述

4.1　上下文环境

对象的访问权限控制并不是静止不变的，而是随着执行任务的上下文环境而变化。具体来说，首先，在工作流环境中，每一步对数据的处理都与以前的处理有关，也就是和某一个任务的状态有关。其次，同一工作流的不同任务要实行不同的访问控制策略。另外，同一个任务在其即将执行时，才对用户授权，当任务执行完毕后就撤销用户权限，这样保证权限在需要的时候才能用到。因此，TBAC是一种上下文相关的访问控制模型。于是，我们引入上下文环境(Context)，它是一个抽象的概念，是指用户、任务、角色和权限的综合体，为某一个具体的工作流协作活动提供协作的环境，可以用来管理和跟踪具体的任务。

工作流管理者(WorkflowManager)对上下文环境进行管理，上下文的成员可以由不同的任务构成，每一个任务可以通过一个登记簿(RegisterBook)连接上下文。我们可以通过上下文对任务进行记录和跟踪。

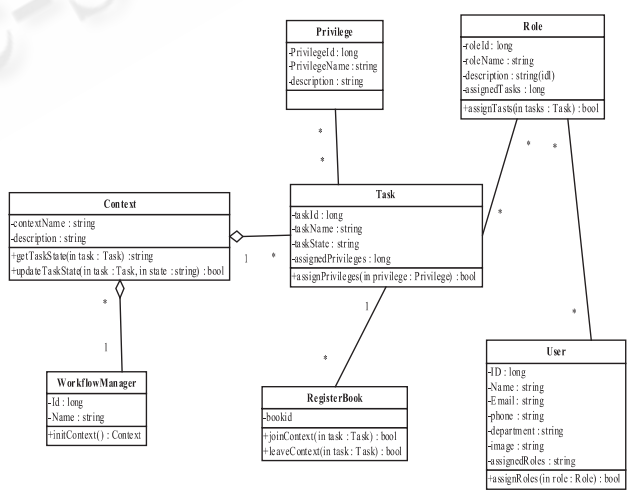

图2　上下文环境下 TBAC 类图

4.2 权限定义

权限的定义和管理是系统权限模型的最重要最基本的步骤,如果没有定义和管理好权限,也无法进行合理有效的权限配置。我们通过系统分析和设计,定义合适的对象,对系统元素进行细化,具体到每一个界面元素以及菜单元素。然后针对每一个对象,采用三元组符号标记法,定义其相应的权限。

4.2.1 对象定义

对象是指系统中各种功能模块、数据、界面元素(包括导航栏、菜单、按钮等各种界面上能控制的控件)等,它们是主体能访问的各种对象。在设计中,提供一个对象定义工具事先定义好系统要控制的对象。主要包括:功能模块定义,界面元素控制,数据信息控制。

一个系统的功能模块,主要是业务功能模块,即:用户完成各自不同的业务功能,也是安全管理的重点保护对象。我们必须根据用户需要完成的工作,配置用户业务功能菜单。

除了功能菜单或导航栏的控制外,对界面元素也应进行定义,大部分的界面元素均包含有相关的业务功能操作。

业务功能模块的大部分界面元素是显示和操作数据内容的基础,也是用户对读取数据和操作数据的主要途径,为了数据信息的安全有必要对界面元素的操作数据予以采取安全保密措施。这就需要对这些界面元素定义相关的数据约束条件。

对象的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。对象定义的流程如下图 3 所示:

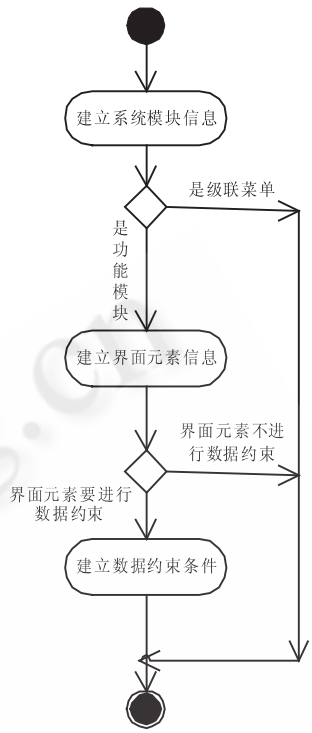

图 3　对象定义活动图

4.2.2 权限定义

在定义好系统对象的前提下,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,a),其中 o 表示访问对象;t 表示访问类型;a 表示谓词。即:在谓词 a 为真时对于对象 o 可进行 t 类型的访问。

权限定义是系统安全管理的基础步骤之一,只有给各种对象定义好访问的权限,才能给任务配置权限。定义权限的流程如图 4 所示:

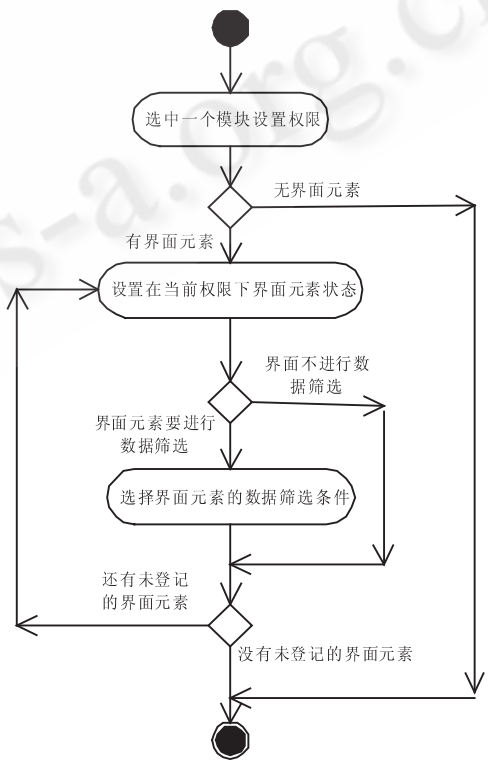

图 4　权限定义活动图

4.3 任务定义与权限配置

4.3.1 任务定义

任务由工作流制定者定义,然后通过登记簿注册到上下文环境,成为上下文的成员,在上下文中管理任务。根据任务在工作流中的生命活动周期,主要可以分为 初始态、活动态、挂起态、取消态和终止态,如图 5 所示。

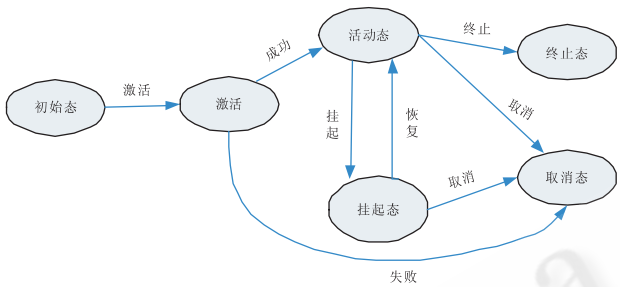

图 5 任务的状态图

4.3.2 权限配置

根据工作流动态的特点,具体到某一个任务,将权限分配给任务。即根据具体的执行任务要求和权限约束,任务具有相应的权限,不同的任务拥有不同的权限。权限的配置分为动态和静态授权。静态授权无需执行工作流就可以指定权限,此权限相对固定。动态授权是在工作流执行过程中进行的,根据任务的状态,动态进行权限的分配和回收。

4.4 角色定义与用户授权

4.4.1 角色定义

在设计中,提供角色定义工具允许系统根据任务对用户的需要(职权、职位以及分担的权力和责任)定义相应的角色。角色之间有相应的继承关系,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。角色继承关系自然的反映了一个组织内部权力和责任的关系,为方便权限管理提供帮助。角色继承关系提供了对已有角色的扩充和分类的手段,使定义新的角色可以在已有的角色基础上进行,扩充就是通过增加父角色的权限去定义子角色,分类通过不同子角色继承同一个父角色来体现。另外,还允许多继承,即一个角色继承多个父角色,多继承体现对角色的综合能力。角色定义流程如下图 6 所示:

4.4.2 用户授权

用户是系统的最终使用者,因此必须建立用户的鉴别机构,登记用户的身份信息。针对某项任务,定义一群用户的集合,即用户集。当用户被加入用户集时,自动对用户的所在用户组拥有的角色进行委派。

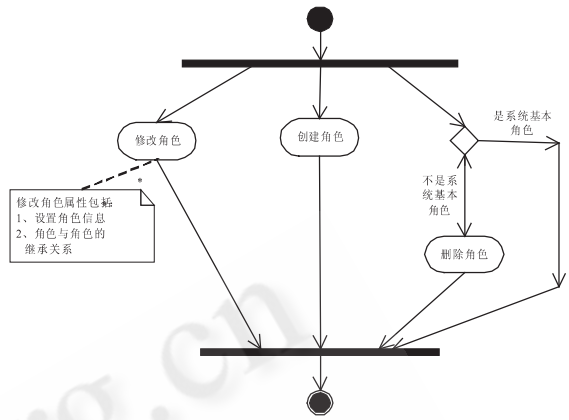

图 6 角色定义活动图

5 结束语

本文通过分析现实工作流授权所需要满足的一些特征,以及基于任务的角色权限管理模型。在此模型基础上进行了扩展,引入了上下文的概念,能够有效的管理和跟踪工作流的具体任务。并针对模型的每一个过程进行了设计和分析,给出了具体的实施办法,并且通过 UML 进行了有效的描述,该模型已经在实际的项目中得到了运用,效果良好。

参考文献

1 Sejong Oh, Seog Park. Task – based access control model. Information Systems, 2002, 28(2003) 533 – 562.
2 王振江,就强. 基于 RBAC 的扩展访问控制模型. 计算机工程与应用,2005,35 23 – 25.
3 邢光林,洪帆. 基于任务状态能力的工作流授权模型. 微型机与应用,2005,(3) 51 – 54.
4 金琼峥,杨树堂,等. 基于 T – RBAC 的企业权限管理方法. 计算机工程,2004,30(19) 93 – 95.
5 洪帆,杜小勇. 办公自动化系统中基于任务的访问控制. 华中科技大学学报,2001,29(3) 6 – 8.
6 黄国言,孙惠学. 共享工作空间下的协作模型研究. 中国机械工程,2006,17(20) 2144 – 2147.