ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.9, September 2007, pp.2295−2305
DOI: 10.1360/jos182295
© 2007 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

协同环境下CAD模型的多层次动态安全访问控制*

方萃浩[1], 叶修梓[1,2+], 彭 维[1], 张 引[1]

[1](浙江大学 计算机科学与技术学院,浙江 杭州 310027)

[2](浙江大学 CAD&CG国家重点实验室,浙江 杭州 310027)

Multi-Level and Dynamic Security Access Control for CAD Models in Collaborative Environment

FANG Cui-Hao[1], YE Xiu-Zi[1,2+], PENG Wei[1], ZHANG Yin[1]

[1](College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

[2](State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: Phn: +86-571-87953039, Fax: +86-571-87952690, E-mail: yxz@zju.edu.cn, http://www.zju.edu.cn

Fang CH, Ye XZ, Peng W, Zhang Y. Multi-Level and dynamic security access control for CAD models in collaborative environment. *Journal of Software*, 2007,18(9):2295−2305. http://www.jos.org.cn/1000-9825/18/2295.htm

Abstract: In this paper, a multi-level and dynamic security access control (MLDAC) model is proposed for CAD models in collaborative environment. A multi-level privilege model is developed to simplify the process of permission definition and assignment for enriching the expression ability and helping to realize multi-grained access control. The dependent relation of permission and the permission state migration are brought into MLDAC for dynamic authorization management based on the basic theory of workflow. Based on the practice, MLDAC model is more efficient to control collaborative operations. It meets the characters of design tasks, i.e. divisible, dependent and interactive.

Key words: access control; workflow; information security; collaborative design; CAD

摘 要: 提出一个专门针对协同环境下CAD模型的多层次动态的安全访问控制(multi-level and dynamic security access control,简称MLDAC)模型.该模型利用一种多层次的权限模型,以简化权限定义及其分配过程,丰富了权限表达能力,实现了产品模型的多粒度访问控制.通过参照工作流的基本理念,引入权限的依赖关系及权限状态迁移概念,实现了权限的动态授权管理.通过实践证明,MLDAC模型可以对协同设计操作进行更加有效的控制,符合设计任务间的分工性、依赖性和交互性的特点.

关键词: 访问控制;工作流;信息安全;协同设计;CAD

中图法分类号: TP391 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60473106, 60333010 (国家自然科学基金); the National Research Foundation for the Doctoral Program of Ministry of Education of China under Grant No.20030335064 (国家教育部博士点基金)

Received 2006-04-18; Accepted 2006-07-10

© 中国科学院软件研究所 http://www.jos.org.cn

随着网络技术的迅速发展和不断应用,信息安全(特别是协同环境中的模型安全)成为人们研究的重点问题之一.当前,对于信息安全的研究包含许多方面,如数据可访性、数据机密性、数据完整性、数据抗抵赖性、身份鉴别、访问控制等.其中,访问控制是信息安全保障机制的核心内容,它是实现数据保密性和完整性机制的主要手段.访问控制的目的是控制系统资源,使其只允许被授权的用户、程序或过程所访问,从而阻止一切破坏系统安全的活动.在协同 CAD 环境中,安全问题也越来越得到人们的重视,其中,产品模型的访问控制对于协同设计来说更具有重大意义,它对于保证产品模型的可用性、完整性、机密性和设计过程的有效性起着关键作用.

由于CAD协同设计环境的特殊性,其用户行为具有不可预知性,其设计活动一般具有一定的相关性及时序性.同时,协同设计过程中的共享数据不是一般的数据模型,而是含有设计意图并具有层次相关性的产品模型,它的访问控制也往往涉及到并发性和一致性控制问题.然而,现有传统的访问控制模型都是从系统的角度(控制环境是静态的)出发保护共享资源.它没有将执行操作所处的环境考虑在内,容易造成安全隐患,不能完全适应于协同环境的特殊需求;并且,其保护的数据模型相对简单,不能适用于复杂CAD产品模型的访问控制.为此,本文提出了一种协同环境下 CAD 模型的多层次动态访问控制(multi-level and dynamic access control,简称MLDAC)模型.本文第 1 节简要介绍相关研究工作.第 2 节详细描述 MLDAC 模型.第 3 节阐述 MLDAC 模型的访问控制策略和机理.第 4 节给出 MLDAC 的一个应用实例.第 5 节是结论.

1 相关研究工作

访问控制技术源于 20 世纪 70 年代,最初的需求是保护大型计算机系统数据集的安全[1].在随后的 30 多年中,先后出现了多种访问控制策略,其中最为常用的访问控制策略有:自主访问控制(discretionary access control,简称DAC)、强制访问控制(mandatory access control,简称MAC)和基于角色的访问控制(role-based access control,简称RBAC).

CAD协同环境中的访问控制研究相对于其他领域起步较晚,还处于起步阶段.初期关于产品模型的安全性问题主要是通过数字水印途径来解决[2−5].然而,数字水印技术只能保证模型的完整性,不具有灵活的控制机制,无法满足协同设计的需求.因此,学者们提出了一些专门针对CAD协同设计环境的安全访问控制模型.

第一个重要的协同环境中访问控制模型是由Shen等人提出的编辑模型(editing model)[6].该模型在访问矩阵基础之上进行扩展,增加了许多适应于协同环境的特殊处理,其中有细粒度的权限设置、群组管理和支持负权限.该模型实现过程比较复杂,且由于基于访问矩阵,不适合大规模的协同环境.Bullock等人提出了一种基于空间划分的协同访问控制模型——Space模型[7].该模型首先引入边界(boundary)和访问图(access graph)两个概念,通过边界首先将大规模的协同环境划分成若干细小的可控区域,每一个可控区域应用一个相应层次的访问控制策略,相同的可控区域应用相同层次的访问控制策略.该模型非常适合于多领域大规模的协同环境.但由于并不是所有的协同环境都可以进行空间划分,可见其应用领域具有较大的局限性.

协同装配设计领域中一个最相关的工作是Shyamsundar和Gadh提出的集成控制方案[8].他们将每一个部件(即子装配体)分隔成若干个接口特征(interface feature),并用一个包裹(envelope)近似表示.包裹可以是零部件的凸包围空间、包围盒(球),或是由零部件外表面围成的包围空间.但他们的工作只是一种简单的信息隐藏技术,没有详细的访问控制机制.van der Hoeven等人提出了一个具有访问控制机制的CAD框架,但其实现仍然只停留在项目层次的角色定义上,无法针对CAD模型的各个部分进行灵活的访问控制[9].

Cera等人提出一种三维模型的安全访问控制技术,称为Role-Based Viewing[10].它基于MAC 和RBAC进行扩展.首先依据设计意图将产品模型分解成若干安全特征(security feature),然后对角色和安全特征进行安全层次标识,用访问矩阵记录不同安全层次角色对不同安全层次特征的访问权限,对于不同的设计者,根据其设计活动的目的,利用MAC策略赋予设计者不同的访问权限,并根据设计者权限提供相应层次的视图模型.由于安全特征的划分、角色及安全特征的安全层次标识等工作必须预先定义,因此该模型不支持动态的权限调整,主要应用于最终产品模型的多视图协同观察,因而不能很好地支持产品模型的协同设计.

随着网络和分布式计算的发展及组织任务的进一步自动化,工作流技术成为一个研究热点.工作流是复杂

© 中国科学院软件研究所 http://www.jos.org.cn

多任务协同建模的一种有效方法,它是为完成某一目标而由多个相关任务(活动)构成的业务流程[11].任务是工作流的最小可执行单元,任务间可能存在复杂的依赖关系.工作流管理系统(WfMS)一般通过任务授权实现资源的访问控制和处理过程的自动化,任务授权一般用三元组$Wf=(T,A,C)$表示[12],其中,$T=\{t_i\}$为任务偏序集,$t_i<t_j$表示t_i先于t_j执行;A是任务授权规则;C是执行任务约束集;若用U表示用户集,则有$A\subseteq T\times U$.目前,绝大多数WfMS把RBAC作为其访问控制模型.由于工作流应用系统环境的复杂多变性,单纯RBAC往往无法满足需求,为此提出了许多新型的面向工作流系统的访问控制模型[12−17].其中,基于任务的访问控制(task-based access control,简称TBAC)从任务(活动)的角度建立安全模型和实现安全机制,通过授权生命期约束、授权依赖关系约束和授权步状态变迁等,在任务处理过程中提供动态实时的安全管理[15].许峰等人提出的面向服务的访问控制模型与TBAC类似,也是通过授权依赖关系和授权迁移实现动态授权.此外,他还引入了服务及服务状态变化等概念以加大对权限的管理力度[17].

如前所述,CAD 协同环境也需要有良好的安全访问机制,以适应产品模型访问控制的灵活性、多层次性和可扩展性发展趋势.由于通用的访问控制不能满足协同环境的特殊需求,同时,现有专门针对 CAD 领域的访问控制机制都比较简单,只适用于简单场景的访问控制.为此,本文在层次RBAC模型基础上,通过借鉴现有面向工作流系统的访问控制模型的优点,结合 CAD 协同环境及 CAD 产品模型的特殊性,提出一个既能满足动态授权管理,又能很好地适应三维 CAD 产品协同设计的 MLDAC 模型.

2 MLDAC 模型

2.1 模型概述

传统的访问控制模型(如访问矩阵模型)是直接在用户和数据之间进行权限控制,如图 1(a)所示.这种直接控制模型没有涉及到如何提供协同权限的说明和控制,需要对每一个用户和数据进行权限授予,不支持群组授权.为此提出了基于角色的访问控制,它在用户和资源之间增加了角色这一层,访问权限不是直接与用户进行关联,而是通过角色间接关联,通过改变用户的角色来改变用户的访问权限,如图 1(b)所示.

(a) Traditional direct access control
(a) 传统直接访问控制

(b) Role-Based access control
(b) 基于角色访问控制

Fig.1 Traditional direct access control and RBAC

图 1 传统直接访问控制与基于角色访问控制

为了满足 CAD 协同环境需求,我们提出了一种多层次动态的安全访问控制(MLDAC)模型.该模型基于层次 RBAC 模型,其总体结构如图 2 所示.

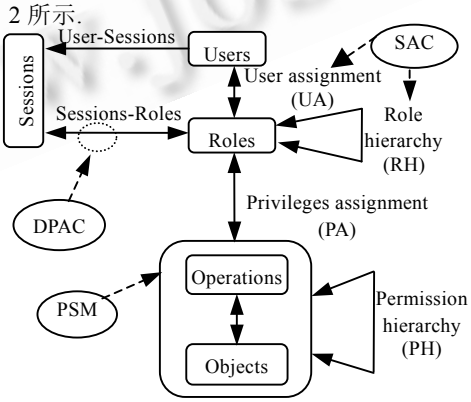


Fig.2 Multi-Level and dynamic security access control model

图 2 多层次动态的安全访问控制模型

© 中国科学院软件研究所 http://www.jos.org.cn

相对于传统的 RBAC 模型,MLDAC 模型增加了角色分层(role hierarchy,简称 RH)和权限分层(permission hierarchy,简称 PH)管理.角色分层简化了权限表示及角色授权过程;权限分层迎合了 CAD 产品的多层次结构特点,增加了角色授权的灵活性.由于不同的设计操作之间存在相互依赖关系,该模型还提供了静态授权约束(static authorization constraint,简称 SAC)管理和动态权限激活约束(dynamic permission activation constraint,简称 DPAC)管理,同时还引入了权限状态迁移(permission state migration,简称 PSM),以更加符合产品设计过程的流程化、交互性的特点.

2.2 相关定义

MLDAC 模型主要包括对象、操作、权限、角色、设计者和会话等要素,下面是各个要素的定义及说明.

定义 1(对象,数据集). 产品模型的所有数据构成了协同设计的操作数据集,用 D 表示.对象是指产品模型数据的一个子集,记为 ob,且$\forall ob,ob\subseteq D$.在层次建模中,ob 可以是 CAD 产品模型的拓扑面、体特征、零件体或装配体等.

定义 2(操作集,操作子集). 操作集(记为 OP)是一个有限集,其中的每个元素都表示一种可以对对象实施的操作 op.如果 $o\subseteq OP$,则称 o 为一个操作子集.

定义 3(访问权限,访问权限集). 访问权限集是集合$D\times 2^{OP}$的子集,记为P.P中的每个元素表示一种权限,记为$p(ob,o)$.每一个权限以对象为单位进行授权,其直观含义是,若$o\subseteq OP$且$ob\subseteq D$,则访问权限$p(ob,o)\in P$,表示对对象ob可以执行操作子集o中的各项操作.

定义 4(角色集,角色). 角色集是由访问权限集的一些子集构成的集合,记为R,即$R\subseteq 2^P$.角色集的每一个元素表示一种角色r,r是一组权限的集合,即$\forall r\in R$,有$r\subseteq P$.

定义 5(角色继承). $\forall r_1,r_2\in R$,如果$r_1\subseteq r_2$,则称r_2继承r_1,即满足继承关系,用$r_1\rightarrow r_2$表示,其中r_1称为父角色,r_2为子角色.角色继承关系是一种偏序(partial orders)关系,具有自反性(reflexive)、传递性(transitive)和反对称性(anti-symmetric).

角色的结构化分层反映了一个组织的授权与责任分离的自然方式,它提供了对已有角色进行扩充和分类的手段,可以在已有角色的基础上定义新的角色.其中,扩充是通过增加父角色的权限去定义子角色,分类是通过不同子角色继承同一父角色来体现.另外还允许多重继承,即一个角色继承于多个父角色,多重继承体现了对角色权限的综合能力.

设角色r继承于角色r_0,则r的权限为r_0的权限与扩充权限之和,即$r=r_0\cup \Delta r$.如果r继承了多个角色$r_1,r_2,\ldots, r_n$,则$r=r_1\cup r_2\cup\ldots\cup r_n\cup\Delta r$.

定义 6(用户). 这是指协同环境中被赋予一定角色集具有相应权限从事协同活动的人员,协同环境的所有用户构成用户集 U.

在协同设计环境中,用户为在协同会话(session)中从事产品开发的设计者,设计者通过名字加以区别;用户一般隶属于某个工作组,通过群组管理;同时,用户关联一个角色集使其具有相应的访问权限,由此,一个用户可用四元组表示,即u=($designer\_name$,$session$,$team$,$roles$).其中,$designer_name$为设计者的名字,$session$为协同会话,$team$为所属的小组,$roles$为所关联角色集.设计者可能在多个不同工作组从事不同角色的工作,这时,这个设计者将分别代表不同的用户.如表 1 所示,设计者Jack在不同会话($session_1$和$session_2$)和不同工作组($team_1$,$team_2$,$team_3$)中从事不同的角色集,并分别代表不同的用户(u_1,u_2,u_3).

Table 1 The definition structure of users

表 1 用户的定义结构

User	Designer	Session	Team	Roles
u_1	Jack	$session_1$	$team_1$	r_1,r_2,r_3
u_2	Jack	$session_2$	$team_2$	r_2,r_4
u_3	Jack	$session_1$	$team_3$	r_1,r_5

然而,同一 $session$ 中,每一个设计者只能以其中的一个用户身份进行工作,这可以通过 User-Sessions(US)管理实现.工作组 $team$ 为一种特殊的用户,它用于存储用户共有属性或方法.同时,$team$ 也可以关联一个角色集,表

© 中国科学院软件研究所 http://www.jos.org.cn

示该工作组所有成员具有的共同角色集.因此,用户的最终角色集为工作组的角色集和用户的定义角色集之和,即 $roles(u)=team(u).roles+u.roles$.

定义 7(权限状态,permission state). 这是权限在执行过程中可能经历的状态,令 $p\in P$,则该权限的状态为 $p\langle state\rangle$.在 MLDAC 模型中定义如下 5 种权限状态:

(1) 睡眠状态(dormant),表示一个权限处于未被激活状态;

(2) 就绪状态(ready),表示一个权限的合法性条件已经得到满足,完成了运行前的基本准备工作;

(3) 挂起状态(hold),表示权限激活时由于等待操作资源或因权限依赖关系约束而被暂停运行,或权限在运行过程中被系统强制暂停运行;

(4) 运行状态(running),表示一个权限被成功激活后正在运行;

(5) 完成状态(accomplished),表示一个权限被正确执行完毕.

图3为权限状态迁移(permission state migration,简称PSM)关系图,当用户请求执行某一个权限时,系统首先根据用户身份和所具有的角色进行合法性检测,如果权限请求合法,则使权限进入"就绪状态".如果此时该权限可用,那么权限将被激活并进入"运行状态";如果不可用,则进入"挂起状态",直到激活条件满足时恢复权限执行并进入"运行状态".如果在激活或执行时出现异常,则终止权限执行并初始化权限状态为"睡眠状态".

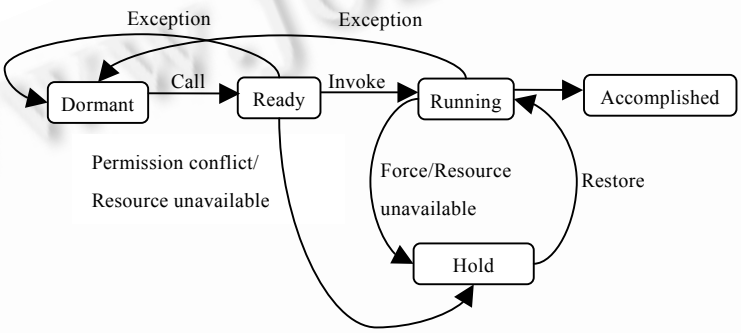


Fig.3 The relationship of permission state migration

图 3 权限状态迁移(PSM)关系图

定义 8(权限关系,permission relation). 这是指不同权限之间的依赖关系.根据权限执行的时间顺序,可以分为如下3类关系:

(1) 同步关系,是指不同权限必须同步执行,即对于任意两个权限p_1和p_2,在执行p_1时,要求p_2也被同时执行,反之亦然,记为$p_1\leftrightarrow p_2$.同步关系为一种等价关系,满足自反性、传递性、对称性,其中,若$p_1\leftrightarrow p_2,p_2\leftrightarrow p_3$,则$p_1\leftrightarrow p_3$.

(2) 顺序关系,是指权限执行存在严格的先后关系,即对于任意两个权限p_1和p_2,只有当p_1执行完成后,才能允许执行p_2,记为$p_1\rightarrow p_2$.顺序关系满足非自反性、传递性、非对称性,其中,若$p_1\rightarrow p_2,p_2\rightarrow p_3$,则$p_1\rightarrow p_3$.

(3) 互斥关系,是指权限执行存在排斥关系,即对于任意两个权限p_1和p_2,在执行p_1时,要求p_2不被执行,反之亦然,记为$p_1\leftrightarrow\neg p_2$或$p_2\leftrightarrow\neg p_1$.互斥关系满足非自反性、非传递性、对称性.

容易看出,互斥关系为静态关系,它与权限的当前执行状态无关,如在协同装配体设计过程中,管理员根据任务分工,将对零件A的操作与对零件B的操作定义为互斥关系,即$op(A)_i\leftrightarrow\neg op(B)_j$,以使一个人只能从事其中某一个零件的设计操作.同步关系和顺序关系为动态关系,它取决于权限所处的执行状态,如后文的图 4(a)所示,零件$part_2$中特征$GearTeeth_{20}$的创建必须在特征$GearBase_{21}$的创建之后,即

$$create(GearTeeth_{20})\rightarrow create(GearBase_{21}).$$

定义 9(角色冲突,role conflict). 这是指$\forall r_1,r_2\in R$,如果$\exists p_1\in r_1,p_2\in r_2$且$p_1\leftrightarrow\neg p_2$,则称角色$r_1$与角色$r_2$冲突,记为$r_1\leftrightarrow\neg r_2$.角色冲突关系满足非自反性、非传递性、对称性.

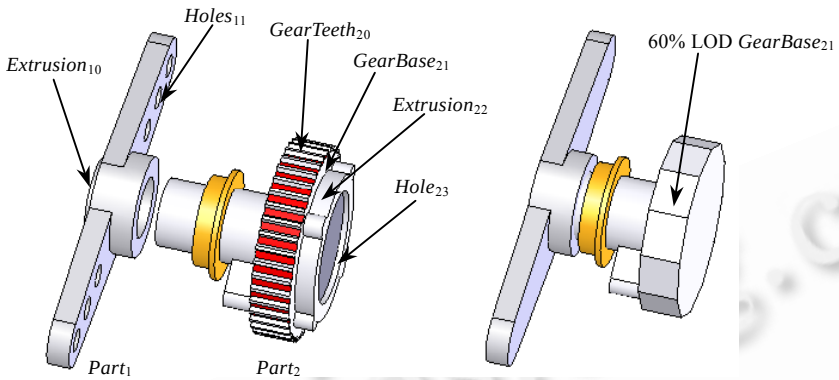
定义 10(静态授权约束,SAC). 这是指在角色授权、角色扩充/综合以及用户与角色的关联过程中必须考虑权限的互斥关系,以防止角色中含有互斥的权限和用户关联相互冲突的角色.如图 2 所示,SAC 在用户角色分配

© 中国科学院软件研究所 http://www.jos.org.cn

(user assignment,简称 UA)过程和角色分层结构设计中起静态约束作用.SAC 有两个约束准则:

(1) 角色权限不互斥:$\forall r\in R,\forall p_i,p_j\in r,p_i\neq p_j$,满足$p_i$和$p_j$不互斥.

(2) 用户角色不冲突:$\forall u\in U,\forall r_i,r_j\in u,r_i\neq r_j$,满足$r_i$和$r_j$不冲突.



(a) Original assembly model (b) Simplified assembly model

(a) 原始装配体模型 (b) 简化后的装配体模型

Fig.4 Assembly simplification based on role's privilege

图 4 基于角色权限的装配体模型简化

定义 11(动态权限激活约束,DPAC). 它是指在协同工作中,权限间的依赖关系使得角色的权限激活必须具备一定的条件.由权限关系定义可知,在权限集P中,权限$p_i\in P$的激活条件为p_i与所有其他权限p_j之间依赖关系的集合,如式(1)所示.

$$activate(p_i)=\bigcup_{p_j\in P,p_i\neq p_j} relation(p_i,p_j) \tag{1}$$

用户在会话过程中行使权限时,只有满足激活条件的权限才可以被执行.角色所关联的权限集称之为角色的授予权限(authorized permissions,简称 AP),满足激活条件的权限集称之为角色的可执行权限(enable permissions,简称EP).因此,角色的可执行权限是角色的授予权限的子集[17],如图 5 所示.

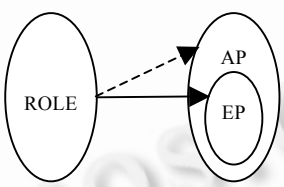


Fig.5 The activation of role permissions of users

图 5 用户的角色权限激活

2.3 多层次权限模型

由访问权限的定义可知,一般的访问权限是对模型对象的一类操作 $p(ob,o)$.为了实现动态的权限管理,第 2.2 节提出了权限状态概念,因此,访问权限可以表示成三元组 $p(ob,o,state)$.

然而,考虑到 CAD 产品模型的层次性特点,MLDAC 采用一种多层次的权限模型.即将访问权限分为两个层次,上层为零件层权限(part-level permission,简称 PP),下层为特征层权限(feature-level permission,简称 FP),每一层权限具有不同的定义和属性.也许某些应用可能认为部件层权限(component-level permission)是必要的,但我们认为,如此设计会使权限定义和授权变得复杂和不便.因此,我们仍然把零件作为最大的权限访问单位,这可以满足大部分应用需求.权限状态 $state$ 是任何层次权限都具有的一个属性,它仅影响权限的激活.为书写方便,在以下各层权限定义中省略了权限状态属性项.

© 中国科学院软件研究所 http://www.jos.org.cn

2.3.1 零件层权限

零件层权限可以定义为一个三元组(*part*,*mode*,*value*),其中,*part* 表示权限操作的零件体对象,*mode* 表示访问控制方式,这里只定义读取(READ)和编辑(EDIT)两种模式,*value* 为表征访问程度的百分比数值.

在协同设计环境中,为了减少模型数据传输量并保证协同设计活动的安全性,往往需要根据用户的不同权限提供不同细节层次(level of detail,简称LOD)的模型[18],而LOD可用一个百分数值来表示.为了更好地阐述问题,以下通过将权限*value*值表征为访问模型的LOD进行举例说明.在实际应用中,权限*value*值当然也可以用于表征其他任何有意义的数值.

2.3.2 特征层权限

类似地,特征层权限也可定义为三元组(*feature*,*mode*,*value*),其中,*mode* 和 *value* 与零件层权限定义中的意义相同,而 *feature* 表示权限操作的特征对象.

2.3.3 权限集的权限计算

为了使上述定义可以得到正确的权限,我们规定以下3条规则:

规则 1. 在分层权限定义中,*value* 的取值范围为 0%~100%.

规则 2. 在产品设计中,编辑操作往往需要模型的所有细节,即提供 100% LOD 模型;否则,可以不提供该模型的任何细节层次,即提供 0% LOD 模型.因此,为符合设计的自然语义,编辑访问模式下权限的 *value* 值只能设为 0%或 100%,不允许存在中间值状态,即 iff $p.mode==EDIT$, then set $p.value$=0% or $p.value$=100%.

规则 3. 在同一个权限集PS中,针对相同操作对象obj(*part*/*feature*)和相同访问模式*mode*(READ/EDIT)的权限只允许存在一个.即iff $p_1.obj==p_2.obj$,$p_1.mode==p_2.mode$, and $p_1\in PS$, then $p_2\notin PS$.

权限集的权限为其所含有的零件层权限和特征层权限的逻辑和.对于给定权限集PS_0,$PS_0(feature_0,m_0)$表示PS_0关于特征$feature_0$在访问模式m_0下的访问权限,若令$part_0$为特征$feature_0$所属的零件,则$PS_0(feature_0,m_0)$的计算过程如下:

set $PS_0(feature_0,m_0)=(feature_0,m_0,0\%)$
for each part-level permission PP_i in permission set PS_0
  if $PP_i.part==part_0$ and $PP_i.mode==m_0$ then
    $PS_0(feature_0,m_0).value=PP_i.value$
    Break //最多只允许一个匹配的零件层权限(规则 2).
next PP_i
for each feature-level permission FP_j in permission set PS_0
  if $FP_j.feature==feature_0$ and $FP_j.mode==m_0$ then
    $PS_0(feature_0,m_0).value=PF_j.value$
    Break //最多只允许一个匹配的特征层权限(规则 2).
next FP_j

如有一个简单产品由两个零件组成,$PD=\{part_1,part_2\}$.每一个零件由若干特征构成,$part_1=\{extrusion_{10},holes_{11},\ldots\}$,$part_2=\{gearteeth_{20},gearbase_{21},extursion_{22},hole_{23},\ldots\}$,如图 4(a)所示.如欲定义一个角色$r$,使其只能读取 $part_1$ 中除 $extrusion_{10}$,$holes_{11}$ 外的所有特征的 100% LOD,只能编辑 $part_2$ 中除 $gearteeth_{20}$,$gearbase_{21}$,$extrusion_{22}$,$hole_{23}$外的所有特征,可以读取$part_2$中$gearbase_{21}$特征的 60% LOD,则角色r权限集$PS_1(r)$定义为:

$PP_1=(part_1,\text{READ},100\%)$ $FP_1=(extrusion_{10},\text{READ},0\%)$ $FP_2=(holes_{11},\text{READ},0\%)$
$PP_2=(part_2,\text{EDIT},100\%)$ $FP_3=(gearteeth_{20},\text{EDIT},0\%)$ $FP_4=(gearebase_{21},\text{EDIT},0\%)$
$FP_5=(extrusion_{22},\text{EDIT},0\%)$ $FP_6=(hole_{23},\text{EDIT},0\%)$
$PP_3=(part_2,\text{READ},0\%)$ $FP_7=(gearbase_{21},\text{READ},60\%)$

$PS_1(r)=\{PP_1,FP_1,FP_2,PP_2,FP_3,FP_4,FP_5,FP_6,PP_3,FP_7\}$

在协同设计系统中,系统通过相应的多分辨率造型技术,根据角色r的权限集为其提供如图4(b)所示的简化

© 中国科学院软件研究所 http://www.jos.org.cn

装配体模型.

2.3.4 多层次权限模型的性质

从多层次权限模型的定义和计算过程可以看出,该模型具有如下 4 个性质:

(1) 默认情况下,权限集PS不具有对任何特征的操作权限,这可以从权限计算中初始化语句set $PS_0(feature_0,m_0)=(feature_0,m_0,0\%)$看出.

(2) 特征层权限对于零件层权限是一种替换关系,而不是累加关系.即对于权限集$PS_m=\{(part_i,mode,v_1),(feature_{ij},mode,v_2)\}$,则$PS_m(feature_{ij},mode)=(feature_{ij},mode,v_2)$.

(3) 权限的表示简洁,表现能力强.CAD 零件一般含有众多的特征,通过权限分层可以极大地简化权限的定义过程并增强权限的表达能力.

若有一个零件$part_1$含有 100 个特征$\{feature_1,feature_2,\ldots,feature_{100}\}$,定义一个只能读取所有特征 10% LOD 的权限,传统权限模型需要针对每个特征$feature_i$分别定义 100 个特征层权限$FP_i(feature_i,\text{READ},10\%)$.通过权限分层则只需要一个零件层权限$PP_1(part_1,mode,10\%)$即可.如欲在此基础上特别限制权限则不能读取特征$feature_5$的任何细节,只需在此基础上再定义一个特征层权限$FP_5(feature_5,\text{READ},0\%)$即可,即

$$PS=\{(part_1,mode,10\%),(feature_5,\text{READ},0\%)\}.$$

(4) 可以表达对新增特征的访问权限,以满足产品设计的动态变化特性.零件层权限的作用范围是零件中的所有特征,因此也适用于后续操作中新增加的特征,默认地表达了对新增特征的权限.

如有权限集为$PS=\{(part,mode,10\%)\}$,则表示PS在$mode$模式下对$part$零件中所有特征有 10% LOD的访问权限.假设t_1时刻零件$part$有 100 个特征$\{F_i\}(i=0\sim99)$,t_2时刻增加了 50 个新特征$\{F_i\}(i=100\sim149)$,则PS权限不但适用于原有 100 个特征,而且也适用于新增加的 50 个特征.

3 访问控制策略和机理

MLDAC 模型是一个随时间、环境变化的动态访问控制模型,通过多层次的权限模型提供多粒度的模型保护机制.在该模型中,访问控制的实施过程分为 3 个阶段:权限分配、角色权限激活和动态权限调整.

3.1 权限分配

权限分配过程包括权限分层定义、角色权限分配、用户角色分配.

(1) 权限分层定义(permission hierarchical definition),主要是根据产品模型的功能特点及设计者的协同目的对产品模型进行划分,分别定义所需要的零件层权限和特征层权限.此过程可以映射为数据集D与操作集OP的关联,可用二元组$\langle ob,op\rangle$表示.在具体定义时,可以先定义大粒度零件层权限,然后再根据用户具体需要定义细粒度的特征层权限.在权限定义时,还需根据特征操作的同步性、顺序性、互斥性关系来定义特征权限间的依赖关系.一般只有少数特征权限间具有依赖关系,因此可以用一个依赖关系对照表来记录权限间的依赖关系.

(2) 角色权限分配(permissions assignment,简称 PA),主要是依据不同用户的设计任务及权限间的相互关系来定义角色关联的权限集.PA 可以映射为角色集R与访问权限集P的关系,可用二元组$\langle r,p\rangle$表示.但在权限分配过程中还必须考虑特征层权限的互斥关系,以防止角色中的权限相互冲突,即$\exists p\in r$ and $p\leftrightarrow\neg p'$,则不能将p'分配给角色r.

(3) 用户角色分配,可以表示成用户集U与角色集R的二元关系,用二元组$\langle u,r\rangle$表示.用户角色分配过程中必须考虑角色的冲突关系,即如果$\exists r\in u.roles$ and $r\leftrightarrow\neg r'$,则不能将r'分配给用户u.

3.2 角色权限激活

用户在执行具体的角色权限时,需要在上述静态的权限分配基础上动态激活相应的权限.这在 MLDAC 模型中表现为系统激活某个权限,并改变权限的相应状态为“运行状态”.系统通过设计 3 个全局权限链表“运行权限链表(running list)”、“挂起权限链表(waiting list)”和“完成权限链表(finished list)”分别记录系统当前处于运行状态、挂起状态和完成状态的权限,以此辅助实现权限的动态激活.

© 中国科学院软件研究所 http://www.jos.org.cn

如图 6 所示,当用户请求执行某个权限 p 时,首先对用户身份进行验证(authentication),判断其是否具有该权限的某个角色,如果条件满足,则更改权限状态为“就绪状态(ready)”,否则,验证失败直接返回;然后计算权限 p 的激活条件 *activate*(p),通过比较激活条件 *activate*(p)与系统全局权限链表,判断权限 p 的激活条件是否满足,如果是,则用户开始执行此权限 p,并更改其状态为“运行状态(running)”,添加至全局 Running List,否则拒绝用户请求,并更改权限为“挂起状态(hold)”,并将它加入至全局 Waiting List 中;当权限 p 执行完毕后,改变其状态为“完成状态(accomplished)”,并将它从全局 Running List 中转移至 Finished List;当权限激活或权限运行时,如果出现异常则终止执行过程,并将权限 p 置为“睡眠状态(dormant)”和移出全局 Running List;如果等待任务列表中的挂起权限的激活条件在某个时刻得到满足,则恢复(restore)执行该权限,并更改其状态为“运行状态”,从 Waiting List 移至 Running List.

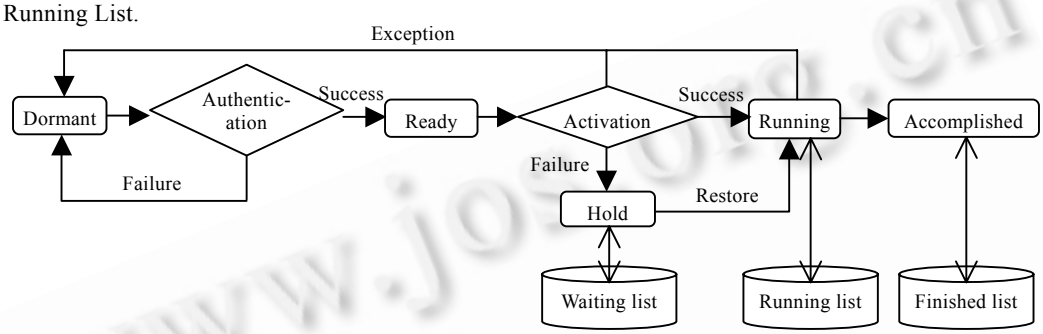


Fig.6 Detailed permission states of an activation step of role privileges

图 6 角色权限动态激活与权限状态迁移

3.3 权限动态调整

在协同设计过程中,当产品模型发生变化,或任务工作安排发生变化时,可能需要改变用户的角色或角色的权限,则会涉及到权限动态调整过程.权限动态调整过程与权限分配过程一样,都需要满足角色中权限不互斥和用户中角色不冲突两个原则,这个过程一般由系统管理员实现.

4 实例系统

为了验证 MLDAC 模型的可行性及其性能,我们在自主开发的协同特征造型原型系统(collaborative feature modeling prototype system,简称 CFMPS)中应用 MLDAC 模型.CFMPS 采用服务器/客户端结构,所有的造型及计算操作由服务器实现.服务器维护全局统一的产品模型,客户端从服务器获取产品模型的离散化数据,用于模型显示.服务器主要由 MLDAC 模型、特征造型器和产品模型组成,其系统结构简图如图 7 所示.

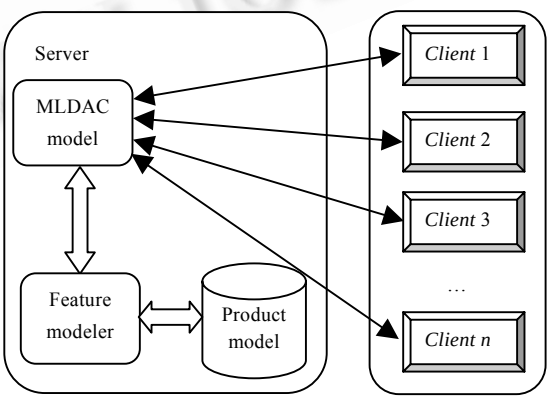


Fig.7 The system structure of CMFPS

图 7 CMFPS 系统结构图

© 中国科学院软件研究所 http://www.jos.org.cn

在某个协同设计会话中,服务器根据设计产品模型的结构特点和参与人员的职责进行任务分工,从而进行权限定义、角色定义与授权、用户定义与授权.当用户在协同设计过程中需要执行某个权限操作时,首先将此需求以消息的方式发送至服务器;服务器对操作用户进行身份验证,判断用户是否属于相应操作权限的某个角色;然后还需要根据当前运行状态对操作权限进行动态激活,如果满足条件则执行相应操作,并将结果数据反馈至各个客户端.当设计人员或其身份发生变化时,需要进行权限的动态调整.

经过验证,MLDAC 模型可以很好地满足协同 CAD 设计的需求,可以更好地辅助系统实现模型的一致性与并发性控制,以保证设计任务有序、正确、安全地执行.

5 结 论

安全问题是协同设计过程的关键问题,现有的访问控制方法由于受到种种限制,都不能完全适用于 CAD 协同环境的特殊需求,为此,我们提出了一个专门针对 CAD 协同设计环境的多层次动态安全访问控制(MLDAC)模型.该模型根据 CAD 产品模型的多层次结构特点,提出了一种多层次的权限模型,以简化权限定义及其分配过程,同时丰富了权限表达能力,实现了产品模型的多粒度访问控制.通过参照工作流的基本理念,引入权限的依赖关系及权限状态迁移概念,实现了权限的动态激活及动态调整.这符合设计任务间的分工性、依赖性和交互性特点,顺应了访问控制模型的发展趋势.由于 MLDAC 模型以层次 RBAC 模型为基础,与传统的访问控制模型相比,它除了具有层次 RBAC 模型的一切优点(如支持职责分离原则和最小特权原则)以外,还有如下特点:

(1) 支持多粒度的权限访问控制机制.传统的访问控制模型没有充分考虑 CAD 协同设计的任务分工、产品层次结构的特殊性,而 MLDAC 模型针对多层次 CAD 产品建模特点,提出了多层次权限模型与之相适应,为协同设计提供更加符合设计过程和设计意图的权限控制.

(2) 支持权限动态更新.在协同设计过程中,用户的访问权限和产品模型是时刻变化的,MLDAC 模型通过分层的权限模型、权限的状态迁移,实现了用户权限可随设计环境和产品模型的变化而变化,以满足协同设计过程的实时性、交互性和分布性特点.同时结合权限间的依赖关系约束,可实现对具有时序性、相关性设计操作进行有序的控制,以保证任务正确有序的执行和防止权力滥用.

(3) 支持群组权限管理.MLDAC 模型通过角色继承和多重性关系实现了基于角色的用户群组管理,通过引入工作组(team)对象加大对用户的管理力度,避免了直接在用户和权限之间进行授权和取消.

(4) 具有更好的安全性.通过权限的依赖关系管理、角色权限动态激活及调整等,使 MLDAC 具有更好的安全性,同时,其安全管理更加灵活、便捷.

然而,由于 MLDAC 模型通过权限层次化、权限依赖关系约束和权限状态迁移等实现权限的灵活控制,其中需要进行权限的层次划分、权限依赖关系的定义及权限状态的定义等预处理过程.对于复杂产品设计来说,这些过程会显得非常繁琐,而且容易出现错误,如出现无效的权限定义、权限依赖关系冲突、权限状态迁移紊乱等.因此,需要设计一个良好的规范或流程,以指导这些预处理过程正确、快速地完成.此外,本文所提出的多层次权限模型主要针对 CAD 领域的特征模型,并不能直接应用于其他类型的产品模型,同时也没有针对大规模装配体模型进行优化.在以后的研究中,我们将针对这些问题展开更加深入的研究.

致谢 感谢审稿专家为本文提出了许多宝贵意见.

References:

[1] Gladney HM, Worley EL, Meyers JJ. An access control mechanism for computing resources. IBM Systems Journal, 1975,14(3): 212−228.

[2] Ohbuchi R, Masuda H, Aono M. A shape-preserving data embedding algorithm for NURBS curves and surfaces. In: Proc. of the Computer Graph Interface 1999 (CGI '99). Washington: IEEE Computer Society, 1999. 180−187. http://citeseer.ist.psu.edu/273603.html

[3] Praun E, Hoppe H, Finkelstein A. Robust mesh watermarking. In: Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'99). New York: ACM Press/Addison-Wesley Publishers, 1999. 49−56.

[4] Ohbuchi R, Takahashi S, Miyazawa T, Mukaiyama A. Watermarking 3D polygonal meshes in the mesh spectral domain. In: Watson B, Buchannan JW, eds. Proc. of the Graphics Interface. Ontario: Canadian Information Processing Society, 2001. 9−17.

© 中国科学院软件研究所 http://www.jos.org.cn

[5] Zhang XY, Peng W, Zhang SY, Ye XZ. Review of watermarking techniques for 3D polygonal models. Journal of Computer-Aided Design & Computer Graphics, 2003,15(8):913−920 (in Chinese with English Abstract).
[6] Shen HH, Dewan P. Access control for collaborative environments. In: Proc. of the 1992 ACM Conf. on Computer-Supported Cooperative Work. New York: ACM Press, 1992. 51−58. http://citeseer.ist.psu.edu/59575.html
[7] Bullock A, Benford S. An access control framework for multi-user collaborative environments. In: Proc. of the Int'l ACM SIGGROUP Conf. on Supporting Group Work (GROUP '99). New York: ACM Press, 1999. 140−149. http://portal.acm.org/citation. cfm?doid=320297.320313
[8] Shyamsundar N, Gadh R. Internet-Based collaborative product design with assembly features and virtual design spaces. Computer-Aided Design, 2001,33(9):637−651.
[9] van der Hoeven A, ten Bosch O, van Leuken R, van der Wolf P. A flexible access control mechanism for CAD frameworks. In: Proc. of the Conf. on European Design Automation. Los Alamitos: IEEE Computer Society Press, 1994. 188−193. http://citeseer.ist.psu.edu/186662.html
[10] Cera CD, Kim T, Han JH, Regli WC. Role-Based viewing envelopes for information protection in collaborative modeling. Computer-Aided Design, 2004,36(9):873−886.
[11] Shi ML, Yang GX, Xiang Y, Wu SG. WfMS: The manage system of workflow. Chinese Journal of Computers, 1999,22(3):325−334 (in Chinese with English Abstract).
[12] Crampton J. A reference monitor for workflow systems with constrained task execution. In: Ferrari E, Ahn GJ, eds. Proc. of the 10th ACM Symp. on Access Control Models and Technologies (SACMAT 2005). New York: ACM Press, 2005. 38−47.
[13] Thomas RK, Sandhu RS. Towards a task-based paradigm for flexible and adaptable access control in distributed applications. In: Proc. of the 1992-1993 ACM SIGSAC New Security Paradigms Workshops. New York: ACM Press,1993. 138−142. http://citeseer.ist.psu.edu/184664.html
[14] Thomas RK, Sandhu RS. Task-Based authentication controls (TABC): A family of models for active and enterprise-oriented authentication management. In: Proc. of the IFIP WG11.3 Workshop on Database Security. 1997. 11−13.
[15] Deng JB, Hong F. Task-Based access control model. Journal of Software, 2003,14(1):76−82 (in Chinese with English Abstract). http://www.jos.org.cn/1000-9825/14/76.htm
[16] Wang XM, Zhao ZT, Hao KG. A weighted role and periodic time access control model of workflow system. Journal of Software, 2003,14(11):1841−1848 (in Chinese with English Abstract). http://www.jos.org.cn/1000-9825/14/1841.pdf
[17] Xu F, Lai HG, Huang H, Xie L. Service-Oriented role-based access control. Chinese Journal of Computers, 2005,28(4):686−693 (in Chinese with English Abstract).
[18] Fang CH, Peng W, Ye XZ. Network-Centric geometry modeling technology. Journal of Computer-Aided Design & Computer Graphics, 2005,17(5):879−888 (in Chinese with English Abstract).

附中文参考文献:

[5] 张新宇,张三元,叶修梓.3D网格数字水印研究进展.计算机辅助设计与图形学学报,2003,15(8):913−920.
[11] 史美林,杨光信,向勇,伍尚广.WfMS:工作流管理系统.计算机学报,1999,22(3):325−334.
[15] 邓集波,洪帆.基于任务的访问控制模型.软件学报,2003,14(1):76−82. http://www.jos.org.cn/1000-9825/14/76.htm
[16] 王小明,赵宗涛,郝克刚.工作流系统带权角色与周期时间访问控制模型.软件学报,2003,14(11):1841−1848. http://www.jos.org.cn/ 1000-9825/14/1841.htm
[17] 许峰,赖海光,黄皓,谢立.面向服务的角色访问控制技术研究.计算机学报,2005,28(4):686−693.
[18] 方萃浩,彭维,叶修梓.以网络为中心的几何造型技术.计算机辅助设计与图形学学报,2005,17(5):879−888.

方萃浩(1977－),男,江西于都人,博士,主要研究领域为几何造型,协同设计,信息安全.

彭维(1973－),男,博士,副研究员,主要研究领域为三维 CAD,网络协同设计,人机交互,3D模型检索.

叶修梓(1966－),男,博士,教授,博士生导师,主要研究领域为 CAD,几何造型,图形图像技术,生物信息,GIS,数据库应用.

张引(1970－),女,博士,副教授,主要研究领域为计算机图形图像处理,CAD.

© 中国科学院软件研究所 http://www.jos.org.cn