ISSN1000-9825 Journal of Software 软件学报 2000,11(7):931~937

基于角色的CSCW系统访问控制模型*

李成锴 詹永照 茅兵 谢立

(南京大学计算机软件新技术国家重点实验室 南京 210093)

(南京大学计算机科学与技术系 南京 210093)

E-mail: lck@dislab.nju.edu.cn

摘要 针对现有的方法不能很好地满足CSCW系统对访问控制提出的新的需求,该文提出了一个基于角色的CSCW系统访问控制模型RBCSAC(role-based collaborative systems access control).该模型形式化地描述了数据、操作、权限、角色和用户等要素及其相互间的关系,提供访问控制信息的记录方法,通过分配和取消角色来完成对用户权限的授予和取消,并且提供了角色分配规则和操作合法性检查规则.该模型针对CSCW系统的多用户、交互、协作、实时、动态等特性,能较好地满足CSCW系统对访问控制的需求.

关键词 CSCW,访问控制,角色.

中图法分类号 TP311

计算机支持协同工作(CSCW)是指一组用户在一个共享的工作环境中协作地完成一个任务[1].协同用户需要对共享环境中的数据进行访问,然而谁能够以何种方式来访问什么数据是需要规定的.首先,各用户具有不同的身份地位、专业特长、任务分工;其次,不同的数据具有不同的共享范围、协同感知需求和安全性要求[2].因此,需要制定访问控制策略.有一些系统给所有用户相同的权限或者依靠社交协议(social protocol)由用户自己协调来进行访问控制,这会导致诸多问题,例如存取错误、冲突和不一致以及非授权访问等[3].

一般的共享系统都要求制定一定的访问控制策略.CSCW系统对访问控制提出了一些有别于其他系统的新的需求,C. A. Ellis[1]以及Shen Hong-hai和Prasun Dewan等人[3]认为这些需求主要有以下几点:

(1) 对用户组进行访问控制:应该提供由用户所在的组来决定其访问权限的机制;

(2) 支持动态改变用户权限:用户权限在不同的协同阶段可以根据需要动态地改变;

(3) 支持协同权限的说明和控制:CSCW系统中除了普通的数据访问操作如读/写外,还有有关用户交互、协作的访问操作,应该提供相应的对协同权限的控制;

(4) 提供方便的授权/取消机制和操作合法性检查机制.

同时,我们认为,根据CSCW系统所具有的特性,访问控制方法中还涉及到如下有别于其他系统的需求:

(1) 用户之间的授权关系:CSCW系统具有多用户交互和协作等特性,各用户可以授予其他用户以某些权限,可对自己和其他用户的视图进行裁剪等.

(2) 支持对操作依赖关系的描述:CSCW系统的实时性以及协同用户的交互性和分布性,使得操作序列具有时间上的依赖关系,虽然这还涉及到并发控制和冲突消解策略,但是,在访问控制方法中提供适当的支持将有助于协同工作的顺利进行.

常用的访问控制方法是由Lampson提出、由Graham和Denning改进的权限矩阵模型[4,5].权限矩阵A的行下标是实体集之中的元素s,列下标为数据集之中的元素o,相应的矩阵元素$A[s,o]$表示实体s所具有的对数

* 本文研究得到国家863高科技项目基金(No.863-306-ZT02-03-01)资助.作者李成锴,1975年生,硕士生,主要研究领域为分布式计算,CSCW.詹永照,1962年生,博士生,主要研究领域为分布式计算,计算机图形学.茅兵,1967年生,博士,副教授,主要研究领域为分布式计算,CSCW.谢立,1942年生,教授,博士生导师,主要研究领域为分布式计算,并行系统.

本文通讯联系人:李成锴,南京 210093,南京大学计算机软件新技术国家重点实验室

本文1999-01-19收到原稿,1999-07-12收到修改稿

据 o 的合法操作集.参照上述几条需求,我们发现:首先,这个模型不能提供对一组用户的访问控制,因为每个矩阵元素都只是对单个实体在单个数据上的访问权限进行了规定;其次,它没有涉及到如何提供协同权限的说明和控制;再次,这种方法不能简明而方便地授权,必须对每一个实体在每一个数据上的权限进行分配,即对每一个矩阵元素进行赋值;最后,没有针对 CSCW 的特点提供对用户之间授权和交互以及实时性的支持.一种与权限矩阵类似的访问控制方法是权能表(capability list).Ellis[5]认为这种方法只适合于非实时的多用户系统,其用户不像在 CSCW 系统中那样是紧耦合的.

基于角色的访问控制在数据库系统以及系统和网络管理领域初露端倪.最新的数据库管理系统标准 SQL3 中已经出现了有关角色的内容,在 Oracle 等大型数据库系统中得到采用[6,2].Novell 的 Netware 和 Microsoft 的 Windows NT 等网络操作系统中也将角色用于系统管理中[6].Zahir Tari 等人提出的一个用于 Intranet 安全管理的基于角色的访问控制模型 I-RBAC[7],通过用户授权机制和用户角色分配机制来保证对 Intranet 资源的安全使用.对比前述几条需求,该模型提供了对用户组(角色)的访问控制,但不是针对协同系统的,未提供对交互、协作、实时性等特性的支持以及对协同权限的控制,对协同会话过程中用户动态、灵活地改变权限的能力也不能较好地支持.

有一部分工作可以作为在 CSCW 系统中将角色用于访问控制的例子.HongHai Shen 等人曾以一个协同编辑框架 Suite 为背景设计了一个访问控制模型[3].该模型从主体、受体、访问操作这 3 个角度去规定访问权限,其中从主体的角度对访问权限按角色进行说明和继承.但该模型还存在一些问题,一个是没有提供形式化的描述方法和应用规则,不利于模型的应用,还有一个是没有针对 CSCW 系统的特性提供对操作依赖关系等的描述.R.B.Smith 等人在文献[8]中介绍了一个共享空间应用系统设计环境 Kansas,认为用户的角色关系到系统的输出和用户的输入.但 Kansas 不支持对协同权限的说明和控制,认为没有对角色的显式表示,角色是通过空间位置和权能系统来体现的,因此只适合于共享空间一类的应用.

综上所述,现有的方法不能很好地满足 CSCW 系统对访问控制的需求,在 CSCW 领域内所作的一些研究还不甚深入和全面.鉴于此,我们提出了一个基于角色的 CSCW 系统访问控制模型 RBCSAC(role-based collaborative systems access control).本文对该模型进行了非形式化和形式化的描述,并给出了该模型的访问控制信息记录方法以及如何应用该模型,还介绍了一个基于该模型的实例系统.RBCSAC 提供显式的、较为完善的角色机制,有形式化的描述和可行的应用规则,能较好地满足 CSCW 系统对访问控制的需求.

1　基于角色的 CSCW 系统访问控制模型 RBCSAC

1.1　模型概述

如图 1 所示,一般的访问控制方法,例如权限矩阵模型,直接在用户与数据之间进行访问控制.这种方法的弊端前已述及,此处不再赘述.在我们的模型 RBCSAC 中,涉及访问控制的要素包括用户、角色、权限、操作和数据.这些要素之间的关系如图 2 所示.

Fig. 1　Direct access control between user and data

图1　直接在用户与数据之间进行访问控制

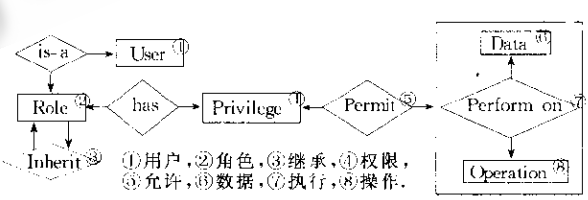

Fig. 2　The relationships between key factors in RBCSAC model

图2　RBCSAC模型各要素之间的关系

数据　这里所指的数据是一个广泛的概念,它可以是文件、数据库、界面元素等.

操作　例如,$OP=\{open, save, close, read, draw, erase, scroll, pageturn, telepoint\}$表示某一白板系统的操作集,其中既有一般的访问操作,如 draw(绘图)、scroll(滚动)等,也有与协同有关的操作,如 telepoint(远程指针).

权限　权限被表示为在某数据上的一组操作的集合.例如:$drawB=$(白板 B,$\{read, draw\}$)定义了一种称

为 drawB 的权限,它允许在白板 B 上进行读和绘图操作.再如 $fileop=$(文件 F,$\{open,close\}$)定义了一种称为 fileop 的权限,它允许对文件 F 进行打开和关闭的操作.

角色 角色是一组访问权限的集合.例如,$drawer=\{drawB,fileop\}$定义了一种称为 drawer 的角色,这种角色具有 drawB 和 fileop 两种权限.模型还支持通过继承机制来定义角色.

用户 参与协作的每个结点上都有一个或多个用户,根据用户之间存在的差异,分别赋予某种或某几种角色,用户就具有该角色所对应的访问权限.

由图 2 可见,借助于角色、权限、操作这些要素,可以在用户与数据之间进行间接的访问控制.

1.2 形式化描述

定义 1(数据集). 数据集是 CSCW 系统中共享数据的有限集合,记为 D.

定义 2(操作集,操作子集). 操作集(记为 OP)是一个有限集,OP 中的每个元素都表示一种可以对数据施行的操作.如果 $o\subseteq OP$,则 o 称为一个操作子集.

定义 3(访问权限集,访问权限). 访问权限集是集合 $D\times 2^{OP}$ 的子集,记为 P,P 中的每一个元素都表示一种访问权限.访问权限的直观含义是,若 $(d,o)\in P$,其中 $o\subseteq OP$,则表示对数据 d 可以有一种访问权限,允许对数据 d 进行操作子集 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 $R\subseteq 2^P$.角色集中每个元素都表示一种角色.

对任何角色 $r\in R$,显然 $r\subseteq P$,r 是一组访问权限的集合.需要说明的是,角色中的权限允许重叠.对于角色 r,权限 $p_1,p_2\in r$,其中 $p_1=(d,o_1)$,$p_2=(d,o_2)$,如果 $o_1\cap o_2\neq\varnothing$,则 p_1 与 p_2 是重叠的.例如,权限 $drawB=$(白板 B,$\{read,draw\}$),$eraseB=$(白板 B,$\{read,erase\}$),如果定义角色 $painter=\{drawB,eraseB\}$,则这个角色中的两个权限是重叠的.

定义 5(结点集). 一个 CSCW 系统的结点集为一有限集(记为 S),S 中的每个元素都表示一个结点.

定义 6(用户). 一个用户定义为一个二元组 $(s\cdot name,UR)$,其中 $s\in S$,$UR\subseteq R$,$name$ 是一个字符串.

在用户的定义中,$s\cdot name$ 在整个系统中唯一地标识该用户,s 是用户所在的结点,$name$ 是用户的名称,同一结点上不同的用户具有不同的名称.UR 是用户的当前角色集,用户可以同时具有多种角色.

定义 7(角色继承). $\forall r_1,r_2\in R$,如果 $r_1\subset r_2$,则称 r_2 继承 r_1,r_1 与 r_2 之间满足继承关系,记为 $r_1\rightarrow r_2$.r_1 称为父角色,r_2 称为子角色.显然,角色继承关系具有传递性.

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角色的基础上进行.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,多继承体现了对角色的综合能力.

设角色 r 继承了角色 r_1,则 r 的权限包括 r_1 的权限和扩充的权限,即 $r=r_1\cup\Delta r$;如果 r 继承了多个角色 $r_1,r_2,\ldots,r_n$,则 $r=r_1\cup r_2\cup\ldots\cup r_n\cup\Delta r$.如图 3 所示的继承关系,则 $painter=drawer\cup eraser$,$telepointer=painter\cup\{telepointB\}$.telepointer 继承了 painter 具有的权限,同时扩充了新的权限 telepointB,其中 $telepointB=$(白板 B,$\{telepoint\}$).一个角色 r,如果它是初始角色,则记录下其全部权限;如果 r 继承了其他角色,则记录扩充的权限 Δr 和继承关系.

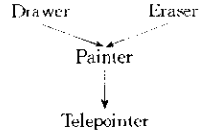

Fig. 3 Inheritance of roles
图3 角色的继承

定义 8(角色指派). $\forall r_1,r_2,r_3\in R$,如果具有角色 r_1 的用户可以把角色 r_3 赋予具有角色 r_2 的用户,则称 r_1 对 r_2 具有在 r_3 上的指派关系,记为 $r_1\xrightarrow{r_3}r_2$.

规则 1. 对于 $r_1,r_2,r_3,r_4\in R$,$r_1\xrightarrow{r_3}r_2$,如果 $r_4\rightarrow r_3$,则 $r_1\xrightarrow{r_4}r_2$.

CSCW 系统中多个协同用户具有不同的权力和责任,指派关系提供用户自己进行权限分配的手段.这样,不仅程序员在系统编写时可以管理用户的角色,用户自身在运行时刻也可以对角色进行管理.角色指派支持用户在协同会话过程中动态地裁剪自身和其他用户的权限.例如,$administrator\xrightarrow{administrator}author$,说明具有 administrator 角色的用户可以把这一角色赋予具有 author 角色的用户.这可以用于管理员中途退出协同会话的情况,

此时管理员可以指派某位具备 author 角色的用户来接任 administrator 这一角色,这对于维护协同会话的连续性很重要,而普通用户是不具有这一指派能力的.又比如,当一个新用户刚加入协同会话时,可以只赋予其旁观的权限,当其对参与协同会话的其他用户和正在进行的工作逐渐熟悉了以后,可以由管理员将权限较高的角色分配给该用户,使其可以参与到工作中来.

定义 9(操作冲突). $\forall d_1,d_2\in D,op_1,op_2\in OP$,如果任何用户都不能同时具有在 d_1 上的操作 op_1 与在 d_2 上的操作 op_2 的权限,则称 d_1 上的 op_1 与 d_2 上的 op_2 是冲突操作,记为 $op_{1(d_1)}|op_{2(d_2)}$.

规则 2. 对于权限 $p=(d,o)$,其中 $o\subseteq OP$,如果$\exists$ $op_1\in o,op_2\in o$,则不允许 op_1 与 op_2 是 d 上的冲突操作.

定义 10(权限冲突). $\forall p_1=(d_1,o_1),p_2=(d_2,o_2)\in P$,其中 $d_1,d_2\in D,o_1,o_2\subseteq OP$,如果$\exists$ $op_1\in o_1,op_2\in o_2$,且 $op_{1(d_1)}|op_{2(d_2)}$,则称 p_1 与 p_2 是冲突权限,记为 $p_1|p_2$.

规则 3. 对于角色 $r\in R$,如果$\exists$ $p_1\in r,p_2\in r$,则不允许 p_1 与 p_2 是冲突权限.

定义 11(角色冲突). $\forall r_1,r_2\in R$,如果$\exists$ $p_1\in r_1,p_2\in r_2$,且 $p_1|p_2$,则称 r_1 与 r_2 是冲突角色,记为 $r_1|r_2$.

性质 1. 对于 $r_1,r_2,r_3\in R$,如果 $r_1|r_2$,且 $r_2\rightarrow r_3$,则 $r_1|r_3$.

规则 4. 对于用户 $u=(s\cdot name,UR)$,如果$\exists$ $r_1,r_2\in UR$,则不允许 r_1 与 r_2 是冲突角色.

角色冲突的概念提供了一种对角色间关系的描述,规定了用户不能同时具有的角色,这样,在角色分配时就可以进行控制.描述角色冲突关系有利于合理划分用户职责:如一个协作装配任务,进行一台机器 m 的装配,希望划分为 n 个子部分,分别交付给 n 个不同的人去完成.可以把不同的子部分上的装配操作 $a_i(i=1..n)$定义为相互冲突的操作,进而定义冲突权限,即 $p_i=(m_i,\{a_i\})(i=1..n),p_i|p_j(i\neq j)$,进一步可以定义冲突角色 $r_i=\{p_i\}(i=1..n),r_i|r_j(i\neq j)$,这样,我们就可以保证各个子部分的装配是分配给各自不同的人完成的.对角色冲突关系的描述同时也是一种基本的安全措施,可以防止权力的滥用.例如,在现实生活中,会计与出纳应该是不能由同一个人同时兼任的,我们可以定义 $cashier|accountant$.

定义 12(操作依赖). $\forall op_1,op_2\in OP,d\in D$,如果对 d 的操作 op_2 必须在对 d 的操作 op_1 执行完成之后才可以执行,则称 op_2 在 d 上依赖于 op_1,记为 $op_2>_t(d)op_1$.

操作依赖的直观含义是:定义操作的时间顺序关系,某个用户在执行对 d 的某种操作 op_2 时,必须已经有某个用户执行了操作 op_1,如果我们同时规定 $op_{1(d)}|op_{2(d)}$,这样就使必须合作的操作序列不能由单个用户完成,防止权力滥用.另外,操作依赖关系也可以对 CSCW 系统的实时性和交互性给予支持.例如,协同用户在进行表决时,需要已经有某个用户将某项提议交付表决,即表决$>_t$(提案)提交.

1.3 访问控制信息的记录

一个角色是一个包含一组访问权限的集合,即 $r=\{p_1,p_2,\ldots,p_n\}$,其中 $p_i=(d_i,o_i)$.实际的表示中,可以采用两种方法.一种方法是字符串标识法,采用字符串来表示数据名、操作名和权限名.例如,$drawB$=("白板 B",{"$read$","$draw$"}),$eraseB$=("白板 B",{"$read$","$erase$"}),角色 $painter$={"$drawB$","$eraseB$"}.另一种是指针标识法,用指针指向数据和权限,操作名仍然用字符串来表示.例如,$drawB$=(↑白板 B,{"$read$","$draw$"}),$eraseB$=(↑白板 B,{"$read$","$erase$"}),角色 $painter$={↑$drawB$,↑$eraseB$},其中↑白板 B 是指向数据对象的指针,↑drawB 和↑eraseB 是指向权限对象的指针.可见,记录角色 r 只要记录其各个权限 $p_i(1\leqslant i\leqslant n)$的标识(字符串或指针)即可,而 p_i 的具体内容包括数据 d_i 的标识和操作子集 o_i 的标识.

1.4 模型的应用

在应用一种访问控制的方法时,以下两个方面的问题是要解决的:(1) 授权和取消,(2) 操作合法性检查.

· 授权和取消:如何授予用户相应的访问权限以及相应地,如何取消用户的某种访问权限.在 RBCSAC 中,通过分配角色给用户,只要进行几次赋值操作即可完成一次授权.取消用户的权限通过取消用户相应的某种角色来进行.具体的角色分配规则如下:

系统级角色分配规则. 根据第 1.2 节中的规则 4,如果$\exists$ $r'\in u.UR$,使得 $r|r'$,则不能将角色 r 分配给用户 u.

用户级角色分配规则. 对于用户 u_1 和 u_2 以及角色 r,假设 u_2 与 r 是符合系统级规则的.如果$\exists$ $r_1\in u_1.UR$,

$r_2 \in u_2.UR$,且 $r_1 \rightarrow r_2$,则 u_1 可以把角色 r 赋予 u_2.

系统级规则是任何时刻分配角色给用户时都应遵循的规则,而用户级规则是运行时刻用户自己进行角色管理时应遵循的附加规则.

·操作合法性检查:在用户欲进行某种访问操作时,如何检查该用户是否具有相应的权限.我们可以给出两条检查规则如下:

静态操作合法性检查规则. 对于角色 $r=r_1 \cup r_2 \cup \ldots \cup r_n \cup \Delta r$ 和数据 d 上的操作 op,如果以下条件之一成立,则具有该角色的用户 $u(r \in u.UR)$在数据 d 上的操作 op 是合法的.

(1) $\exists (d,o) \in \Delta r$,满足 $op \in o$;

(2) $\exists (d,o) \in r_i (1 \leqslant i \leqslant n)$,满足 $op \in o$.

动态操作合法性检查规则. 假设用户 u 和数据 d 上的操作 op_2 满足静态合法性.如果 $\exists op_1 \in OP$,使 $op_2 > _{(d)} op_1$,则必须在某个用户(可以为 u)对数据 d 执行了操作 op_1 之后,u 在数据 d 上的操作 op_2 才是合法的.

可见,要在满足静态合法性的基础上,同时满足动态合法性,用户的操作才是合法的.

1.5 对CSCW系统需求的满足

RBCSAC模型能较好地满足前面所述的CSCW系统对访问控制的需求,分析如下:

(1) 基于角色对用户组进行访问控制:对一组用户比对单个用户进行访问控制合理,用户的角色即代表了用户所在的组,同时支持角色的继承和多重继承.通过改变用户的当前角色集就可以改变用户的权限,而改变某种角色所包含的权限时又可以改变一组用户的权限.基于角色的访问控制有3个方面的作用:(a) 简化了权限管理,避免直接在用户和数据之间进行授权和取消.NIST(National Institute of Standards and Technology)的研究[9]指出,用户所具有的权限易于发生改变,而某种角色所对应的权限更加稳定;(b) 有利于合理划分职责,用户只有其所应具有的权限,这样可以避免越权行为,有关角色冲突关系的描述即是对此的支持;(c) 防止权力滥用,敏感的工作分配给若干个不同的用户完成,需要合作的操作序列不能由单个用户完成.

(2) 支持动态地改变用户权限:访问权限不是静态的,而应是动态的,例如,新加入会话的用户往往对系统不熟悉,不应给予过多的发言权和过高的权限,以免打乱协同会话过程.RBCSAC模型支持在协同会话过程中根据需要动态地改变用户的角色,通过角色的分配和取消来完成对一组权限进行授权和取消.同时,对一组权限进行操纵是合乎逻辑的,各种权限并不是互相独立而是相互关联的,例如在取消某用户在白板上的绘图权的同时,可能也应该取消其擦除白板的权力.

(3) 支持协同权限的说明和控制:RBCSAC模型对权限的说明是细粒度的,可以描述有关协同权限.例如,我们前面定义过权限telepointB.又如,我们可以定义一种称为awareB的权限,$awareB=$(白板 B,$\{aware\}$),该权限允许感知其他用户的操作,诸如某用户改变了字体的颜色或者某用户退出了等等.

(4) 提供方便的授权/取消机制和检查机制:只要进行简单的赋值操作即可完成授权.同时,由角色分配规则和操作合法性检查规则指导模型的应用.

(5) 用户之间的授权关系:依据角色指派关系,运行系统中的用户自身可以对角色进行管理,这提供了又一种动态改变用户权限的手段.通常,角色指派的权力都在系统中具有管理责任的用户手中.

(6) 支持对操作依赖关系的描述:操作依赖关系的定义及其在操作的动态合法性检查中的应用,符合CSCW系统的实时、交互和分布的特性,与角色冲突等概念的结合也提供了防止权力滥用的手段.

2 实例系统

采用RBCSAC作为访问控制的方法,我们实现了一个实例系统——共享白板系统NUWBoard.系统界面如图4所示,包括多页式共享白板、聊天室、图片浏览窗口和消息感知窗口4个部分.

图5是该系统的部分角色继承关系图.initRole是一个初始角色,它没有父角色,其他角色都直接或间接地继承该角色.具有角色reader的用户可以观看白板,但不能在白板上进行操作;具有角色painter的用户可以读、绘制、擦除白板的内容;具有角色telepointer的用户可以操纵远程指针;具有角色awareRole的用户可以在消息

感知窗口看到其他用户的状态以及执行了哪些操作.具有角色 viewchanger 的用户可以在白板上滚动和翻页.系统中总共定义了 28 种角色,图中没有全部列出,在此不一一赘述.需要注意的是,有些角色并没有实际的用户与之对应,例如,initRole,recorder,operator 等,实际的用户是与其直接或间接子角色相对应的.

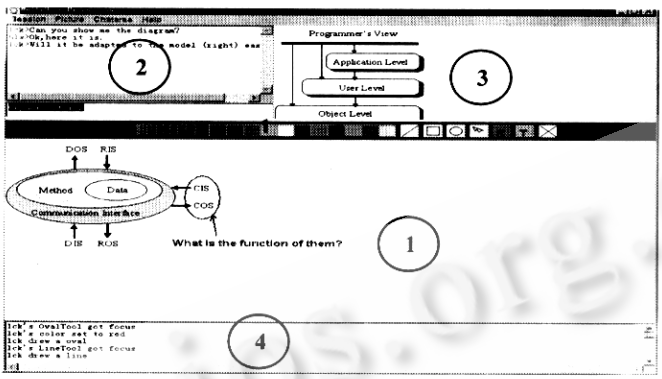

Fig. 4 The user interface of NUWBoard, a sample system

图 4 实例系统 NUWBoard 的用户界面

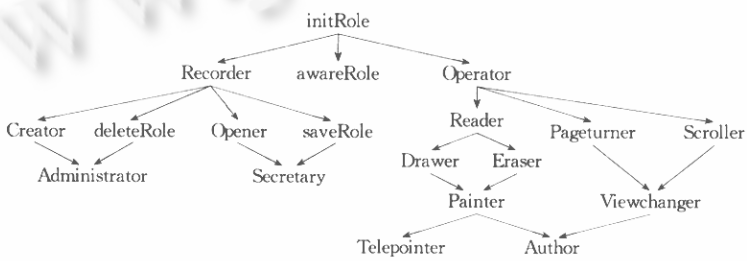

Fig. 5 Role inheritance relationships in NUWBoard, a sample system

图5 实例系统NUWBoard的角色继承关系

3 结束语

本文提出了一个基于角色的 CSCW 系统访问控制模型 RBCSAC.文中对该模型进行了非形式化和形式化的描述,对数据、权限、角色等概念以及角色继承、角色指派、角色冲突、操作依赖等关系给出了形式化的定义、规则和性质等,对于如何记录访问控制权限作了初步的探讨,并且给出了模型的应用规则.RBCSAC 模型能较好地满足 CSCW 系统对访问控制提出的有别于其他系统的需求,本文的工作对如何在 CSCW 系统中提供显式的基于角色的访问控制机制进行了有益的尝试和探讨.我们进一步的工作是开发几个较大规模的实验系统,以进一步验证模型的正确性和有效性,并且研制出一个基于该模型的开发工具,以此作为我们的 CSCW 系统开发工具包中的一个组成部分.

致谢 感谢评审专家对本文初稿提出的正确而中肯的意见,使我们受到很大的启发,并对原文作出了重要的修改和补充.感谢何丹博士,与他的讨论使我们受益匪浅.

参考文献

1 Ellis C A, Gibbs S J, Rein G L. Groupware: some issues and experiences. Communications of the ACM, 1991,34(1):39~58

2 Mao Bing, Xie Li. An object-based computing model for CSCW systems. China Science (Series E), 1997,27(6):542~547 (茅兵,谢立.基于对象的协同计算模型.中国科学(E 辑),1997,27(6):542~547)

3 Shen Hong-hai, Dewan P. Access control for collaborative environments. In: Turner J, Kraut R eds. Proceedings of the ACM CSCW'92 Conference on Computer Supported Cooperative Work. New York: ACM Press, 1994. 51~58
4 Lampson B W. Protection. ACM Operating System Review, 1974,8(1):18~24
5 Graham G S, Denning P J. Protection-Principles and practice. In: Denning P J ed. Proceedings of the Spring Joint Computer Conference. Montvale, NJ: American Federation of Information Processing Societies Press, 1972. 417~429
6 Sandhu R S, Coyne E J, Feinstein H L *et al*. Role-Based access control models. IEEE Computer, 1996,29(2):38~47
7 Zahir T, Shun W C. A role-based access control for Intranet security. IEEE Internet Computing, 1997,1(5):24~34
8 Smith R B, Hixon R, Horan B. Supporting flexible roles in a shared space. In: Poltrock S, Grudin J eds. Proceedings of the ACM CSCW'98 Conference on Computer Supported Cooperative Work. New York: ACM Press, 1998. 197~206
9 Ferraiolo D F, Gilbert D M, Lynch N. An examination of federal and commercial access control policy needs. In: Proceedings of the 16th NIST-NCSC National Computer Security Conference. Gaithersburg, Maryland: National Institute of Standards and Technology, 1993. 107~116

A Role-Based Access Control Model for CSCW Systems

LI Cheng-kai ZHAN Yong-zhao MAO Bing XIE Li

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)
(Department of Computer Science and Technology Nanjing University Nanjing 210093)

Abstract CSCW systems introduce new requirements for access control, which cannot be met by using existing models. In this paper, a new role-based access control model, RBCSAC (role-based collaborative systems access control), is introduced to meet these requirements. This model formally describes the relationship between the key elements of access control such as data, operation, privilege, role and user. It provides the method for recording access control information. The model grants and revokes access privileges of cooperative users by assigning them some roles and canceling their roles. Two role-assignment rules are also provided and two operation legality checking rules. RBCSAC model is brought forward aiming at accommodating with the characteristics of collaborative systems such as multi-user, interaction, collaboration, real-time, dynamic. This model can meet the requirements for access control in CSCW systems adequately.

Key words CSCW, access control, role.