基于情景演算的动态访问控制模型

翟浩良[1] 韩道军[2] 李 磊[1]
(中山大学软件研究所 广州 510275)[1] (河南大学数据与知识工程研究所 开封 475004)[2]

摘 要 访问控制模型定义了安全系统访问控制的整体框架。现有的访问控制模型大多是静态授权模型,尽管可以通过扩展来实现局部动态性(比如可以通过定义条件来实现角色的临时激活等),但在应用时受到了扩展元素的限制,并且已有的大部分模型无法描述授权的动态变化过程。针对以上问题,提出了一种基于情景演算的动态访问控制模型(SCDAC)。SCDAC用逻辑事实和规则来描述访问控制属性和策略,把授权在某一时刻的状态(逻辑事实和规则集合)看作一个情景,通过动作来实现情景的变化,同时刻画了动作执行的前提条件和后续状态的变化情况。最后通过一个实例说明了用SCDAC来描述授权状态的动态变化是可行的。

关键词 访问控制模型,动态,情景演算
中图法分类号 TP301 **文献标识码** A

Dynamic Access Control Model Based on Situation Calculus

ZHAI Hao-liang[1] HAN Dao-jun[2] LI Lei[1]
(Software Research Institute, Sun Yat-Sen University, Guangzhou 510275, China)[1]
(Institute of Data and Knowledge Engineering, Henan University, Kaifeng 475004, China)[2]

Abstract Access control model defines the whole framework of security system access control. The existing access control models are mostly static authorization models. Although the models can be extended to realize local dynamic nature (such as activate roles temporary by defining conditions of them), but they are restricted by the extensional elements in real applications, and most of them can not describe the dynamic changes of the authorized process. To solve the problem, this paper proposed a dynamic access control model based on situation calculus(SCDAC). SCDAC describes the attributes and strategy of access control with logic facts and rules and treats the authorization state at an instance of time as a situation. It can achieve the changes of situation through actions, and portrays the preconditions of the action and successor state axioms. Finally an example was adopted to illustrate describing the dynamic changes of authorization states by SCDAC is feasible.

Keywords Access control model, Dynamic, Situation calculus

1 引言

访问控制是一种重要的保护计算机系统资源的信息安全技术,它通过某种途径和规则显式地准许和限制主体对客体的访问来保证系统资源安全、合法地使用。安全策略是实现访问控制技术的一套特定规则的集合[1]。访问控制模型是从抽象的层次来定义访问控制系统安全属性和表达安全策略的一种概念性框架,它决定了安全策略的表达能力和灵活性,应该具有更新和扩展能力[2]。

传统的访问控制模型,如DAC、MAC和RBAC都是静态授权模型[3,4],一次权限的授予经常可以无数次使用,而且其大多数都使用三元组(主体,操作,客体)来描述系统的授权,缺乏对主体、操作和客体本身属性和属性变化的描述,表达能力和灵活性不强,而且很难适应环境的动态变化和授权的需求变化。随着实际应用的需求和发展,出现了一些基于用户信任度、工作流状态和环境上下文的动态访问控制模型[5-12]。这些模型大都是在授权过程中通过引入条件来实现一定程度的权限动态性,比如通过用户信任度的判断、工作流的状态和环境上下文来动态激活角色等等。但这些模型本身仍然是静态模型,不能对系统授权状态的变化及其过程进行直观而统一的描述;而且只能实现局部动态性(比如角色),缺乏操作、客体和环境动态变化的描述;此外,这些模型也不能描述操作授权的结果和影响,比如上一个操作授权可能会影响下一个操作的授权。

针对以上问题,本文提出一种基于情景演算的动态访问控制模型(SCDAC)。情景演算[13,14]是一种描述动态系统状态变化和动作推理的逻辑形式化方法,具有强大的动态变化描述能力和稳固的理论基础。SCDAC通过情景来描述授权的状态,通过动作来描述授权状态到状态之间的变化;同时通过属性来描述授权的3大元素:主体、操作和客体,增强了模

到稿日期:2011-07-18 返修日期:2011-11-24 本文受澳门科学技术发展基金(013/2010/A),中国博士后科学基金(20100480806)资助。
翟浩良(1983-),男,博士生,主要研究领域为人工智能、信息安全等,E-mail:zhaihliang@163.com;韩道军(1979-),男,博士,主要研究领域为信息安全;李 磊(1951-),男,教授,博士生导师,主要研究领域为计算机软件、数据库与知识库等。

型的授权表达能力。

本文第 2 节介绍了相关的研究工作；第 3 节是基于情景演算的动态访问控制模型(SCDAC)，给出了具体的模型结构描述；第 4 节是一个关于 SCDAC 具体应用的例子；最后总结全文。

2 相关工作

自从 20 世纪 70 年代访问控制作为一个独立的研究方向以来，出现了 3 个应用和研究最广泛的访问控制模型：自主访问控制(DAC)模型、强制访问控制(MAC)模型和角色访问控制(RBAC)模型。

DAC 是一种灵活且易实现的访问控制模型，它的特点是自主授权，即每个被授权的主体都可以自由地把自己拥有的权限赋予另外的主体。但 DAC 也由于其灵活的授权方式，造成了权限管理困难，系统管理员容易失去对系统授权的整体控制。MAC 通过制定主体和客体的访问级别和严格的访问规则来控制主体对客体的访问。MAC 安全性高、保密性强，但难以实现，且不够灵活，因此多用于保密性要求高、多层次级别的军事项目中。RBAC[3]通过角色实现了用户与权限的逻辑分离，可通过增加、修改和删除角色来实现一定的动态性，而且引入角色简化了权限的管理，因此在研究和实际应用中获得了广泛的关注。但 RBAC 是一个基于封闭环境的静态授权模型[4]，没有考虑客体和环境的动态变化，而且 RBAC 无法进行细粒度的权限控制。

随着实际应用的发展和授权复杂性以及多样性的变化，在近 10 年又出现了一些访问控制模型，这些模型大都是对 RBAC 不同程度的扩展。Joshi 在 2005 年提出了一个通用的基于时态的角色访问控制模型(GTRBAC)[16]。GTRBAC 是在 TRBAC[15]的基础上提出的，在角色激活、用户角色授予和角色权限授予的过程中引入时态约束，可以描述角色授予随着时间动态变化。基于任务的访问控制模型(TBAC)[17]和面向服务的访问控制模型(SOAC)[18,19]在访问控制中分别引入任务和服务来动态控制授权过程，利用任务和服务的时效性来动态控制权限的激活和释放。基于属性的访问控制模型(ABAC)[20]使用属性来描述访问控制中的所有元素，如主体、客体和权限等，并通过属性的关系来进行授权和约束控制。ABAC 具有强大的表达能力，能提供细粒度的访问控制，并且有良好的灵活性和扩展性[23]。但 ABAC 没有描述访问控制元素和策略在流程执行过程中的变化以及这种变化与访问控制授权本身的关系。访问控制(UCON)模型[21,22]在访问控制中引入了两个新元素：条件(conditions)和义务(obligation)。条件用来描述访问控制执行需要满足的约束，义务用来描述访问控制执行以前和以后主体必须履行的行为。UCON 兼容性强，支持约束和附加行为的描述，但实现复杂，不支持权限委托。文献[5-12]通过在访问控制过程中引入状态、信任度和环境上下文等条件来实现动态授权，虽然能实现一定程度的权限动态变化，但适用范围受到引入元素的限制。Daniel 在文献[24]中首次提出使用状态机来描述动态访问控制模型，它把环境看作是一个可变化的逻辑事实的集合，策略看作是访问规则的集合，访问控制模型是一个关于策略和环境的状态机，用谓词来描述状态之间的动作。Daniel 在文中还证明了策略之间的包含和等价关系以及目标在策略中的可满足关系等。状态机模型是一个完全的动态访问控制模型，具有很好的理论意义，但模型中的状态很难预先定义，状态之间的变化过程难以描述，因此造成了应用的复杂性。

综上所述，目前提出的大部分访问控制模型都只能实现局部的动态性，比如引入条件限制角色的授予、权限的授予等；而且缺乏对授权变化过程的描述。本文在以上研究工作的基础上，提出基于情景演算的动态访问控制模型(SCDAC)。情景演算具有很强的动态系统描述能力，并已成为动态世界建模方法的标准[25]。因此 SCDAC 使用情景演算理论对授权变化的动态过程进行描述，把情景(即状态)看作是逻辑事实和规则的集合，把新情境的产生看作是动作作用的结果，把授权的变化看作是一系列情景的变化。在 SCDAC 中，逻辑事实使用基于属性的方法来描述，从而增强了模型的表达能力。

3 基于情景演算的动态访问控制模型

情景演算是一个用来刻画动态系统变化规律的一阶系统(个别地方使用了二阶逻辑)[13]，因此它适合用来描述系统的授权变化以及变化过程。一个基于情景演算的动态访问控制模型(SCDAC)就是一个情景演算系统。

3.1 SCDAC 的基本概念

由文献[13]知，情景演算包含以下 3 个基本概念：动作(action)、情景(situation)和流(fluent)。

动作(action)是引起世界状态变化的行为，动态世界的变化都是动作作用的结果。动作可以用函数来表示。在 SCDAC 中，授权状态的变化可以由以下两个动作引起：

(1)增加($add(sub,obj,des)$)：对对象 obj 进行增加操作，增加的内容为 des，sub 为操作的主体即用户。

(2)删除($del(sub,obj,des)$)：对对象 obj 进行删除操作，删除的内容或标识为 des，sub 为操作的主体即用户。

情景(situation)是动态变化的世界在某一时刻的状态，在文献[14]中，情境被定义为初始情景 S_0 经过一系列动作后的系统状态的反映。情景可以用 $do(a,s)$ 来描述，它表示情景 s 下执行动作 a 得到的新情景。如 $do(add(sub,obj,des),s)$ 表示在情景 s 下执行增加动作后产生的新情景。在 SCDAC 中，情景其实是一个流的集合。

流(fluent)是与情境相关的组成系统状态的原子属性，可以用来描述动作的效果。在文献[13]中，流分两种类型：函数流和关系流。在 SCDAC 中，我们增加规则流。关系流可以用谓词来描述，规则流可以用逻辑规则来描述。关系流具体描述如下：

(1)$sub(X,X_1,X_2,\cdots,X_n)$ 表示主体 subject 及主体属性的关系流，X 表示主体的标识，$X_i(1\leqslant i\leqslant n)$ 表示主体的属性。

(2)$oper(Y,Y_1,Y_2,\cdots,Y_n)$ 表示操作 operation 及操作属性的关系流，Y 表示操作的标识，$Y_i(1\leqslant i\leqslant n)$ 表示操作的属性。

(3)$obj(Z,Z_1,Z_2,\cdots,Z_n)$ 表示客体 object 及客体属性的关系流，Z 表示客体的标识，$Z_i(1\leqslant i\leqslant n)$ 表示客体的属性。

(4)$in(Attr_1,Attr_2)$ 表示主体属性或客体属性之间的从属(包含)关系流，即属性 $Attr_1$ 是属性 $Attr_2$ 的从属属性。

(5)$conflict(Attr_1,Attr_2)$ 表示主体属性或客体属性之间的互斥关系流，即属性 $Attr_1$ 和属性 $Attr_2$ 是互斥的属性。

(6)$delegate(X_1,X_2)$表示主体之间的权限委托关系流，即主体 X_1 把权限委托给主体 X_2。

说明：n 的取值跟具体应用需求有关。例如，在实际应用中，可取 $n=4$ 或 $n=8$ 等。

一个关系流 $pred(X_1,X_2,\cdots,X_n)$，如果在情境 s 的流集合中存在，则关系流在 s 中为真，否则为假。

在 SCDAC 中，具体的规则流描述如下。

(1)主体操作访问规则：主体操作访问规则定义了主体和操作之间的可访问关系，主要有两种类型：主体操作访问许可规则和主体操作访问拒绝规则，也即满足什么条件的主体可操作和不能操作该操作。两种形式的规则如下：

$sub\_oper\_permit(X,Y)$:-$sub(X,X_1,X_2,\cdots,X_n)$,$oper(Y,Y_1,Y_2,\cdots,Y_n)$,$X_{i1}\nabla Y_{j1},\cdots,X_{ik}\nabla Y_{jk}$

$sub\_oper\_deny(X,Y)$:-$sub(X,X_1,X_2,\cdots,X_n)$,$oper(Y,Y_1,Y_2,\cdots,Y_n)$,$X_{i1}\nabla Y_{j1},\cdots,X_{ik}\nabla Y_{jk}$

约定 在主体操作访问规则和下述访问规则中，$\{X_{i1},\cdots,X_{ik}\}\subseteq\{X_1,X_2,\cdots,X_n\}$，$\{Y_{j1},\cdots,Y_{jk}\}\subseteq\{Y_1,Y_2,\cdots,Y_n\}$，$\{Z_{j1},\cdots,Z_{jk}\}\subseteq\{Z_1,Z_2,\cdots,Z_n\}(k\geqslant 1)$，$y_i,z_i(1\leqslant i\leqslant n)$是常量，$\nabla$是比较关系符号，$\nabla\in\{=,!=,>,<,\leqslant,\geqslant\}$。$sub\_oper\_permit$ 表示许可规则，sub_oper_deny 表示拒绝规则，以下类似。

(2)操作客体访问规则：操作客体访问规则定义了客体和操作之间的可访问关系，也即一个客体可以被执行什么操作。与主体操作访问规则一样，操作客体访问规则也有两种类型，具体形式如下：

$oper\_obj\_deny(Y,Z)$:-$oper(Y,Y_1,Y_2,\cdots,Y_n)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$Y_{i1}\nabla Z_{j1},\cdots,Y_{ik}\nabla Z_{jk}$

$oper\_obj\_permit(Y,Z)$:-$oper(Y,Y_1,Y_2,\cdots,Y_n)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$Y_{i1}\nabla Z_{j1},\cdots,Y_{ik}\nabla Z_{jk}$

(3)主体客体访问规则：主体客体访问规则定义了主体和客体之间的可访问关系，也即客体可以被满足什么条件的主体所访问，什么条件的主体不能访问。具体形式如下：

$sub\_obj\_permit(X,Z)$:-$sub(X,X_1,X_2,\cdots,X_n)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$X_{i1}\nabla Y_{j1},\cdots,X_{ik}\nabla Y_{jk}$

$sub\_obj\_deny(X,Z)$:-$sub(X,X_1,X_2,\cdots,X_n)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$X_{i1}\nabla Y_{j1},\cdots,X_{ik}\nabla Y_{jk}$

(4)主体、操作和客体访问规则：主体、操作和客体访问规则定义了主体、操作和客体之间的访问关系，也有两种类型的规则：许可规则和拒绝规则，即主体可以对客体进行操作和不能对该客体进行操作。许可规则的形式如下：

$permit(X,Y,Z)$:-$sub\_oper\_permit(X,Y)$,$oper\_obj\_permit(Y,Z)$,$sub\_obj\_permit(X,Z)$

拒绝规则有两种描述方式：一种是强制拒绝，表示只要主体和操作、操作和客体或主体和客体单独拒绝访问，则主体就不可对客体进行操作；另一种是柔性拒绝，表示只有主体和操作、操作和客体、主体和客体同时拒绝访问时，主体、操作和客体才拒绝访问。它们的形式如下：

- $deny(X,Y,Z)$:-$sub\_oper\_deny(X,Y)$
  $deny(X,Y,Z)$:-$oper\_obj\_deny(Y,Z)$
  $deny(X,Y,Z)$:-$sub\_obj\_deny(X,Z)$
- $deny(X,Y,Z)$:-$sub\_oper\_deny(X,Y)$,$oper\_obj\_deny(Y,Z)$,$sub\_obj\_deny(X,Z)$

(5)主体、操作和客体访问许可扩展规则：主体、操作和客体访问许可扩展规则定义了主体、操作和客体之间的扩展访问关系，即主体属性之间和客体属性存在从属关系，比如角色层次、文件位置等。主体、操作和客体访问扩展规则有 3 种扩展形式：主体属性扩展、客体属性扩展以及主体和客体属性同时扩展。主体属性扩展和客体属性扩展规则可以用如下两条规则来描述：

$permit(X,Y,Z)$:-$permit(XX,Y,Z)$,$sub(X,X_1,X_2,\cdots,X_n)$,$sub(XX,XX_1,XX_2,\cdots,XX_n)$,$in(X_1,XX_1),\cdots,in(X_n,XX_n)$

$permit(X,Y,Z)$:-$permit(X,Y,ZZ)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$obj(ZZ,ZZ_1,ZZ_2,\cdots,ZZ_n)$,$in(Z_1,ZZ_1')$,$\cdots$,$in(Z_n,ZZ_n)$

主体和客体属性同时扩展可以用以上两条规则共同来描述。

(6)权限委托规则：权限委托规则定义了主体和主体之间的权限传递关系，即如果一个主体由于某种原因不能执行权限对应的事务，则可以把自己的权限委托给另一个主体来进行操作。权限委托规则有两种形式：一种是全部权限委托，即委托人把自己的权限全部委托给另一主体；另一种是部分权限委托，即主体只能把一部分权限委托给另一主体。两种形式的规则具体描述如下：

$permit(X,Y,Z)$:-$permit(XX,Y,Z)$,$delegate(XX,X)$

$permit(X,Y,Z)$:-$permit(XX,Y,Z)$,$delegate(XX,X)$,$oper(Y,Y_1,Y_2,\cdots,Y_n)$,$obj(Z,Z_1,Z_2,\cdots,Z_n)$,$Y_1\nabla y_1,\cdots,Y_n\nabla y_n,Z_1\nabla z_1,\cdots,Z_n\nabla z_n$

(7)主体约束规则：主体约束规则定义了主体必须满足的约束条件，具体的规则形式如下：

$error(system)$:-$sub(X_1,X_{11},X_{12},\cdots,X_{1n})$,$sub(X_2,X_{21},X_{22},\cdots,X_{2n})$,$conflict(X_{1i},X_{2i})$

(8)一致性约束规则：一致性约束规则描述策略的一致性要求，具体规则形式如下：

$error(system)$:-$permit(X,Y,Z)$,$deny(X,Y,Z)$

(9)完备性约束规则：完备性约束规则描述策略的完备性要求，具体的规则形式如下：

$error(system)$:-$\neg(permit(X,Y,Z),\neg(deny(X,Y,Z)$

规则的真假定义如下：

对于一个形如 L_o :-$L_1,\cdots,L_n$ 的规则为真，如果在情景 s 中有替换 θ，使得 $L_1\theta,\cdots,L_n\theta$ 为真，则 $L_o\theta$ 为真，否则为假。其中 L_i 为文字。

3.2 SCDAC 的动作理论

由 Reiter 和 Lin 的情景系统的动作理论[13,14]知，动作理论是一个公理集合，由 5 部分公理组成：基本公理、动作前提公理、后续状态公理、唯一名公理和初始情景公理。SCDAC 的动作理论也由这 5 部分组成。

(1)基本公理

基本公理描述了情景系统推理的基本理论，是与领域无关的。由文献[13,14]知，基本公理有 4 条，分别是

$$do(a,s)=do(a',s')\supset a=a'\wedge s=s' \tag{1}$$

$$\forall P.[P(S_0)\wedge\forall a,s.P(s)\supset P(do(a,s))]\supset\forall s P(s) \tag{2}$$

$$\neg s\subset S_0 \tag{3}$$

$$s\subset do(a,s')\equiv s\subseteq s' \tag{4}$$

其中，P 表示性质，$P(s)$表示在情景 s 下性质 P 满足；$\subset$是偏

序关系，用来刻画情景出现的先后次序，$s \sqsubset s'$表示从初始情景 S_0 到情景 s 的动作序列是到情景 s' 动作序列的子序列，即情景 s 比情景 s' 先出现，$s \sqsubseteq s'$ 与 $s \sqsubset s' \vee s = s'$ 等价。

公理(1)是情景唯一名公理，两个情景完全相同当且仅当它们是在同一个情景下执行同一个动作产生的；公理(2)是二阶归纳公理，要说明在任意情景下 P 都满足，首先要说明在初始情景 S_0 下 P 满足，然后对于任一动作 a 和情景 s，如果在情景 s 下 P 满足，都能得到 P 在情景 $do(a,s)$ 下满足；公理(3)表示 S_0 为初始情景，即任何情景都是从初始情景经过动作得到的；公理(4)表示如果情景 s 比情景 $do(a,s')$ 先出现，则情景 s 与情景 s' 相同或者情景 s 比情景 s' 先出现。

(2)动作前提公理

动作前提公理定义了情景系统中动作执行的前提条件。在 SCDAC 中，有两种类型的动作，其分别是增加、删除。动作前提公理定义如下：

$$Poss(add(sub,obj,des),s) \equiv sub = subject \wedge add = operation \wedge obj = object \wedge des = \text{fluent} \wedge s| = permit(sub,add,obj) \wedge s \cup \{des\} \ | \neq error(system) \quad (5)$$

$$Poss(del(sub,obj,des),s) \equiv sub = subject \wedge del = operation \wedge obj = object \wedge des = \text{fluent} \wedge s \ | = permit(sub,del,obj) \quad (6)$$

$Poss$ 是情景系统的保留符号，$Poss(a,s)$ 表示在情景 s 下动作 a 可执行。公理(5)表示动作 add 执行的前提是参数 sub 为主体，add 为系统定义的操作，obj 为客体，内容 des 为流，并且该动作在 s 下是允许执行的，而且增加流 des 后不能违反系统约束；公理(6)表示动作 del 执行的前提是参数 sub 为主体，del 为系统定义的操作，obj 为客体，内容 des 为流，并且该删除动作在 s 下是允许执行的。

(3)后续状态公理

后续状态公理定义了执行动作后情景流的变化。在 SCDAC 中，增加和删除动作的后续状态公理的定义如下：

$$Poss(add(sub,obj,des),s) \rightarrow s \circ des \quad (7)$$

$$Poss(del(sub,obj,des),s) \rightarrow s - des \quad (8)$$

$\circ$，$-$ 为系统保留符号，公理(7)是动作 add 的后续状态公理，$s \circ des$ 表示情景 s 中增加流 des；公理(8)是动作 del 的后续状态公理，$s - des$ 表示在情景 s 中删除流 des。

(4)动作唯一名公理

动作唯一名公理保证了动作命名的唯一性。根据文献[13,14]，具体描述如下：

$$A(x_1,\cdots,x_n) = A(y_1,\cdots,y_n) \supset x_1 = y_1 \wedge \cdots \wedge x_n = y_n \quad (9)$$

$$A(x_1,\cdots,x_n) \neq A'(y_1,\cdots,y_n) \quad (10)$$

式(9)、式(10)保证了动作命名的唯一性，动作名一样则一定是相同的动作，动作名不一样则是不同的动作。

(5)初始情景公理

初始情景公理描述了系统授权的初始状态，主要由关系流(逻辑事实集合)和规则流(逻辑规则集合)构成。

4 实例及分析

以下通过一个实例来说明应用基于情景演算的动态访问控制模型(SCDAC)描述系统授权和授权的变化过程。

假设 Billy 是系统安全管理员，他可以管理整个系统人员的权限。Alice 和 Bob 都是业务部的职员，Jack 是业务部的经理，Henry 是业务部的副经理。Alice 和 Bob 都能查询和修改自己制作的业务报表，业务经理可以处理自己的业务外，也能查看和修改部门职员制作的业务报表，但副经理不可以查看和修改员工制作的业务报表，只负责处理业务部其他事务，同时这些事务也可以由经理处理。

假设发生以下情况：

(1)权限委托：假设经理 Jack 出差，此时业务部的事务可能需要副经理 Henry 处理。授权情况该怎么描述？Jack 出差回来后，权限又有什么新变化？

(2)权限增加：假如业务部有新员工 Fred 进入，在不影响其他人的情况下，权限需要进行什么样的变更？

(3)权限修改：假设经理也不能处理业务员制作的业务报表，该怎么描述？权限总体变化情况是怎么样的？

此时，可利用情景演算系统来描述上述授权的变化过程。具体描述如下：

初始情景 S_0：

sub(billy,info,administrator)

sub(jack,sales,manager)

sub(alice,sales,staff)

sub(bob,sales,staff)

sub(henry,sales,vice_manager)

oper(read,business_report,sales,staff)

oper(write,business_report,sales,staff)

oper(add,permission_file,info,administrator)

oper(del,permission_file,info,administrator)

obj(file1,business_report,alice,X,Y)

obj(file2,business_report,bob,X,Y)

obj(file3,permission_file,X,info,administrator)

in(manager,staff)

sub_oper_permit(X,Y):-sub(X,X_1,X_2),oper(Y,Y_1,Y_2,Y_3),$X_1=Y_2$,$X_2=Y_3$

oper_obj_permit(Y,Z):-oper(Y,Y_1,Y_2,Y_3),obj(Z,Z_1,Z_2,Z_3),$Y_1=Z_1$

sub_obj_permit(X,Z):-sub(X,X_1,X_2),obj(Z,Z_1,Z_2,Z_3,Z_4),$X=Z_2$,$X_1=Z_3$,$X2=Z_4$

permit(X,Y,Z):-sub_oper_permit(X,Y),oper_obj_permit(Y,Z),sub_obj_permit(X,Z)

permit(X,Y,Z):-sub(X,X1,X2),sub(XX,XX1,XX2),in(X2,XX2),permit(XX,Y,Z)

permit(X,Y,Z):-permit(XX,Y,Z),delegate(XX,X)

(1)权限委托

假设经理出差，需要把权限委托给副经理。此时的权限变化可以通过动作 $add(billy, file3, "delegate(jack, henry)")$ 来实现，此时系统权限的新情景可以描述为 $s = do(add(billy, file3, "delegate(jack, henry)"), S_0)$。根据动作的后续状态公理，此时情景流集合中增加一条关系流 $delegate(jack, henry)$。

假设经理出差回来，需要收回权限，则可以通过动作 $del(billy, file3, "delegate(jack, henry)")$ 来实现。此时的新情景为 $s' = do(add(billy, file3, "delegate(jack, henry)"), s)$。

(2)权限增加

假设业务部有新员工 Fred 进入，此时的权限变化可以通过动作 $add(billy, file3, "sub(fred, sales, staff)")$ 来实现。新情景为 $s = do(add(billy, file3, "sub(fred, sales, staff)"), S_0)$。

(3)权限修改

假设经理也没有权限处理员工做的报表，此时可以通过

动作 $del(billy, file3, "in(manager, staff)")$来实现权限的变化。新情景为 $s = do(del(billy, file3, "in(manager, staff)"), S_0)$。

从以上实例可以得到，使用 SCDAC 可以描述访问控制的授权变化过程，而且可以通过具体动作来实现授权状态的动态变化。在具体应用时，可以根据具体的安全需求来定义动作的前提和后续状态变化。

结束语 随着资源复杂性和安全需求多样性的变化，信息系统的授权通常需要进行动态变化。现有的访问控制模型大多是静态模型，只能实现局部的动态性，比如通过定义条件来判断用户角色是否激活等，而且无法描述授权的变化过程。情景演算具有很强的动态系统描述能力，因此适合用于描述访问控制的授权变化。本文在此基础上提出了基于情景演算的动态访问控制模型(SCDAC)，它把系统在某一时刻授权的状态(逻辑事实和规则集合)看作是情景，通过动作来实现情景的变化，即授权状态的变化，并定义了动作实现的前提条件公理和后续状态公理。最后通过一个实例说明了使用 SCDAC 来描述系统授权变化过程的可行性。对动作的进一步扩展及模型的实际应用是需要进一步研究的工作。

参 考 文 献

[1] 韩道军，高洁，翟浩良，等. 访问控制模型研究进展[J]. 计算机科学，2010，37(11)：29-33

[2] 孙宇清. 协同环境中访问控制模型与技术研究[D]. 济南：山东大学，2006

[3] ANSI. American national standard for information technology-role based access control[S]. ANSI INCITS 359-2004，2004

[4] 窦文扬，王小明，张立臣. 普适环境下的动态模糊访问控制模型研究[J]. 计算机科学，2010，37(9)：63-67

[5] 裘炅，谭建荣，张树有，等. 应用角色访问控制的工作流动态授权模型[J]. 计算机辅助设计和图形学学报，2004，16(7)：992-998

[6] 刘道斌，白硕. 基于工作流状态的动态访问控制[J]. 计算机研究与发展，2003，40(3)：417-421

[7] Vassilis K，Loukas H，Dimitris K，et al. A Dynamic context-aware access control architecture for e-services[J]. Computer & Security，2006，25：507-521

[8] 方萃浩，叶修梓，彭维，等. 协同环境下 CAD 模型的多层次动态安全访问控制[J]. 软件学报，2007，18(9)：2295-2305

[9] 邓勇，张琳，王汝传，等. 网格计算中基于信任度的动态角色访问控制的研究[J]. 计算机科学，2010，37(1)：51-54

[10] 崔永泉，洪帆，龙涛，等. 基于使用控制和上下文的动态网格访问控制模型研究[J]. 计算机科学，2008，35(2)：37-41

[11] 姚寒冰，胡和平，卢正鼎，等. 基于角色和上下文的动态网格访问控制研究[J]. 计算机科学，2006，33(1)：41-44

[12] Sajjad A，Rohiza A. Design of Algorithm for Environment based Dynamic Access Control Model for Database Systems[J]. International Journal of Computer Applications，2011，21(10)：0975-8887

[13] Lin Fang-zhen. Situation calculus [M]// Lifschitz V，van Harmelen F，Porter F，eds. Handbook of Knowledge Representation，Elsevier，2007：649-669

[14] Reiter R. Knowledge in Action：Logical Foundations for Describing and Implementing Dynamic Systems[M]. MA：MIT Press，2001

[15] Bertino E，Bonatti P A，Ferrari E. TRBAC：A Temporal Role-based Access Control Model[J]. ACM Transactions on Information and System Security，2001，4(3)：191-233

[16] Joshi J B D，Bertino E，Latif U，et al. A Generalized Temporal Role-based Access Control Model [J]. IEEE Transactions on Knowledge and Data Engineering，2005，17(1)：4-23

[17] 邓集波，洪帆. 基于任务的访问控制模型[J]. 软件学报，2003，14(1)：76-82

[18] 许峰，赖海光，黄皓，等. 面向服务的角色访问控制技术研究[J]. 计算机学报，2005，28(4)：686-693

[19] 徐伟，魏峻，李京. 面向服务的工作流访问控制模型研究[J]. 计算机研究与发展，2005，42(8)：1369-1375

[20] 李晓峰，冯登国，陈朝武，等. 基于属性的访问控制模型[J]. 通信学报，2008，29(4)：90-98

[21] Park J，Sandhu R. The $UCON_{ABC}$ Usage Control Model [J]. ACM Transactions on Information and System Security，2004，7(1)：128-174

[22] Sandhu R，Park J. Usage Control：A Vision for Next Generation Access Control[C]// MMM-ACNS 2003. LNCS 2776，2003：17-31

[23] 王小明，付红，张立臣. 基于属性的访问控制研究进展[J]. 电子学报，2010，38(7)：1660-1667

[24] Dougherty D J，Fisler K，Krishnamurthi S. Specifying and reasoning about dynamic access-control policies[C]// 3rd International Joint Conference on Automated Reasoning (IJCAR). LNCS 4130，2006：632-646

[25] 陈宁，冯博琴. 一种基于情境演算的进化 MAS 系统[J]. 计算机研究与发展，2006，43(Suppl.)：147-151

[26] 刘一松. 面向主体的状态演算及其在智能虚拟人中应用的研究与实现[D]. 南京：南京理工大学，2009

(上接第 24 页)

[11] 张鳌. 基于 ANP 的科技规划决策支持系统研究[D]. 北京：清华大学，2008

[12] 许永平，杨峰，王维平. 一种基于 QFD 与 ANP 的装备作战需求分析方法[J]. 国防科技大学学报，2009，31(40)：134-139

[13] 汪彦明，徐培德. ANP 的指挥控制系统作战效能评估[J]. 火力与指挥控制，2007，32(11)：77-80

[14] 曾明露，严聪，蔡凯. 基于 GM-ANP 的维修人员保障能力评估[J]. 火力与指挥控制，2010，35(6)：77-80

[15] Mikhailov L，Singh M G. Fuzzy analytic network process and its application to the development of decision support systems[J]. IEEE Transactions on Systems，Man and Cybernetics(Part C：Applications and Reviews)，2003，33(1)：33-41

[16] 孙永河. 基于非线性复杂系统观的 ANP 决策分析方法研究[D]. 长春：吉林大学，2009

[17] 舒宇，谭跃进. 改进的网络分析法及其应用研究[C]// 中国系统工程学会决策科学专业第八届学术年会论文集. 2009：15-21

[18] 冯向前，魏翠萍，胡钢，等. 区间数判断矩阵的一致性研究[J]. 控制与决策，2008，23(2)：180-186

[19] 高洁，盛昭瀚. 可拓层次分析法研究[J]. 系统工程，2002，20(5)：6-11

[21] Yager R R. OWA aggregation over a continuous interval argument with applications to decision making [J]. IEEE Transactions on Systems，Man and Cybernetics-PartB：Cybern-Etics，2004，34(5)：1952-1963

[22] Yager R R. On ordered weighted averaging aggregation operators in multicriteria decision making[J]. IEEE Trans. on Systems，Man and Cybernetics，1988，18(1)：183-190