计算机科学1999Vol.26№.10

一种高效可靠的对象访问控制模型*)

An Efficient and Reliable Object Access Control Model

严 悍 张 宏 刘凤玉

(南京理工大学计算机系 南京 210094)

Abstract This paper is aimed mainly at large object multiuser systems, in which objects may have complex structure and relations. These systems mostly may be distributed and concurrency, and may have much more users holding diverse roles of access. An object access control model designing is presented in this paper using object modeling technology to ensure efficient, reliable and consistent access.

Keywords Object, Object-oritented modeling, Access control, Role-Based Access Control (RBAC), Model

大型分布式系统不仅类的数量大、结构复杂、分布范围广,而且用户多、职责多样、不同职责的用户对于不同的类及其对象乃至对象中不同的属性都可能具有不同的访问控制权限。此外,既有用户的访问控制,还需顾及其它系统对本系统的访问控制。因此信息系统中设计一个协调高效的访问控制是保证系统安全可靠运行的一个重要问题。

面向对象建模是对象技术中对大型复杂系统进行分析、设计的有效手段。统一建模语言(Unified Modeling Language, UML)[1]由 Booch 方法、OMT、OOSE 以及多种其他对象方法发展演变而来。UML 提倡用例驱动的、以体系结构为中心的、循环增长的、基于构件的软件开发过程。1995年 Grady Booch 和 James Rumbaugh 在 ACM OOPSLA'95会议上提出统一方法(Unified Method)。1996年对统一方法进行重新修订并更名为统一建模语言(UML)。1997年 UML 被提交 OMG(对象管理组)成为标准建模语言和规范方法。UML 已成为面向对象分析和设计的标准。

计算机系统与外界信息的交流是必然的,因此访问控制是信息系统中必不可少的部分。目前已开发多种访问控制技术,如 DAC(自由式访问控制)、MAC(强制式访问控制)、ACL(访问控制表)、LABC(基于网络的访问控制)、RBAC(基于角色的访问控制)等[2]。其中基于角色的访问控制(RBAC)是采用用户在机构中所担任的角色来决定其访问权限。RBAC 的研究和应用日益深入和广泛。RBAC 作为一个完整部分已应用于 SESAME(Secure European System for Applications in a Multi-vendor Environment)分布式系统的安全模型中。RBAC 也是数据库语言 SQL3的一个组成成分。此外,OMG 在 CORBA 安全规范中采用 RBAC 作为访问控制机制的一个示例,该机制用于 OMG 定义的分布式对象技术。ACM 在1995,1997和1998年召开了三次 RBAC 专题研讨会,内容涉及模型,规范、设计及实现等诸多方面[5,8-10]。目前多数研究与应用着重于数据库的访问控制,缺乏对象式应用系统的访问控制设计。

本文采用统一建模语言 UML 的概念和记法,把角色的概念和性质抽象为一个类及其关联,提出"角色转递"的概念以表示用户交互时角色权限的动态控制。基于角色和角色转递建立一种面向对象的访问控制模型。此模型与面向对象的分析、设计及实现的软件过程紧密结合。该模型设计遵循如下原则:

①高效:存储、修改和检查访问权限的代价应最小。

②可靠:对每个对象的每次访问均能保证其授权正确。

③通用:应能支持任一种特定的应用。

④最小权力:每个应用与每个用户使用完成其任务所必需的最小的权力来进行操作。

1 对象系统的访问控制

面向对象的开发是用例(use case)驱动式的,即面向对象的需求分析产生用例,用例指导系统的设计、实现和测试。用例是一个系统(或其他实体)与该系统的

*) 本文受到国家自然科学基金和江苏省自然科学基金资助。

作用者进行交互时所执行的动作序列的规范。一个用例实例表示在用例中所规范的动作序列的执行。用例中的作用者(actor)表示用户与用例交互时所承担的一组相关角色的集合。角色是一个实体在特定语境中被命名的特定行为[1]。一个作用者对于交互的每个用例具有一个角色。识别、描述并扩展用例是系统需求分析的主要内容。从应用的用例集合可识别系统中被保护的对象,确定角色的所有权力,并能严格满足最小权力原则。文[2]提出使用权力规范来扩展用例的方法以确定角色的权力。总之,面向对象的需求分析应确定应用中被保护的对象,以及各角色的权力,这是构造安全可靠系统的重要前提。

不同的应用系统中被保护的对象各异。如管理信息系统中着重于应用后台的永久化对象的保护,如OS中文件或DBMS中的数据库记录,这也是目前大多访问控制基于OS和数据库的原因。而其他系统如CSCW,也包含界面对象的保护。在对象式系统中,任何对象都是可保护的,而且界面对象的保护也可控制系统后台的永久化对象,并具有高效率。文[3]指出一个交互式系统仅有后台的访问控制的缺陷:①反馈滞后;②对话期独立绑定;③物理保护单元;④物理权力;⑤部分保护。此外,还有交互语言的差异,如我们基于某种英文版DBMS开发中文应用系统,当发生权限错误时往往返回英文信息。因此后台的访问控制不能高效而直观地适用于多用户交互式应用环境,我们更为强调界面对象的保护。

系统的用例指导模型设计及实现,即用例被设计实现以类、接口、属性、操作和方法。对象式系统中基本的访问权力是对于一个类是否可创建对象,即类的公用构造器是否可调用;对于接口所提供的操作是否可调用;若可调用,此操作在什么条件下可作用于哪些对象。虽然类的构造器与其操作和方法有区别,但从访问控制的角度,公用构造器可看作为类的一种特殊操作来加以控制。此外,因对象系统中类和接口具有封装性和内部访问控制权限,当不同角色对于同一对象的不同属性具有不同的控制时,可采用不同操作及方法来区别,从而避免对属性的直接访问。因此,对象式系统的基本访问控制就可归结为可执行哪些公用操作,及其许可条件。

从用例确定访问权力应保证其一致性和完整性。被保护的操作之间具有固有的复杂关系:隐含或前提关系、互斥关系等。一个操作往往隐含其他操作,如改变对象状态隐含着能读取对象状态。一个操作的执行须以其他操作为前提,如打印对象以读取对象状态为前提。互斥表示两个操作之间语义的差别使得两者不能被同一用户在同一个语境中调用。显然,两个操作之间隐含或前提关系与互斥关系不可共存。一个操作也可具有限定性,即该操作仅能由一定数量的用户执行。这些关系决定了权力的协调性。

2 模型设计原则

根据对象系统的访问控制特点,我们建立对象的访问控制模型的设计原则如下:

(1)许可而非排除:缺省为无权;显式许可才有权。这比缺省许可更安全可靠,且符合最小权力原则。

(2)预定义与自定义的类属权力:应提供尽可能多的预定义权力以方便实现。如创建对象、读取对象状态、输出或打印对象、改变对象状态、消除对象等权力。具体应用可扩展自定义权力或重定义权力。

(3)细粒度权力与对象:应能为每个公用操作定义单独的权力,且每个对象都可选择,保证其可靠性。

(4)显式许可与权力蕴含:某操作的权力应自动蕴含具有隐含或前提关系的操作权力,不仅可保持权力的一致性,而且减少管理负担,提高管理效率。对象式系统中一种合理的蕴含规则是:若对于某类或接口 c_i 有权,就隐含着对于 c_i 所继承的所有类或接口都有权;反之,若对于某类或接口 c_i 无权,就隐含着对于所有继承 c_i 的类或接口都无权。

(5)一般化与异常:若系统仅支持细粒度的对象和权力,那么权限管理代价巨大。模型不仅应制定一般化的权力作用于多个操作和对象,而且可确定其中的异常,以提高管理效率。在对象式系统中一种合理的一般化与异常规则是:对于某类或接口 c_i 若具有一般化权力,那么对于继承 c_i 的所有类或接口均具有权力,除非其中确定无权异常情形。

(6)悲观与乐观的访问控制:在运行时刻不同用户共享同一对象时,须考虑该对象在本地交互缓冲区的保护问题。当调用操作或提交本地对象时,根据是否检查操作的并发冲突,并发控制可分为悲观和乐观两种,其选择依赖于具体应用。

3 访问控制模型

由以上原则我们建立一个对象式系统的访问控制模型,如图1所示。

3.1 权限管理

模型中作用者表示在系统外部与之交互的所有实体的类;其实例包括所有可能的用户以及外部其它系统。事实上,本系统也可能作为其他系统的作用者。

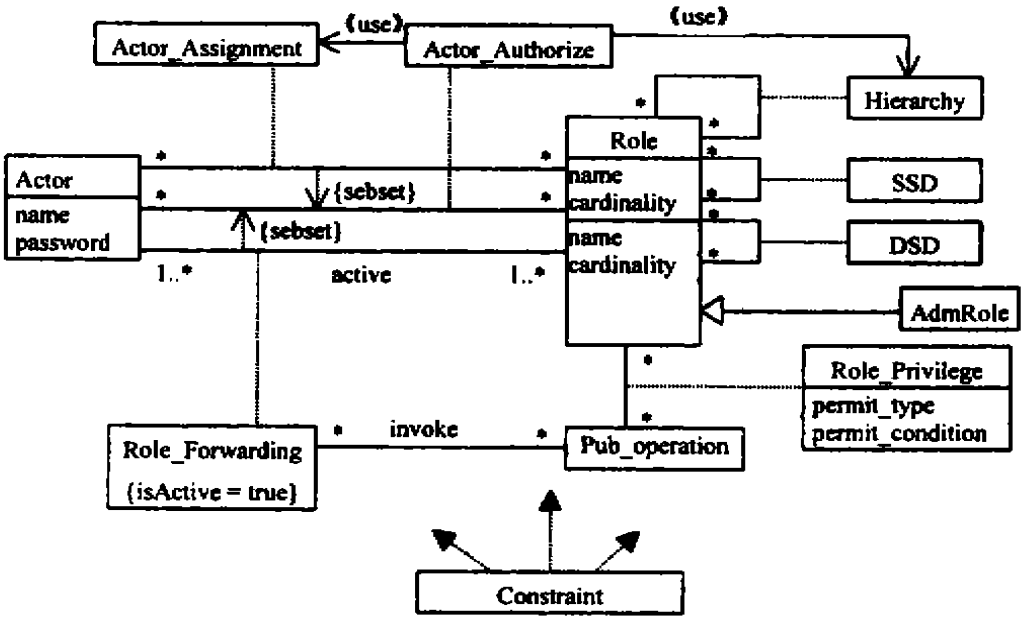

图1 基于角色和角色转递的对象访问控制模型

模型中我们定义角色为作用者在与系统元素的交互和协作的语境中被命名的特定行为。一方面表示机构中用户的职责划分；另一方面表示访问系统对象的权限集合。基于角色的访问控制可极大地简化权限管理，提高管理效率，并可避免出错[6]。角色实例之间具有关系：(1)层次，角色层次是构作角色的一种自然的方法，反映一个机构内职业的授权、责任和能力。高级角色的权力继承低级角色，并且可增加新权力。此关系是角色实例集合上的偏序关系，即自反、传递和反对称[4]。层次也可表示角色之间的前提关系。一个用户可被授予角色 A 仅当其已被授予角色 B，那么角色 B 是 A 的前提，角色层次可表示为 A 继承 B。(2)静态职责分割(Static Separation of Duties，SSD)是一种利益冲突关系，根据业务规则两个角色不可授予同一个作用者。(3)动态职责分割(Dynamic Separation of Duties，DSD)是另一种利益冲突关系，两个角色虽可授予同一个作用者，但在运行时刻不能同时被激活[5]。显然，SSD 与 DSD 均为非自反、非传递、对称关系。为简化管理，文[5]指出层次关系与 DSD 互不相交。

本模型中角色作为一个类，其实例为应用系统中的职业。如在一个外贸业务系统中角色实例可能是单证员、业务员、审核员、经理等。每个角色实例除具有一个名字(name)之外，还有一个基数(cardinality)约束，表示该角色可授予作用者的最大数目。管理员角色(AdmRole)是一种特殊的角色。角色之间的三种关系作为角色类的三个相互独立的关联类。

模型中角色权限(Role_Privilege)表示角色与对象系统中公用操作之间的关系；表示为角色类与公用操作之间的关联类。该类的一个实例表示一个角色是否可执行一个操作(或一般化权力)，即许可类型(permit_type)，以及许可条件(permit_condition)。此条件可确定操作的对象范围，也可能是其他条件，如操作时间限制等[7]。显然，两个互斥操作不能授权给同一角色；且若两个互斥操作授权给两个角色，则导致此两个角色利益冲突。隐含或前提关系的操作可授权给同一角色，也可通过角色层次来继承。

作用者指派(Actor_Assignment)表示作用者与角色之间的显式指派关系。作用者授权(Actor_Authorize)表示授权。若角色 r_1 被指派给作用者 a，或角色 r_2 继承 r_1 且 r_2 已被指派给 a，则角色 r_1 被授权给作用者 a。显然，前者是后者的一个子集；且角色层次关系影响后者。模型中两者都表示为作用者与角色之间的关联类。

基于角色的访问控制(RBAC)可有效管理大型复杂系统的用户权限。因系统中角色及其权限相对稳定，因而管理员通常的工作是确定雇员职务的任免升迁。只有当业务发生改变或机构重组时，才有必要调整角色或角色权限，必要时需调整操作。如此极大地简化了用户权限的管理，提高了管理效率，而且可避免发生错误，提高可靠性。

当用户被授权角色之后，就可与系统交互以完成其任务。为表示运行时刻用户、角色和操作之间的关系，我们提出角色转递的概念。

3.2 角色转递

本模型提出"角色转递"(Role Forwarding)的概念，表示在运行时刻作用者激活角色，并与之交互的关

系，以实现高效的访问控制。角色转递在运行时刻接受用户的交互事件，按照相应角色定义的权限，决定是否启动系统操作的调用事件。因其在角色权限的约束下转递用户的请求操作事件，故称为"角色转递"。

模型中角色转递作为一个主动类。一个角色转递实例是一个主动对象，持有自己的线程(事件接受器和观察器)，能启动操作调用，并能与其他主动对象并发运行[3]。模型中角色转递是作用者和角色之间的关联类。一个作用者同时可激活多个角色，故可持有多个角色转递。每个角色转递对应于一个作用者与一个角色。当作用者激活某角色时，即创建一个角色转递对象，并获取该角色所持有的权力并使权限可视化。在交互过程中，角色转递接受作用者的请求事件，并把符合权限的事件转递给系统操作调用；角色转递同时也受管理员的动态监控。此外，角色转递相互之间可通过其自身的线程相互通讯，可实现悲观或乐观的并发控制以解决访问冲突问题。当用户退出系统，其所持有的角色转递即被消除。

在基于网络的大型分布式系统中，若没有角色转递或访问控制仅作用于系统后台 OS 或 DBMS，要实现可靠性原则，即对每个对象的每次操作都保证其授权正确，就必须通过网络频繁传输授权验证的信息，如此必降低效率。采用角色转递，用户的权限在其本地得以控制，避免了网络传输所引起反馈滞后等问题，从而提高执行效率，并能通过自身的线程与其当前授权保持协调一致，而且管理员可动态监控，必要时可消除角色转递以保证系统安全可靠运行。因此，角色转递在与用户交互时使其权限可视化，以实现主动的控制。

总之，角色转递对于用户来说是一种交互界面；对于角色来说是被激活的角色或某种角色的一次执行；对于系统操作来说则是一种动态的接口。本模型基于角色和角色转递来设计和实现对象式系统的访问控制。

3.3 约束

模型中约束作用于各元素，以保证访问控制的可靠性、一致性和安全性。约束是一种语义条件或限制[1]。文[4]指出约束是 RBAC 最主要动机。约束对于设计和实施高层机构管理策略是一个有力的机制，尤其对于分布式管理，使高级安全员可限制管理员的权力。约束可表示为特定规范语言书写的语句。(约束的规范表示另文再述)模型中的约束关系如图2所示。

模型中一种重要的约束就是互斥。互斥约束可支持职责划分，是用一个有效方法来限制关键权力的分布。互斥关系的操作权力导致角色的利益冲突，即 SSD 或 DSD 的关系。SSD 将约束作用者指派 Actor_assignment 关系；DSD 将限制角色转递的创建。

基数约束是另一种限制关键权力分布的手段，是一种定量约束。限定性操作的权力被授权给角色的数目、角色授权给作用者的数目、以及可被同时激活的角色的数目都可能具有基数约束。

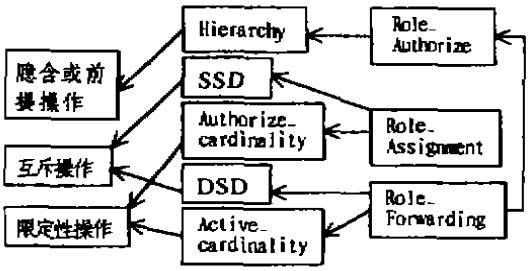

图2 模型中约束的依赖关系(箭头表示依赖关系，其反向表示制约关系)

操作的隐含或前提关系是一种约束以保持权力的一致性。角色层次结构本质上也是一种约束，一方面将操作权力的隐含或前提关系表示为角色继承结构；另一方面使作用者指派决定作用者授权，如此既保持一致性，又减轻管理负担。

此外，系统中业务对象自身对于用户在不同角色组合中的成员资格可判断是否合理，如外贸业务中一个用户在不同业务中可同时担任业务员和审核员，但在同一笔业务中则不可接受。

本模型支持在一个软件过程中设计和实施约束：给角色授予操作权限；给作用者授予角色；作用者激活角色；作用者调用操作。最终当某作用者调用某操作时是否许可必须是确定的、无随机性且满足约束。总之，访问控制的本质就是设计并实施约束，以保证其可靠性、一致性和安全性。

结束语 信息系统其存在的意义在于能给外界提供信息，因而交互式访问是系统重要的组成部分。因大型对象式系统中对象结构和关系复杂，用户量大且职责多样，现有的方法不能提供可靠且高效的访问控制。本文提出了一种基于角色和角色转递的对象访问控制模型，该模型适用于各种大型复杂的对象式系统的权限管理和动态监控，并可保证其具有可靠性和高效率。该模型已在多个系统中得到应用。

参考文献

1 UML Summary Version1.1，UML Semantics Version1.1，UML Notation Guide. Sep 1997. Available at：www.rational.com/uml

2 Fernandez E B，Hawkins J C. Determining Role Right from Use Cases，RBAC' 97. In：Proc. of the second ACM workshop on role-based access control. 1997. 121～125

(下转第56页)

时，$Q(s,a)$按下式公式进行更新[4]：

$$Q_t(s_t,a_t)=\begin{cases}(1-\alpha_t)Q_{t-1}(s_t,a_t)+\\ \quad\alpha_t[r_t+\gamma V(s_{t+1})] & s=s_t;a=a_t\\ Q_{t-1}(s_t,a_t) & \text{其它}\end{cases}\quad(13)$$

只要每一个变化经过无穷多次，当$t\to\infty$时，$\alpha_t\to 0$，$Q(s,a)$就会收敛于最优值。在式(13)中，$V(s_{t+1})=\max_{a\in A}\{Q_{t-1}(s_{t+1},a)\}$是由动作$a$产生的状态值；$\alpha_t$为学习速率；$\gamma$为折扣系数；$r_t$立即强化信号。由于这个结果只允许对被选中动作的V值进行更新，且更新可以实时完成，这个算法被命名为Q-学习算法。

3.3.4 *基于模型的算法* 另一种在环境中学习动作的方法是建立环境内部表示，也就是建模。利用学会的模型来选择适当的动作。在这种方法中学习的目的是产生对环境行为描述最好的逼近模型，有时称为动作模型。由于环境可以看成一个自动机，在给定初始状态和执行动作后模型应能够正确预测环境的变化。

强化学习Agent的外部环境模型在某种程序上模仿了环境的行为。例如，给定一个状态和一个动作，模型可以预测下一个状态和下一个强化值。模型可能要占据较大的存储空间，如果有$|S|$个状态和$|A|$个动作，那么整个模型将占据的空间与$|S|\times|S|\times|A|$成比例。这是由于它把状态-动作对匹配成整个状态空间的概率分布。相反，强化值和值函数把状态匹配成一个实数，其维数是$|S|$；而随机策略的维数是$|S|\times|A|$。

并不是所有的强化学习Agent都利用模型，从未使用模型的方法叫无模型强化学习方法，无模型方法非常简单，也许能令人惊讶地找到最优行为。基于模型的方法能够较快地找到最优行为(利用较少的经验)，最令人感兴趣的情况是Agent事先没有完美的模型，而是利用学习的方法来确定它。

与联想算法类似，基于模型强化学习系统的输入信息包括环境状态，强化值由强化模块R来计算。由于合适动作的选择依赖于模型的正确性，所以强化值是动作模型预测环境行为真实程度的一种测量，策略模块的内部状态W_t是环境模型的一种编码。

结论 本文讨论了强化学习系统的一般组成结构，并对结构模型进行了描述，对组成系统的输入模块、强化模块特别是策略模块的实现方法进行了研究，重点讨论了策略模块的实现方法。强化学习存在学习速度慢的问题。所以，如何提高强化学习系统的速度，是今后研究的一个重要问题。

参考文献

1 Sutton R S. Temporal Credit Assignment in Reinforcement Learning: [PhD thesis]. University of Massachusetts, Amherst, MA, 1984

2 Sutton R S. Learning to Predict by the Methods of Temporal Difference. Machine Learning, 1988, 3: 9～44

3 Sutton R S. The Challenge of Reinforcement Learning. Machine Learning, 1992, 8: 225～227

4 Watkins J C H, Dayan P. Q-learning. Machine Learning, 1992, 8: 279～292

5 Peng J, Williams R J. Increment Multi-Step Q-Learning. Machine Learning, 22: 283～291

6 Thrum S, Mitchell T M. Lifelong Robot Learning. Robotics and Autonomous System, 1995, 15: 25～46

7 Ilg W, Berns K. A Learning Architecture Based on Reinforcement Learning for Adaptive Control of the Walking Machine LAURON. Robotics and Autonomous System, 1995, 15: 32～334

8 阎平凡. 再励学习—原理、算法及其在智能控制中的应用. 信息与控制, 1996(1): 28～34

9 Barto A G, et al. Neuronlike Adaptive elements that can solve difficult learning control problems. IEEE Transaction on System, Man, and Cybernetics, 1983, 13: 834～846

10 Beom H B. A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning. IEEE Trans. on SMC, 1995, 25(3)

11 Moure A W, Atkson C G. Prioritized Sweeping: Reinforcement Learning with Less Data and Less Time. Machine Learning, 1993, 13: 103～130

(上接第91页)

3 Dewan D, Shen H. Controlling access in multiuser interface. ACM Transactions on Computer-Human Interaction, 1998, 5(1): 34～62

4 Sandhu R S, et al. Role-Based Access Control Models. IEEE Computer, 1996, 29(2), 38～47

5 Gavrila S I, Barkley J F. Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management. RBAC' 98. In: Proc. of the third ACM workshop on role-based access control. Fairfax, VA, 1998. 81～90

6 Ferraiolo D F, et al. Role-Based Access Control: Features and Motivations. In: Annual Computer Security Applications Conf. IEEE Computer Society Press, 1995

7 Giuri L, Iglio P. Role templates for content-based access control. RBAC' 97. In: Proc. of the second ACM workshop on Role-based access control. Fairfax, VA, 1997, 153～159

8 Radack B F. An Introduction to Role-Bases Access Control, NIST CSL Bulletin, 1995