基于同构化客体状态适配监控的RBAC扩展模型研究

周启海[1,2] 陈森玲[2] 靳 健[3] 黄 涛[1,2] 李忠俊[1,2] 林 珣[1,2]

(西南财经大学信息技术应用研究所 成都 610074)[1] (西南财经大学经济信息工程学院 成都 610074)[2]

(广州铁路集团公司广州机务段 广州 510010)[3]

摘 要 为了改进和提高角色访问控制(RBAC)的角色管理与权限配置效能,本文提出了一种基于同构化客体状态适配监控的RBAC扩展模型。该模型在RBAC的角色与权限之间新增设一个基于客体状态的同构化适配监控层,使不同客体状态所对应的角色及其权限建立起关联;并在客体状态发生转换而引起其相关角色和权限变更时,可通过观察者模式正确地及时捕捉、自动回应和动态适配这些转换和变更,以尽可能地避免访问冲突、阻绝非法访问;从而,能对各客体在不同状态下的安全访问与访问安全,实施更有序和高效的敏捷控制。

关键词 同构化,角色访问控制(RBAC),客体状态,适配监控,转换变更

Research on an Expanded Model of RBAC Based on Isomorphic Adapter-monitor of Objects' States

ZHOU Qi-hai[1,2] CHEN Sen-ling[2] JIN Jian[3] HUANG Tao[1,2] LI Zhong-jun[1,2] LIN Xun[1,2]

(Research Institute of Information Technology Application, Southwestern University of Finance and Economics, Chengdu 610074, China)[1]

(School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China)[2]

(Locomotive Depot of Guangzhou, Guangzhou Railroad Group, 510010, China)[3]

Abstract In order to improve and enhance the efficiency of role-managing and permission-assigning of traditional RBAC, an expanded model of RBAC based on isomorphic adapter-monitor of objects' states is proposed, which associates roles and corresponding permission while objects are in different states. When the states switch and accordingly alter the assignment of roles and permission, with Observer Pattern, the adapter-monitor can catch it in time, response it automatically and then adapt it dynamically, to avoid accessing conflict and prevent illegal access as possible, thus more efficient and agiler to control the access safety and its implement in different objects' states.

Keywords Isomorphic, Role-based Access Control (RBAC), Objects' states, Adapter-monitor, Transfer and alteration

1 引言

继自主访问控制(Discretionary Access Control,即DAC)和强制访问控制(Mandatory Access Control,即MAC)之后,为了更有效地防范侵害电脑信息(例如:泄漏、失窃、窜改、毁损等)、保障信息资源安全性与完整性,角色访问控制(RBAC,即Roles-Based Access Control)于20世纪70年代应运而生。RBAC技术[1]在通常身份鉴别基础上,进一步通过角色来简化用户及其权限关系,强化对计算机资源的访问请求控制,以保护主机系统和应用系统的信息安全,已成为当今信息系统安全的基本技术与重要措施之一。自2004年RBAC被接受为ANS标准[2]后,人们在各层次软件系统中,已不同程度地应用和实现了角色访问控制(例如:Windows NT、Linux等操作系统,Oracle、SQL server等数据库,Web方式下各级应用程序等)。随着逐步把RBAC应用扩展到面向XML文档、工作流、信息流或网格模型、基于域分层概念模型[3-7]等新需要,使RBAC既不断拓广其新应用领域,也不停面临着新问题挑战。例如:在传统RBAC模型中,使用诸如角色基数约束、静态职责和动态职责分离等约束限制,对角色和用户之间的联系建立必要约束,进而有效控制主体状态。但技术进步的普及推广速度和现实生活的需求发展速度,既使应用环境中可访问的客体资源常常增长快速、不可预料,也使客体资源的访问控制及其冲突问题往往不时发生、难以避免。因此,如何更有序地控制客体对象的访问需求,更有效地实现客体对象的动态授权和角色管理,是一个亟待解决的RBAC应用新问题。为解决这一新问题,改进和提高角色访问控制(RBAC)的角色管理与权限配置效能,本文提出了一种基于同构化客体状态适配监控的RBAC扩展模型。

2 客体、角色与权限基本关系

在对RBAC进行模型扩展研究时,须全面认识和正确分析客体、角色与权限三者间所存在的"一对一"、"一对多"、"多对一"、"多对多"四种基本关系。一般说来,客体、角色与权限随其所处场景的空间差异与时间不同,而往往可有这四种基本关系中的一种或多种:

(1)角色与权限间的基本关系。一方面,"同一角色,可拥有不同权限",而"同一角色,可拥有一种或多种权限";另一方面,"同一权限,可由不同角色拥有",而"同一权限,可由一个

周启海 教授,博(硕)士生导师,主要研究方向为计算几何、算法研究与应用、财经计算、同构化信息处理等;**陈森玲** 硕士研究生,主要研究方向为信息安全;**靳 健** 技术员,主要研究方向为计算机应用;**黄 涛** 讲师,主要研究方向为计算机应用;**李忠俊** 讲师,主要研究方向为电子商务等;**林 珣** 讲师,主要研究方向为计算机应用。

或多个角色拥有”。

(2)客体与角色间的基本关系。一方面,“同一客体,可允许某个不同角色操作”,且“同一客体,可能允许一个或多个角色同时操作”;另一方面,“同一角色,可操作某个客体”,而“同一角色,可能操作一个或多个客体”。

(3)客体与权限间的基本关系。一方面,“同一客体,可存在不同权限”,且“同一客体,可能同时存在一个或多个权限”;另一方面,“同一权限,可作用不同客体”,而“同一角色,可能作用于一个或多个客体”。

显然,这就决定了客体、角色与权限三者间的关系必然多种多样,且常随计算机系统越大就越会呈现出错综复杂、纷繁多变的复杂景象。因此,如何对客体、角色与权限三者间的关系进行同构化简约与归并,是更有序地控制客体对象的访问需求,更有效地实现客体对象的动态授权和角色管理的主要瓶颈之一。

事实上,从“凡同类事物,客观上总存在着使其结构本质与构造实质趋于或达到一致的自然规律”的同构化基本原理[8]来看,客体、角色与权限三者间关系的本质应当是:角色与其权限,要求和决定了客体及其行为;而客体的行为方式,则受制于各客体当前自身状态。因此,从根本上说,可利用同构化的客体状态来控制角色与其权限的同构化适配与监控。

3 RBAC扩展模型的基本架构

本RBAC扩展模型的基本思想与研究思路是:在RBAC的角色与权限之间新增设一个基于客体状态(Objects' States)的同构化适配监控层,使不同客体状态所对应的角色及其权限建立起关联;并在客体状态发生转换而引起其相关角色和权限变更时,可通过观察者模式的一对多依赖关系(即:当一个对象的状态发生改变时,所有依赖于它的对象都得到通知并被自动更新)[9]正确地及时捕捉、自动回应和动态适配这些转换和变更,以尽可能地避免访问冲突、阻绝非法访问;从而,能对各客体在不同状态下的安全访问与访问安全,实施更有序和高效的敏捷控制。

3.1 基本概念定义

1)客体(Objects):接受权限操作和角色访问的对象。它一般是数据表、文件、设备、应用程序等。其形式多样化,决定了其不同的客体状态(Objects'States)特征。

2)用户(Users):一个可以独立访问系统数据或数据表示的其他资源的主体,是发出访问操作和存取要求的主动方。它一般是人为者,但也可为 Agent 等智能程序。

3)角色(Roles):应用领域内一种权力和责任复合体的语义综合体。它可以是一个抽象概念,也可以是对应于实际系统中的特定语义体(比如:组织内部的职务等)。

4)权限(Permission):特约者(例如:角色)可对系统资源进行特定模式访问操作的授权范围前提。它通常与实现机制密切相关。

5)许可(Permissions):准许角色有权对计算机资源进行其权限范围内的访问和操作。它反映的是授权的结果。在传统RBAC模型中它描述的是角色和权限之间的一种关联关系,即某一角色对某一对象所具有的权限及权限状态。

6)指派(Assignment):指派者为被指派者设定的授权任务或任务职权,并分为用户指派和许可指派。用户指派是表示用户与角色的指派关系,而许可指派表示的是为角色指派计算机资源的访问和操作权限。

7)会话(Session):用户、角色和权限三者间动态关系的通信接口。用户请求必须通过建立会话来激活角色,并得到相应的访问权限。

8)约束(Constraints):用户与角色、角色与权限之间许可相应操作的限制条件。例如:角色互斥约束,最大基数约束,先决条件约束,最小权限约束,时间频度约束等。

3.2 总体基本架构

基于同构化基本原理所建立的本RBAC扩展模型,一改传统RBAC的“角色←→权限”二级直接联系控制模式,而为“角色←→客体状态←→权限”三级间接联系控制模式。它新引进了客体状态作为“角色与权限”二者的同构化适配监控器,来间接而高效地同构化监控并适配二者间的关联,以精准区分客体不同状态下对应的角色和权限。通过客体状态这一同构化接口,不仅可统一描述和处理计算机系统多种多样的访问客体,而且能分别与相关角色、权限建立同步适配关系,使系统权限管理在客体层面上得以细化;同时,可利用观察者模式对客体状态实行实时监控,能及时跟踪、发现、提醒和适配因状态变更而引起的角色变化和权限转移,配合业务逻辑重新分配角色和权限,从而有序、高效地控制客体访问,尽可能地减少了访问冲突。本RBAC扩展模型的总体基本架构,可示意如图1。

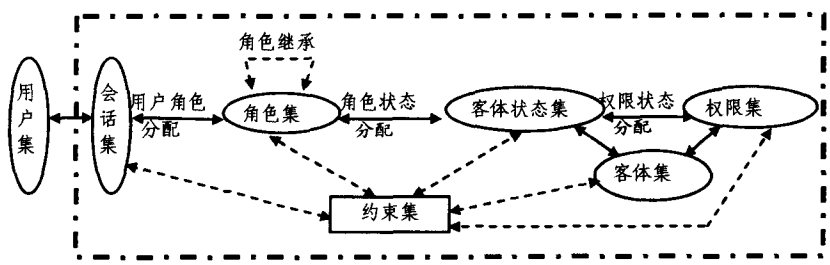

图1 RBAC扩展模型总体基本架构示意图

用户集 $U=\{u_1,u_2,u_3,\cdots,u_i,\cdots,u_n\}(1\leqslant i\leqslant n,n$ 为系统中用户总数)。其中,u_i 表示系统中第 i 个用户 $u_i(id_i,name_i,other_i)$,id_i、$name_i$、$other_i$ 分别为用户 u_i 的认证序列号、用户名、其它必要信息。显然,通过 id_i 可唯一确定系统中各用户的合法身份。

角色集 $R=\{r_1,r_2,r_3,\cdots,r_i,\cdots,r_m\}(1\leqslant i\leqslant m,m$ 为系统中角色总数)。其中,r_i 表示系统中第 i 个角色 $r_i(id_i,name_i,hierarchy_i,priorhierarchy_i)$,$id_i$、$name_i$、$hierarchy_i$、$priorhierarchy_i$ 分别为系统中角色 r_i 的序列编号、角色名称、角色在系统层次中的等级编号(譬如:超级管理员默认值通

常设置为0)、本角色所继承的前驱角色的等级编号。通过 id_i 可唯一确定各角色特征，而此数据模型实现时可直接使用本表直接记录角色继承信息，且利用表内递归可实现其无级分类。

客体集 $O=\{o_1,o_2,o_3,\cdots,o_i,\cdots,o_h\}(1\leqslant i\leqslant h,h$ 为系统中的客体总数)。其中，o_i 表示系统中接受权限操作和角色访问的第 i 个客体 $o_i(id_i,name_i,other_i)$，id_i、$name_i$、$other_i$ 分别为客体 o_i 的认证序列号、客体名称、其它必要信息。显然，通过 id_i 可唯一确定系统中不同客体。

客体状态集 $OS=\{os_1,os_2,os_3,\cdots,os_i,\cdots,os_k\}(1\leqslant i\leqslant k,k$ 为系统中所有客体的状态总数)。其中，$os_i^{(j)}$ 表示客体资源 j 在系统中可能存在的第 i 种状态 $os_i^{(j)}(id_i,name_i,advisble_i,objectid_j,objectname_j)(1\leqslant j\leqslant h,h$ 为系统中的客体总数)。id_i、$name_i$、$objectid_j$、$objectname_j$ 分别表示客体状态唯一标识码、客体状态名称、对应客体唯一标识码、对应客体名称。此外，$advisble_i$ 表示目前系统中客体 j 的第 i 种状态可否被激活标志，由系统实时监控与反应。一般来说，同一会话过程中只允许一个状态可被激活。客体状态集，具体描述不同访问对象在系统中所呈现的不同状态(如数据表和文件的读写状态、设备的占用及空闲状态、不同模块调用时的应用程序结果状况、不同工作时期的资源调度状况等)，故它由客体 id 与状态 id 同时确定。

权限集 $P=\{p_1,p_2,p_3,\cdots,p_i,\cdots,p_l\}(1\leqslant i\leqslant l,l$ 为系统中所有客体允许的权限总数)。用 $p_i^{(j)}$ 表示系统中客体 j 所允许的第 i 个权限操作 $p_i^{(j)}(id_i,name_i,objectid_j)(1\leqslant j\leqslant h,h$ 为系统中的客体总数)，id_i、$name_i$、$objectid_j$ 分别表示权限唯一标识码、权限名称、对应客体唯一标识码。

约束集 $C=\{c_1,c_2,c_3,\cdots,c_i,\cdots,c_x\}(1\leqslant x,x$ 为系统中所需约束条件个数)。用 c_i 表示依据业务逻辑，针对用户、角色、客体状态以及会话过程建立的各种约束，在实现过程中可用程序逻辑作用于上述集合和过程来体现；它不作为本模型的重点环节，故不予详述。

会话集 $S=\{s_1,s_2,s_3,\cdots,s_i,\cdots,s_y\}(1\leqslant x,x$ 为系统中所需会话表现个数)。其中 s_i 表示系统中在用户提出访问请求时动态建立的，涉及到下述“角色状态分配集 UA、角色状态分配集 ROS、权限状态分配集 POS”三者之间的调度与分配，完成客体不同状态下有效角色的权限许可，同时在客体状态发生转换时调整角色与权限的配置是否可被继续等。

用户角色分配集 $UA\subseteq U\times R$：描述用户与角色的分配关系，通常为多对多关系，即一个用户可隶属于多种角色(互斥角色除外)，一个角色可拥有多个用户成员(一般以用户集最大元素个数作为限制)。

角色状态分配集 $ROS\subseteq R\times OS$：描述角色与客体状态的分配关系，通常为多对多关系，即一个角色在某个客体哪些状态下有效以具有操作权限，同时客体的某个状态可允许哪些角色对其进行操作；

权限状态分配集 $POS\subseteq P\times OS$：描述权限与客体状态的分配关系，通常为多对多关系，即一个权限在某个客体的哪些状态下可被激活，同时客体的某个状态可允许哪些权限操作。

4 RBAC扩展模型的监控适配

本RBAC扩展模型的角色及其权限的监控适配，分为系统资源的静态适配与动态适配。

4.1 系统静态适配过程

本RBAC扩展模型，遵照通用原则及实现逻辑；其建立权限机制并完成系统资源的静态适配过程，可简述如下：

1) 系统集合建立：按照业务逻辑建立系统的上述各类集合，即在满足实际约束的条件下确定系统合法用户、参与分配的角色、存在的客体状态以及所允许的权限操作，为其制定唯一编码，以及其它必要属性。为描述它们：其角色集 R，应需确定相应的层次等级信息；其客体状态集 OS，要能描述各种客体所拥有不同状态；其权限集 P，要能描述各种角色可能拥有不同权限。系统中，观察者模式除可实时监控客体状态外，在转换状态时还可切换客体激活信息(即：转为另一状态时，原客体状态不可被激活)。

2) 用户角色指派：根据系统要求及业务划分原则，在确保角色的互斥限定、集合上限等约束情况下，寻找每一合法用户所隶属的角色，形成用户角色分配集 UA，完成用户角色指派。

3) 客体状态指派：根据现有逻辑要求，在客体状态分别建立对应状态下的角色分配与权限设置，其中用户角色状态分配集 ROS 描述所有客体状态下的有效角色，权限状态分配集 POS 描述所有客体状态下的权限许可，并以其客体状态为桥梁，建立各角色与其权限的静态关系，从而完成角色、权限间的适配与指派。

至此，在系统权限机制形成的同时，也完成了各角色与其权限的系统静态资源适配。

4.2 系统动态适配过程

在静态系统资源适配完成后，系统必须对角色变迁会话，进行动态实时监控(如图2所示)；图2中，虚线部分表示：系统可通过观察者模式始终监控客体状态，通过客体的观察者接口保证客体状态的及时通知，通过角色与权限的更新接口保证对其的动态响应。

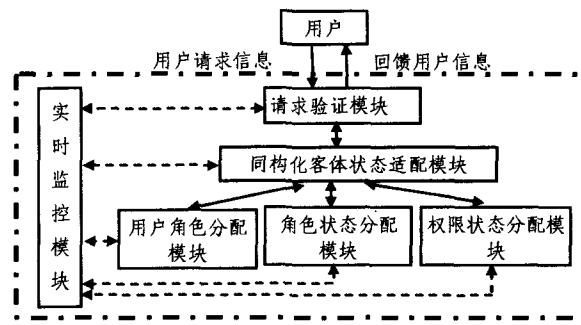

图2 用户请求与回馈的会话监控过程机制示意图

本RBAC扩展模型的会话监控过程机制，可概述如下。

1) 用户提出请求前，同构化客体状态适配模块负责建立用户集 U、客体集 O、客体状态集 OS 及权限集 P；用户角色分配模块完成用户角色指派，建立用户角色分配集 UA；角色状态分配模块及权限状态分配模块完成客体状态指派，建立用户角色状态分配集 ROS 和权限状态分配集 POS；

2) 当一个用户登录系统后，提出客体及权限访问请求时，系统首先调用请求验证模块记录用户信息、客体信息及权限信息，验证用户是否合法、客体是否可用及权限是否存在，如果有一项无效，则返回用户请求失败信息，否则进行下一步；

3) 调用搜索用户角色分配集 UA，确定用户隶属角色；

4) 调用同构化客体状态适配模块，确定此时该客体处于何种可被激活状态；

(下转第219页)

4.4 礼花模拟过程

根据上述的分析，利用粒子系统模拟礼花生成的过程就是对 P 组礼花粒子的属性进行不断更新的过程。为了保证礼花的模拟过程具有实时性和真实感，需要以一定的时间间隔更新每组粒子的属性。图 3 所示为本文算法在一个时间单位内对每组粒子所进行的操作。

5 实验结果与分析

为了验证所提出算法的有效性，对模拟礼花的过程进行了实验，实验中，参数 $M=100$，$N=0.05$，$P=60$。模拟是在配置为 Pentium(R)4 CPU2.66GHz、DDR256M 内存及 Geforce 5200(64M)显卡的 PC 机上进行的。实验结果如图 4 所示。从图中可以看出，在模拟礼花时能具有较逼真是的效果，并且实时性也较好，画面流畅达到 30 帧/秒。

结束语 对于粒子系统而言，用传统的方法建立粒子，粒子的显示已经很消耗时间和资源，加之采用传统画面绘制算法将导致复杂的消隐计算。当粒子数很大时，实时性往往达不到。本文设计合适的二维粒子纹理面片，应用 OpenGL 的纹理映射技术等对礼花进行模拟，能够实时显示礼花，在兼顾实时性与逼真性方面取得不错的效果。考虑到较强的真实感效果，应该在以后的工作中把风力等外力作用考虑进去，使这个系统更逼真。

参 考 文 献

[1] Reeves W T. Particle Systems-A Technique for Modelling a Class of Fuzzy Objects[J]. Computer Graphics, 1983, 17(3): 359-376

[2] 张芹. 基于粒子系统的建模方法研究[J]. 计算机科学，2003，30：144-146

[3] Peachey D. Modeling waves and surf. ComputerGraphics[J]. 1986, 20(4): 65-74

[4] Fournier A, Reeves W T. A simple model of ocean waves[J]. Computer Graphics, 1986, 20(4): 75-84

[5] Goss M E. A real time particle system for display of ship wakes [J]. IEEE Computer Graphics and Applications, 1990, 10(3): 30-35

[6] Wan Hua-Gen, Jin Xiao-Gang, Peng Qun-Sheng. Physics based real time animation of fountain[J]. Chinese Journal of Computers, 1998, 21 (9): 774-779(in Chinese)

[7] Simon P, Michael A. Rendering natural waters[C]// Proc. of Pacific Graphics. 2000: 23-30

[8] 王静秋，钱志峰，基于粒子系统的焰火模拟研究[J]. 南京航空航天大学学报，2001，33(2)：166-170

[9] 丁纪云，陈利平，李思昆. 基于 OpenGL 的烟花动态模拟方法的研究与实现[J]. 计算机工程，2002，28(4)：240-241，275

[10] 唐荣锡，汪嘉业，彭群生，汪国昭. 计算机图形学教程. 科学出版社，1990

[11] Angel E. OpenGL 程序设计指南. 李桂琼，张文祥，译. 清华大学出版社，2005

[12] 张英杰，赵汝嘉. 一种实现正向纹理映射的新方法[J]. 工程图学学报，1997 (1)：161-165

[13] 刘耀周，张锡恩. 基因粒子系统的导弹飞行航迹及烟雾的特效生成[J]. 计算机工程，2004，30：174-176

(上接第 215 页)

5) 调用角色状态分配模块，搜索角色状态分配集 ROS，并判定用户所属角色在当前客体状态是否有效：若无效，则返回用户请求失败信息，否则进行下一步；

6) 调用权限状态分配模块，查找状态权限分配集 POS 中当前客体状态下允许的权限操作与访问请求是否匹配：若不匹配，则返回用户请求失败信息，否则进行下一步；

7) 实时监控模式以观察者模式为设计思想，即：以客体为被观察的目标对象，以角色和权限为其观察对象。当客体的状态发生改变时，它的观察者们将得到通知，并自动、连锁地做出相应的响应行为。在整个会话过程中，如果客体状态没有发生转换，则依原序执行用户请求的权限操作；否则，由实时监控模块检测到的客体状态变化信息将被及时返回给同构化客体状态适配模块，同时更新客体状态集 OS 中的状态信息，连锁激活相关状态，并通知角色状态分配模块及权限状态模块，而与之对应的角色和权限也将被重新适配，系统进程将返回本小节步骤“4)”。

至此，完成一个用户访问的会话监控过程，也就完成了该角色与其权限的系统资源动态适配。

结束语 本文依据同构化基本原理，提出了一种基于同构化客体状态适配监控的 RBAC 扩展模型。它从角色访问控制的客体(即被访问对象)出发，在其角色与权限之间新增设一个基于客体状态的同构化适配监控层，使不同客体状态所对应的角色及其权限建立起关联和适配机制，可在客体状态发生转换而引起其相关角色和权限变更时，通过观察者模式正确地及时捕捉、自动回应和动态适配这些转换和变更，以尽可能地避免访问冲突、阻绝非法访问；从而，能对各客体在不同状态下的安全访问与访问安全，施行更有序、高效的实时敏捷控制。本模型仅为客体状态概念模型，它意在突出不同客体状态的访问控制方式，侧重于客体状态的设置、状态转换时的信息流动及转换后的权限调度与适配监控。对可能引起适配调度失败的各类原因(诸如：权限不足、客体访问冲突、客体状态约束等)及其解决方案，本文尚未作深入讨论；而这些正有待我们同更多学者一起进行其后续研究。

参 考 文 献

[1] Sandhu R, Coyne E, Feinstein H, et al. Role-based Access Control Models[J]. IEEE Computer, 1996, 29(2): 38-47

[2] American National Standard 359-2004 is the Information Technology Industry Consensus Standard for RBAC[EB/OL]. http://csrc.nist.gov/rbac/, 2005-10-20

[3] 耿晖，王海波. 基于 XML 的角色访问控制(RBAC)[J]. 计算机应用研究，2002，19(12)：14-15，42

[4] 洪帆，赵晓斐. 基于任务的访问控制模型及其实现[J]. 华中科技大学学报(自然科学版)，2002，30(1)：17-19

[5] 刘益和，刘嘉勇. 一个基于角色的信息流模型及应用[J]. 四川大学学报，2004，36(5)：236-243

[6] 孙鹏，杨德婷，等. 网格环境下动态访问控制模型的研究与实现[J]. 计算机工程与应用，2007，43(11)：132-134

[7] 韦超鹏，韩继红，王亚弟. 一种基于域分层概念的 RBAC 模型[J]. 微计算机信息，2006，22(03)：296-298

[8] 周启海. C++同构化对象程序设计原理[M]. 北京：清华大学出版社，北方交通大学出版社，2004，2

[9] Gamma E, Helm R. 设计模式：可复用面向对象软件的基础[M]. 北京：机械工业出版社，2000，2