

基于模型驱动的安全访问控制的研究

张赛男, 乔正洪, 雷小宇, 王梅娟

(中国人民解放军理工大学 理学院, 江苏 南京 211100)

摘 要: 随着计算机应用到各个领域的广度、深度不断扩大, 软件的安全需求变得越来越复杂. 然而开发人员对系统进行建模时一般不考虑安全需求, 系统安全策略和安全机制往往是开发人员在系统开发后期对系统的补充和措施的完善. 由此提出将访问控制需求集成到软件开发过程的分析阶段, 将访问控制策略建模与功能需求建模集成. 通过模型驱动方法为功能需求模型相关的安全方面提供模型, 研究从功能模型产生安全模型, 最终形成安全访问控制策略.

关键词: 模型驱动; UML; 访问控制策略; 安全建模

中图分类号: TP311; TP309

文献标识码: A

文章编号: 1000-7180(2011)08-0136-04

Research of Security Access Control Based on Model-Driven

ZHANG Sai-nan, QIAO Zheng-hong, LEI Xiao-yu, WANG Mei-juan

(Institute of Science, PLA University of Science and Technology, Nanjing 211100, China)

Abstract: Along with the development of the depth and width in applying computers into all kinds of domains, the software's security requirements become more and more complicated. However, the present software system modeling does not deal with security. System security policy and mechanism often become supplement at the end of system development. This paper investigates how to integrate access control requirements into analysis phrase of the development process and how to integrate the modeling of access control policies into the UML modeling of functional requirements. Through the approach of model driven, the paper provides models for security aspects related to the models for functional requirements and investigates how the security models can be generated from the functional models. Finally security access control policies will be generated.

Key words: model-driven; UML; access control policy; security modeling

1 引言

随着计算机应用到各个领域的广度、深度不断扩大, 软件的安全需求变得越来越复杂. 然而开发人员对系统进行建模时一般不考虑安全需求, 系统安全策略和安全机制往往是开发人员在系统开发后期对系统的补充和措施的完善, 这主要有两方面的原因: 第一, 安全需求难于分析和建模; 第二, 软件设计人员缺少安全软件开发所需的专业技术. 因为安全策略依据非常专业的安全模型来说明, 而这些安全模型没有和软件功能模型集成.

为了解决上述问题, 文中采用了国际对象管理

组织(OMG)提出的一种全新的软件开发框架(Model Driven Architecture, MDA). MDA技术将软件系统建立在各种模型的基础上, 通过模型的变换来驱动系统的开发, 它在建模这一层次上设计轻便的可操作的应用程序. 采用MDA为分布式系统的异构模型资源的集成提供了一个可扩展的、开放的体系结构.

本文研究将访问控制需求集成到分布式系统开发过程的分析阶段, 将访问控制策略建模与功能需求的UML建模集成. 通过模型驱动方法为功能需求模型相关的安全方面提供模型, 研究从功能模型产生安全模型, 最终形成由视图策略语言(VPL)描

述的安全访问控制策略。

2 系统中的访问控制

针对分布式系统的访问控制策略的设计和管理, Gerald^[1]提出了视图的概念,并用视图策略语言 VPL(View Policy Language)来管理访问控制策略。与数据库中的视图不同的是,这里的视图表示一个对象类型的操作集。

基于视图的访问控制 VBAC^[2] (View-based Access Control), 可以看成是基于角色的访问控制 (RBAC) 的扩展, 图 1 所示为 VBAC 的简易模型。从图 1 可以看出 VBAC 中的实体是主体、角色、视图和对象。一个主体可以扮演不同的角色或者一个角色由一些主体扮演。一个角色可以有一些视图或者一个视图可以为不同的角色所有。但一个视图总是对应了一个对象, 而一个对象可以有多个视图。基于视图的访问控制决策过程是: 当一个主体(或者角色)视图通过其拥有的操作访问一个对象时, 如果主体(或角色)所拥有的视图包含了这一个操作, 且操作对象属于视图的控制对象类型时, 访问被允许, 否则被拒绝。

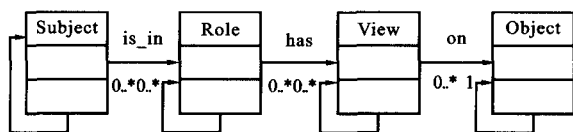


图1 VBAC 的简易模型

图 1 中相同类之间的关联表现出的更多的是继承关系, 例如角色之间、视图之间的继承关系。

3 模型驱动安全分析

针对如何将访问控制策略建模与功能需求的 UML 建模集成, 本文提出了一种模型架构, 既体现模型驱动的思想, 同时又能实现安全访问控制需求与开发过程的集成。

图 2 中描述的是被用于表示和集成安全需求所需要的模型以及它们之间的联系。左边表示的是抽象安全模型(ASM), 右边是功能模型。ASM 由系统开发过程中使用的安全访问控制模型来决定, 用于描述独立于任何应用以及平台的安全模型。因此 ASM 可以是基于角色的模型、强制模型或者这些模型的组合, 等等。软件开发人员依据应用环境及需求来选择具体的 ASM。

文中使用基于视图的访问控制模型来支持访问控制策略的设计和管理。具体的 ASM 由以下模型

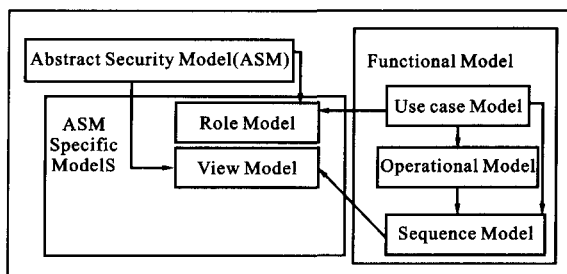


图2 安全模型与功能模型的集成

组成。(1) Role Model: 角色模型, 包含了用例模型中的访问控制角色。(2) View Model: 视图模型, 包含了视图中的被调用的操作所形成的访问权限。视图中规定的可以访问的权限由时序模型产生。

图 2 右边的功能模型由用例模型、操作模型和时序模型组成。

3.1 功能分析

3.1.1 用例模型

文中使用备忘录系统来阐述模型驱动安全分析。图 3 是系统用例图。用例图中有三个角色 Staff, Manager 以及 Secretary, 其中 Manager、Secretary 继承了 Staff。用例图的非形式化的描述如下:

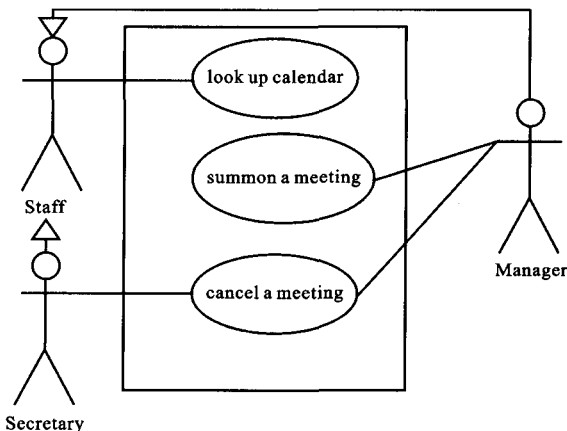


图3 系统用例模型

look up calendar: 职员只能查看自己的备忘录, 秘书可以查看自己和经理的备忘录。

summon a meeting: 仅有经理能召集与会人员参加某一时间的会议, 如果所有的与会人员同意参加会议, 这个会议才有效。经理将有效的会议添加到自己的备忘录中。

cancel a meeting: 经理或者秘书可以取消会议。

3.1.2 操作模型

软件开发过程中, 对象模型可以表达系统的需求。分析阶段的对象建模包含了对象类, 这些对象类表示现实世界的对象以及对象之间的关系^[3]。由于基于 VBAC 模型的访问控制策略仅需要被调用对

象的操作,在执行访问控制的过程中,不考虑对象数据.因此操作模型中省去了数据,仅留下对象的操作.基于相同的原因,也不考虑对象操作的参数以及返回值.操作模型包含了带有操作的对象以及对象之间的关系.图4所示为备忘录的操作模型.

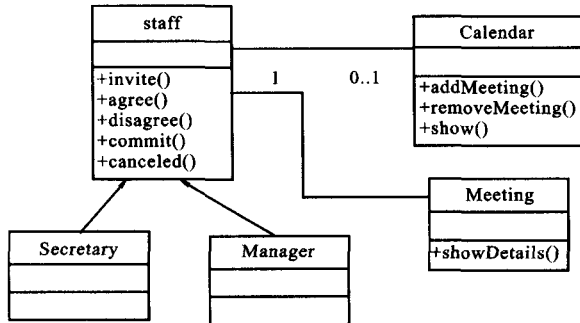


图4 系统的操作模型

此操作模型包含了五个对象类: Meeting, 只有一个操作 showDetails(), 这个操作用于描述会议时间以及与会人员等一些信息. Calendar 有三个操作, 其中 addMeeting() 和 removeMeeting() 分别表示添加会议以及删除会议, show() 操作描述了备忘录中所有的会议. Staff 对象类中 invite() 表示邀请职员参加会议. 操作 commit() 表示如果与会人员曾被邀请, 并同意参加会议, 那么就能成功安排此次会议. 如果与会人员不同意参加此次会议, 那么就可以使用操作 canceled() 取消这个会议. 操作 agree() 以及 disagree(), 前者表示同意参加会议, 后者表示不能参加被邀请的会议.

操作模型说明了执行用例所需的操作, 但是没有说明访问控制需求. 例如: 根据操作模型, 秘书可以通过调用 invite() 来召集会议, 但是这个操作只能由经理所访问, 这一点操作模型就没有考虑. 但是这种需求在安全模型中说明了. 这就导致了功能需求和访问控制需求的严格分离.

3.1.3 顺序模型

顺序模型描述的是参与者以及系统对象之间的交互. 文中描述了实现每个用例的时序图. 时序图中使用了操作模型类图中的操作.

图5是用例 summon a meeting 的两个时序图. 第一个时序图模拟经理调用 show() 操作查看备忘录后, 调用 invite() 操作邀请职员参加会议. 当所有被邀请的人员调用 agree() 同意参加会议后, 经理通过 addMeeting() 把会议添加到备忘录中, 并且调用 commit() 通知所有的与会人员有关会议的事项.

第二个时序图模拟某个被邀请参加会议的职员通过操作 disagree() 不同意参加此次会议, 经理就

会调用 canceled() 通知所有的与会人员取消此次会议.

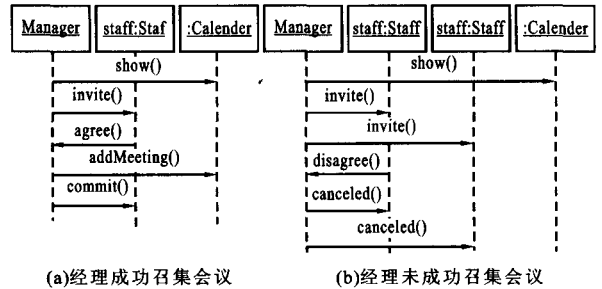


图5 用例 summon a meeting 的两个时序图

秘书也可以代替经理取消会议. 图6模拟了这种情况. 首先, 秘书通过 show() 阅读了经理的备忘录, 然后通过 removeMeeting() 选择应该被删除的会议. 最后秘书通过 canceled() 通知所有的与会人员取消会议. 类似, 经理也可以取消会议. 时序图和图6类似, 只需要把 Secretary 换成 Manager.

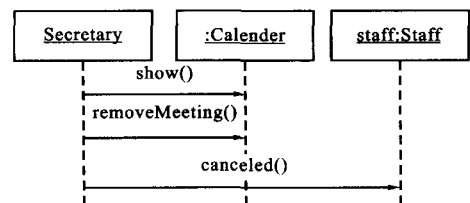


图6 秘书取消会议的时序图

图7描述了用例 look up calendar 的两种情况. 第一个时序图描述职员查看自己的备忘录(由对象约束语言说明). 第二个时序图描述秘书查看其经理的备忘录(由对象约束语言说明).

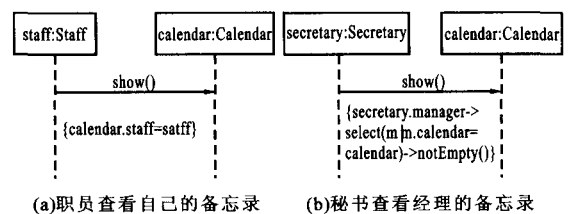


图7 look up calendar

3.2 安全分析

3.2.1 角色模型

UML 参与者是与用例通信的行为者. UML 角色由涉及通信关联的两个端点建模, 行为者与用例的关联就确定了一个角色^[4]. VBAC 中的角色是较粗粒度的任务的抽象, 表示的是一系列的功能. VBAC 的角色对应于 UML 中的参与者.

对于备忘录系统的角色模型, 有三个访问控制角色: Staff, Manager 以及 Secretary. 角色之间有继承关系, Manager、Secretary 分别继承了 Staff.

3.2.2 视图模型

时序图描述的是参与者执行用例时调用操作形成的内在的访问控制信息.例如:图5中时序图描述了经理想要召开一次会议必须调用操作 show(), addMeeting(), invite()以及 commit().被邀请的职员需要操作 agree()和 disagree()答复.

VBAC 的视图用于构造一系列的访问控制,这些访问控制是对某个对象的调用操作.文中为每个参与者以及每个时序图定义一个包含有被参与者调用的操作的视图,假设 A 是参与者, S 是时序图,视图 V(A,S)包含了被参与者 A 调用的每个操作.考虑图5中第一个时序图,命名为 Su,参与者 Manager. Manager 调用了操作 show(), addMeeting(), invite()以及 commit().因此,给出视图 V(Manager, Su),包含有访问控制权限 show, addMeeting, invite 以及 commit.视图 V(Staff, Su)包含访问权限 agree,没有参与者 Secretary 的视图.

视图规范了角色调用对象操作的权限,视图赋予角色的语义是角色具有视图规范中的访问权限.视图描述的是静态的访问权限.视图用与 UML 类图相似的图素来表示,它分成3部分.第1部分视图标识.第2部分说明视图限定的角色.根据角色之间的继承关系,这部分可以简化,某种意义上讲,按最少的角色原则列举该语句涉及的成员.最后一部分包含了这个视图的访问权限.视图之间可以相互继承.如果视图 V 继承视图 V',那么视图 V 包含了视图 V' 的访问权限.此外,视图还具有可传递性^[4].图8描述的是备忘录系统完整的视图模型.

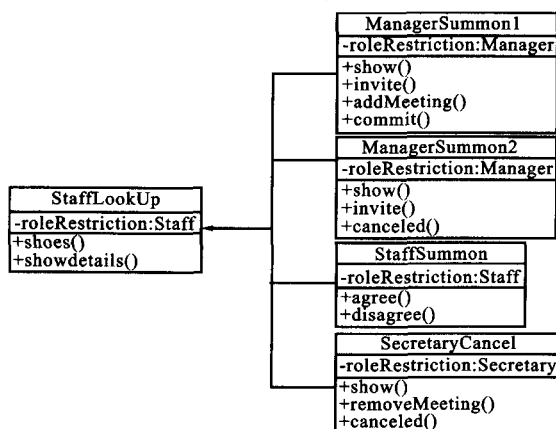


图8 系统的视图模型

3.3 VPL 访问控制策略规范

应用层的访问策略可以使用视图策略语言 VPL 描述. VPL 策略由关键字 policy 开始,它包含了每个角色的声明,角色声明语句的关键词 holds

后列举了安全策略的涉及视图,在关键词 on 之后说明视图针对的接口^[1-2].用 VPL 描述的 VPL 策略如图9所示.它规范了系统角色赋予的视图.

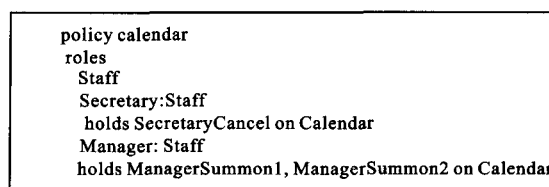


图9 VPL 策略

视图模型中的每个视图产生一个与其自身有相同标识以及相同访问权限的 VPL 视图定义.例如,图10描述的视图 StaffSummon.

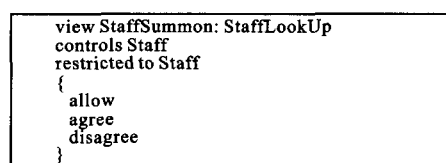


图10 视图 StaffSummon 的 VPL 描述

VPL 视图为类或者接口的部分操作定义访问控制.关键字 controls 引入接口,关键字 allow 引入授权的访问.视图之间的继承 VPL 中用“:”表示,“:”后面是父视图名称.视图模型描述 VPL 视图之间的扩展关系.如果视图 V 到视图 V' 之间有继承关系,V 生成的 VPL 视图就是对 V' 的 VPL 视图的扩展并且 V 的 VPL 视图中省略 controls 语句.

由 VPL 产生的访问控制策略规范,最终形成安全描述文件,可在目标环境中连同应用程序一起部署,类似于部署 EJB 组件.同时与系统安全管理系统相结合,保持系统安全的维护^[4].

4 结束语

文中通过模型驱动方法提出了如何将访问控制策略建模与功能需求的 UML 模型集成,展示了如何把功能模型作为安全模型的基础,最后形成访问控制策略.许多 UML 工具提供 UML 模式到 XMI 的导出. XMI 模式可以通过 XSLT 转换成 VPL-XML. VPL-XML 作为描述文件部署到 EJB、Web Service 或其他环境中.作为后续工作,模型驱动方法中约束的一致性检测值得深入研究.

参考文献:

- [1] Brose G. A view-based access control model for CORBA[C]// Security Internet Programming: Security Issues for Mobile and Distributed Objects. Berlin: LNCS 1603, Springer, 2002:237-337. (下转第142页)

式的特点进一步在编码的基础上定义诸如模式长度、距离、模式空间的基等概念,基于这些概念来制订多样化的模式空间划分策略,从而将原始数据集映射到满足不同编码特征的子模式空间中去,根据各个子模式空间的编码特征分别进行挖掘。

4.3 划分模式空间

初始数据集中的每个分片都部分地蕴含了结果模式集中的模式,所以无法分别在每个分片中直接进行目标模式的挖掘。通过对初始数据集进行处理,将其中蕴含的模式信息分别映射到互相独立的子模式空间中去以得到分别完整地蕴含各个子模式空间中的模式的分划数据集是划分模式空间的目的所在。分划数据集的每个分片对应一个子模式空间,位于集群的一个节点上。图2示意了初始数据集有2个分片,划分为3个子模式空间及生成相应的3个分划数据分片的情况下该过程中的模式信息的转移汇聚情况。

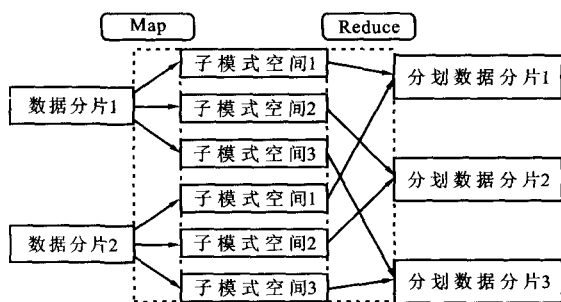


图2 模式信息转移汇聚过程图

划分的模式空间的个数应根据总的处理任务、集群中节点的个数、每个节点的处理能力等灵活地确定。而实现该点的关键就是提取出全局数据集的特征信息,在此信息指导下确定模式空间的基的个数亦即子模式空间的粒度。基的选取决定了子模式空间的粒度及对应的各个分划数据集上进行子模式挖掘的时间、空间复杂度。

在通过一轮 Map/Reduce 提取模式空间的特征信息后,用户程序通过分析算法对其进行分析,确定模式空间的编码方案、模式空间的基及编码后的数

据集到子模式空间的映射方式。用户可以通过主节点将这些分析出的信息及特征信息本身组播到所有从节点,实现 share-nothing 体系结构下全局信息的高效访问。在这些全局信息的指导下,各从节点并行地对数据分片进行编码、映射等操作。

5 子模式空间上的模式挖掘

模式空间划分完成后,子模式空间中的模式被分划数据集的某个数据分片唯一地蕴含,设计合理的算法在全局数据的特征信息的指导下根据各个子模式空间的特征分别在相应的数据分片上并行地进行模式的挖掘。

6 结束语

本文在深入分析 Map/Reduce 并行编程框架进行数据分析的特点的基础上,提出在对复杂类型模式挖掘的问题本身重新建模的基础上,通过实现模式空间划分来实现在各个子模式空间上并行地进行模式的挖掘的方法。与通过 Map/Reduce 化将传统算法迁移到云集群上相比,本文提出方案具有更广泛的适用性并可取得更高的并行化程度。

参考文献:

- [1] Ng Andrew Y, Bradski Gary, Chu ChengTao. MapReduce for machine learning on multicore [R]. NIPS, 2006.
- [2] Dan Gillick, Arlo Faria, John DeNero. Map/Reduce: distributed computing for machine learning[R]. 2006.
- [3] Dean Jeffrey, Ghemawat Sanjay. Map/Reduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1):107-113.
- [4] 陈康,郑伟民. 云计算:系统实例与研究现状[J]. 软件学报, 2009, 20(5):1337-1348.

作者简介:

刘 赛 男, (1985-), 博士研究生. 研究方向为分布式计算、云计算、软件工程。

(上接第 139 页)

- [2] 张赛男,殷兆麟. 软件系统 UML 建模与其安全建模的集成[J]. 计算机工程, 2007, 33(8):86-88.
- [3] Pende T. UML 宝典[M]. 北京:电子工业出版社, 2004:146-152.
- [4] Brose G, Koch M, Lohr K P. Integrating access control design into the software development process [C]//

Proc. of the 6th International Conference on Integrated Design and Process Technology. Trondheim, 2002.

作者简介:

张赛男 女, (1979-), 硕士, 讲师. 研究方向为软件工程、网络安全。