电子机构基于角色的访问控制模型①

Electronic Institutions Role-Based Access Control Model

李红霞　蔡国永　(桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘　要： 根据电子机构的基本概念、结构，提出了电子机构基于角色的访问控制模型，并与传统的基于角色的访问控制模型进行比较。采用重写逻辑 Maude 工具中面向对象的建模方法对该模型进行建模，并将模型应用到医院管理信息系统中。测试结果表明该模型可以作为一个共享的安全模块内核，对于具体的应用域电子机构只需做一些简单的定义和配置就可以运行。

关键词： 电子机构 重写逻辑 访问控制

1　引言

电子机构(EIs，electronic institutions)是人类组织代理的副本，为提供支持、信任和合法的商业应用而设计。其功能就象在人类社会中组织机构一样，以机构规范来创造信任，防止欺骗并通过验证规则来减少欺骗。对于开放、自治的电子机构，对其安全性的研究尤为重要。将基于角色的访问控制(RBAC，role based access model)模型嵌入到电子机构模型中，将是一个提高电子机构安全性的一个可行途径。

尽管对电子机构的开发已做了大量研究。如文献[1]介绍了为支持电子机构开发而设计的一些辅助软件：Islander 用于电子机构图形规约；AMELI 用于电子机构的基于 Agent 的中间件，构成电子机构的执行平台。一些其它研究提出采用义务逻辑、状态变迁以及进程代数等方式来对电子机构进行建模和分析等，如文献[2-4]。然而对电子机构的安全性方面研究涉及还甚少。本文在提出电子机构基于角色的访问控制模型(EIs-RBAC)的基础上，用重写逻辑 Maude 工具来实现该模型，验证了该模型在电子机构安全性开发上的可用性：即该模型可以作为一个共享的安全模块内核，对具体的应用域电子机构只需做一些简单的定义和配置就可以运行。

2　电子机构基于角色的访问控制模型

基于角色的访问控制其基本思想[5]是:将角色赋给用户，权限不直接赋予用户而是赋予角色。用户通过担任某些角色而获得权限。这能极大地简化权限管理，减少管理访问控制策略的开销，并且易于描述和理解。电子机构基于角色的访问控制模型如图 1 所示：

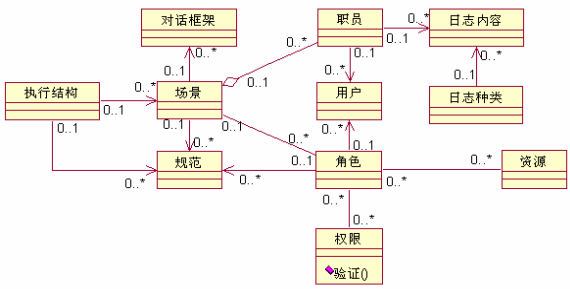

图 1　电子机构基于角色的访问控制模型(EIs-RBAC)

说明：图 1 中，菱形是聚合(Aggregations)，是强连接，是整体与个体的关系。

在图 1 中，需要明确下列概念：

1) 执行结构：指定场景之间的关系，定义哪些场景可以由每个不同的角色到达。在执行结构之内，代理可以在同一时间由不同的角色参与到不同的场景中。

2) 场景：两个代理间的交互是通过代理组汇合点明确表达，这称为场景。在场景的一些特殊状态和根据这个场景中的角色，代理可以进入场景或离开场景。

3) 对话框架：定义代理可能交换的有效表达行

① 收稿时间:2008-12-25

为，并且定义参与者角色和角色关系。通过共享一个对话框架，使异种代理与其他代理交换知识。

4) 规范：定义什么是允许的，什么是禁止的以及什么是责任，包含如何描述不同的角色可以合法地在场景间移动。

5) 职员：是系统的使用者，是一个访问计算机系统中的数据或者用数据表示的其它资源的主体，是对系统功能及数据对象进行操作的主体。

6) 用户：这里的用户仅仅维护用户的标识并用于绑带角色，是虚拟的用户，并不是指一个实际的人或被拓展定义的机器、网络以及聪明的代理。关于用户其它重要的信息在职员中体现。

7) 角色：是指一个组织或任务中的工作或位置，代表了一种资格、权利和责任。如：医生、护士、传达员。

8) 权限：权限表用户对系统中的功能进行某种操作的权力。如：对数据库表的增加、删除等。

9) 资源：包括所有受访问控制的资源，在不同的应用背景下可以有不同的定义。如：在数据库中可以是一个表中的记录；在我们具体的应用系统中，可以是其中的一个功能。

10) 日志：是应用软件本身的日志，记录用户进入系统后所做的各种自己权限范围内的操作。

2.1 传统 RBAC 与 EIs-RBAC 模型的比较

传统 RBAC 的核心思想几乎是所有访问控制的研究员都公认的一种思想，即用户与角色、角色与权限、角色与资源对象之间的关系均是多对多的关系，而在 EIs-RBAC 模型中，角色与用户之间是一对多的关系，并增加了职员，且职员与用户之间也是一对多的关系。这样设计的优点是：1) 可以简化编码时的复杂判断；2) 使登录用户与对系统功能及数据对象进行操作的主体的基本信息分离，不会使数据库表用户的内容很庞大。

其次，在 EIs-RBAC 模型中,权限与资源之间没有直接的关系,图 1 中的权限只针对角色,而不针对资源。其优点是：1) 由于资源较多而角色相对较少，使角色关系具有相对稳定性和易维护性；2) 在大型系统中，可以减少角色、权限和资源的维护量；3) 降低访问控制算法实现的复杂度；4) 简化了角色、权限和资源之间的关系。缺点是，在一些机密系统中，权限设置过粗，不能对每个角色访问每个资源的权限进行控制。

最后，在传统的 RBAC 模型中没有日志管理，而 EIs-RBAC 模型中引入了日志。日志管理对进入系统的每个用户所做的操作都要进行记录。其优点是：1) 可以反馈当时的系统情况，为工程商查找故障时提供线索；2) 是落实某些操作员违反公司规定操作的责任和证据，从而也进一步保证了系统的安全操作。

3 Maude工具介绍

3.1 Maude 工具简介

Maude[6]工具由 UIUC 大学开发，同时支持等式和重写规则的强大工具,通用性比较强,表达能力丰富,而且具有良好的性能。具有简单性、易用性和高效性这三个方面的优点。

在重写逻辑 Maude 工具中，重写理论[6]被定义为一个四元组 $\Re = (\Sigma, E \cup A, \Phi, R)$，其中，$(\Sigma, E \cup A)$ 是模块的等式理论部分；Φ 是一个函数，将 Σ 中的每个操作符映射到相应的冻结参数；R 是(条件)重写规则。在 Maude 工具中，重写规则使用下面的记法：rl [标签名]：<t> =><t'>.条件重写规则采用 crl [标签名]：<t> =><t'> if condition. 的方式表示。它们的重写关系用“=>”表示。等式理论可被视为形如$(\Sigma, E \cup A, \Phi_\phi, \phi)$的重写理论的一种简化，其中 $\Phi_\phi(f) = \phi$，即对基调 Σ 中的每个操作符 f，没有 f 的参数被冻结。

在 Maude 中，有三种模块类型。

1) 面向对象模块(object-oriented Module)，简称 omod。

Maude 中的面向对象模块用于定义并行的面向对象系统，可用于描述面向对象的状态迁移。其主要的面向对象特征是分类定义、子类说明、消息说明以及重写规则。在 Maude 中描述面向对象系统的三个基本类型是对象(Object)、消息(Msg)和配置(Configuration)。其关键字说明如下:

omod <模块名> is <说明和描述> endom

2) 功能模块(functional Module)，简称 fmod。

用于定义各种数据类型和操作，这些定义都是通过等式理论来完成。功能模块用如下关键字声明:

fmod <模块名> is <声明和描述> endfm

模块名必须是一个标识符。声明部分包括其他函数模块的入口声明,类别(sort)、子类别(subsort)和操

作声明(operator declarations)等。描述部分包括等式关系和成员关系的原子式。

3) 系统模块(system Module)，简称 mod。

用于说明重写规则。重写规则中包括类别(sort)、种类(kind)和操作，有等式、成员关系和规则三种表述，这三种表述即可以是有条件的又可以是无条件的。因此，任何重写理论都隐含着等式理论，其实就是等式、成员关系加上规则。系统模块用如下关键字声明:

mod <模块名> is <声明和描述> endm

系统模块与函数模块类似，系统模块中的模块名也必须是一个标识符。系统模块中的一声明和描述部分可以是以下几种声明的一个或多个的组合:

a) 模块入口声明;

b) 类别和子类别声明;

c) 操作声明;

d) 变量声明;

e) 等式和成员关系声明;

f) 规则说明。

3.2 模型在 Maude 工具中的定义

电子机构基于角色的访问控制模型在重写逻辑 Maude 工具中的定义，主要采用 Maude 工具中面向对象的方法和过程[7]来完成。

首先,可以把 EIs-RBAC 模型看成一个大类--电子机构类(Electronic-institutions)，该类主要用于记录电子机构中的一些基本信息。为了从 Electronic-institutions 中根据不同类名获取不同的值，需要定义执行结构(Performative-structure)、用户(User)和角色(Role)这三个类。

其次，需要定义获取资源、权限、角色等，同时写入日志的规则 user-get-framework。

另外，需在登录成功后，才能获取资源、权限等，故需建立登录规则 user-login。

最后，为了测试建立的模型是否正确，首先建立一个配置 EISConf。

到此，EIs-RBAC 模型在 Maude 工具中已经定义完毕。

4 模型具体应用

为了具体说明电子机构基于角色的访问控制模型在 Maude 工具中的实现，设计一个简单的医院信息管理系统，主要体现，在该系统中基于角色的访问控制是如何设计的。医院信息管理系统包含很多的科室;医院或科室对医护人员某些行为的限制，即操作规范;医护人员之间通过发送消息来交流(如：邮件);每个科室都包括医生、护士这两个基本的角色；只有医生才能给病人开处方、写病历等，护士只能查看病历；该系统需要有日志功能，日志分登录日志和操作日志;用户登录系统成功后，才获取该用户可以访问的资源、该用户扮演的角色以及其他相应的信息。根据医院信息管理系统的业务需求，我们可以用 ER-Studio 建立医院管理信息系统的部分逻辑结构图，如图 2 所示。在图 2 中建立了医院基本信息、医院科室基本信息、医院人员基本信息、医院人员操作规范、用户信息(登录系统)、角色、资源(病历)、权限、消息内容、日志种类和日志内容等实体。图 2 中建立的实体与图 1 模型的对应关系如图 3 所示：

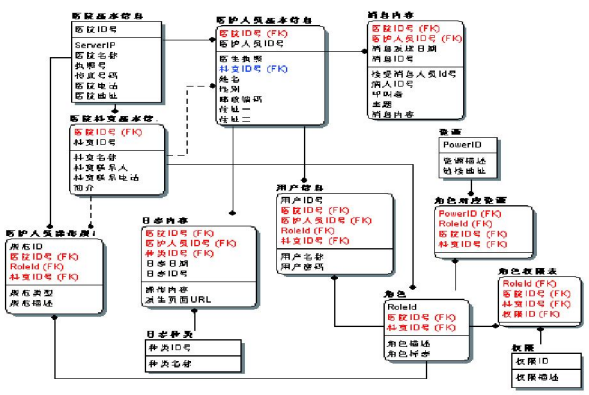

图 2　医院信息系统的部分逻辑结构图

图 3　医院信息系统与 EIs-RBAC 模型实体的对应图

在重写逻辑 Maude 工具中，我们将医院基本信息(Hospital)定义为执行结构的子类，如： subclass

Hospital<Performative-structure .将医院信息管理系统中的用户 Doctor1、Doctor2、Nurse1 定义为用户的子类，如：subclass Doctor1 Doctor2 Nurse1<User . 将医院信息管理系统中的医生、护士定义为角色的子类，如：subclass Doctor Nurse< Role.除此之外，医院信息管理系统中的其它信息直接在 EISConf 中配置。最后运用 down 命令运行，结果如图 4 所示：

从图 4 我们可以看到：用户 U1 是医院 H 中的一个职员 666，在 1111 部门工作，他的权限是 2(可读写病历)，扮演的角色是 5(医生)，他需要遵循的规范是 777，他与其他人的交流信息是 115，并且写入了登录日志<U1,1>。

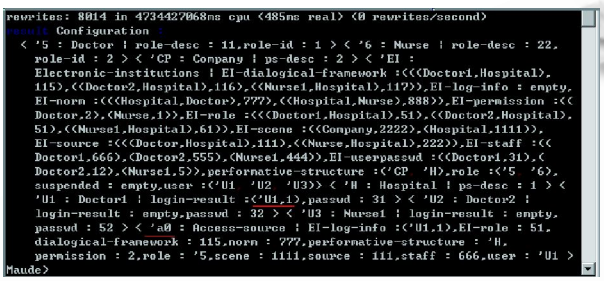

图 4　医院信息管理系统运行结果

根据上面这个例子，并结合我们的现实社会，执行结构可理解为一个单位或公司等；场景可理解为一个单位或公司包含的部门；规范可理解为对员工的行为标准或操作流程；对话框架可理解为部门之间、员工之间的信息交流；角色可理解为在单位或公司所担任的职务，如总经理、部门经理等；资源可理解为单位或公司为了日常的运转，所需要的软硬件。权限可理解为在单位或公司，担任什么职务级别的人才能使用某些硬件或审批某些项目，或者在软件系统中，对某些功能或数据的操作权力。职员可简单理解为单位或公司的员工。

5　结论

本文对 EIs-RBAC 模型进行了研究。根据上面的运行结果，可以了解到，在重写逻辑 Maude 工具中，实现了 EIs-RBAC 模型的模拟。虽然医院信息管理系统是一个简单的例子，但可以得出下面的结论：1)EIs-RBAC 这个模型在 Maude 中是可以执行的模型；2)针对具体的机构，我们可以根据具体的需求将一些类定义为 EIs-RBAC 模型中相应类的子类；3)据具体的需求可以将机构中的一些信息直接在 EISConf 中配置；4)可以把电子机构基于角色的访问控制模型作为一个共享的安全模块内核，对于具体的应用域电子机构只需做一些简单的定义和配置就可以运行。

参考文献

1 Sierra C, Antonio J, guez-Aguilar R í , Noriega P, et al. Engineering multi-agent systems as electronic institutions. European Journal for the Informatics Professional, 2004,170:33－39.

2 Dignum V. A model for organizational interaction. [Ph.D. Thesis]. Dutch Research School for Information and Knowledge Systems, 2004.

3 Sibertin-Blanc C, Amblard F, Mailliard M. A Coordination Framework Based on the Sociology of Organized Action. Computer Science, 2006,3913:3－17.

4 蔡国永,高济,董荣胜.电子机构的进程代数模型研究.微电子学与计算机, 2007,24(10):74－77.

5 杜萍,刘弘.改进的协同系统中基于角色的访问控制模型.计算机工程与应用, 2006:8－10.

6 http://maude.cs.uiuc.edu/

7 李红霞,蔡国永.电子机构的安全性分析研究.计算机系统应用, 2008,17(8):46－50.