

一种基于矩阵组织的动态授权模型^①

潘松寒 (北京燕山石化电信事业部 北京 102500)

摘要: 动态授权技术是授权技术领域的一项重要技术。DAFMAT 框架是该领域中经典的授权框架,然而应用范围受医疗系统组织结构特征所限制。笔者通过改造 DAFMAT 模型,安排正式的横向角色,加强矩阵组织的横向机制,提出一种名为 DAFMO(矩阵组织的动态授权框架)的模型。DAFMO 符合矩阵组织的普遍特征,其应用范围将得到极大的拓展。

关键词: 多授权类型的动态授权框架 矩阵组织的动态授权框架 任务角色驱动 矩阵组织

A Dynamic Authentication Based on Matrix Organization

PAN Song-Han

(Telecom Division, Beijing Yanshan Petrol-Chemical LTD, Beijing 102500, China)

Abstract: Dynamic authentication is an important technique in the field of authentication. DAFMAT is a classical authentication framework in this field, however to which application is confined by healthcare organization structure characters. This paper strengthens lateral mechanism by rebuilding DAFMAT model and arranging formal liaison role, and presents a model named DAFMO(dynamic authentication framework for matrix organization). DAFMO satisfies common characters in matrix organization, whose application is extremely extended.

Keywords: DAFMAT; DAFMO; TRD; matrix organization

1 引言

对于动态授权模型而言,目前学界提出的最完整的理论框架,是一种名为 DAFMAT(Dynamic Authorization Framework for Multiple Authorization Types)模型,DAFAMT 是一种支持多授权类型的授权框架,这种模型主要是针对医疗应用系统。它把 RBAC(Role-based Access Control)和 DTE(Domain and Type Enforcement)结合起来,并使用一个逻辑驱动认证引擎。DAFMAT 包括两个部分:混合访问控制模型和逻辑驱动授权引擎。混合访问控制模型是 RBAC 和 DTE 结合,而授权引擎则基于一阶谓词逻辑。DAFMAT 能够动态根据用户、角色、主体、上下文变量以及优先级等信息把访问请求分为 3 种:一般、紧急和基于上下文的。然后根据不同的授权请求类型应用不同的验证规则,并给通过验证的用户会话分配正确的域^[1]。

本文在 DAFMAT 基础上通过安排正式的集成角色,着重加强系统内部的横向联结。具体的模型的实现上是在 RBAC 模块和 DTE 模块的基础上嫁接任务模块,下文称为 TRD(Task Role Driver)模块,参见图 1。通过系统默认的授权流程,或者是经过任务角色和职能角色沟通后形成的一致意见,解决职能权力间的显性冲突。笔者将修改过的授权模型命名为 DAFMO(Dynamic Authorization Framework for Matrix Organization)。

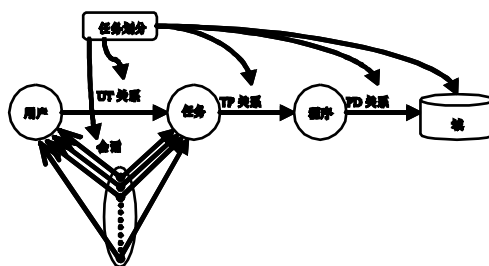


图 1 TRD 模块图

^① 收稿时间:2009-04-07

Role(t,f)→FuncRole_Domain(f,d);

TRD 模块是通过给用户分配任务角色,但是必须满足一个约束,有行政级别的职能角色不能再被赋予任务角色,任务角色可以依据需要与否缺省,由程序控制代替人为指派。

$\forall (u,t,f) \{User1_FuncRole(u,f) \& Taskrole_FuncRole(t,f) \rightarrow User2_TaskRole(u,t)\}$

2.4 动态授权规则相关要素

2.4.1 格式化授权请求谓词

DAEMAT 是一个支持应用的授权框架。因此它必须有将应用中用户活动转化为合适的授权请求。

```
Current_Action(menu_option,session)
Session_User1(session,user1)
Session_FuncRole(session,funcrole)
Session_User2(session,user2)
Session_TaskRole(session,taskrole)
Menu_Operation(menu_option,process)
Menu_Context(menu_option,cv)
Session_Priority(session,pc)
(pc=NR 代表一般请求, pc=ER 代表紧急请求)
Auth_Req(user1,funcrole,process,cv,cv_value,pc)
```

2.4.2 指派授权类型

将授权类型指定给授权请求,授权类型分为三类:

- ① 一般授权请求
- ② 紧急授权请求
- ③ 基于上下文的授权请求

授权类型指派逻辑

```
IF pc='ER'
THEN Emergency_Auth_Req(user1,funcrole,
process)
ELSE
IF cv='NONE'
THEN
Normal_Auth_Req(user1,funcrole,process)
ELSE
Context_Auth_Req(user1,funcrole,user2,taskrole,
process,cv,cv_value)
```

2.4.3 确认不同授权类型的规则

授权类型分为三类:一般授权、紧急授权、基于上下文授权。

- ① 确认一般授权请求的条件

Process_FuncRole(process,funcrole)→

Normal_Auth_Req(user1,funcrole,process)

- ② 确认紧急授权请求的条件

ER_FuncRole_Map(funcrole,mapped_funcrole)&

Process_funcRole(Process,mapped_funcrole)→

Emer_Auth_Req(user1,funcrole,process)

- ③ 确认基于上下文的授权请求的条件

IF cv='CTXT_VAR1'

THEN

```
Process_FuncRole_TaskRole(process,funcrole,taskrole)&Context_Predicate_relevant_for_CTXT_VAR>→
Context_Auth_Req(user1,funcrole,user2,taskrole,process,cv,cv_value)
```

授权过程的最后一步将对应的域指定给用户会话。在静态授权框架中,授权要求关联程序直接从域——类型访问矩阵中的实体获得。在系统确定与程序关联的域以及与资源关联的类型之后,资源才被赋予给用户。在动态授权模型中,决定程序联系的不仅有静态关系,还有临时性动态关系。将这种临时性关系纳入角色关系的评估中,也就意味着授权过程是一个动态过程。

3 DAFMO模型的应用实例

本文依据 CRM 的主干业务,草拟出 SSS(Client Service,Sales,Store)系统(见图 3),可应用于家用电器零售。简化不必要的程序,同时给与各种角色必要的最小授权。当权力间发生冲突时,授权的分类,有助于市场策略的应用。SSS 模型涉及三大职能部门:客户服务部(Client Service Department)、销售部(Sales Department)、仓储部(Store Department),以及六种职能角色:客服部经理(CS manager)、客服人员(Client Server)、销售部经理(Sales Manager)、业务员(Sales Operator)、仓储部管理员(Store Custodian)、送货人(Deliverer),以及两种任务角色:退货经理(Return Manager)和债务经理同时依据任务主线划分 7 个业务流程:订单确认流程(Order Admissions Process)、退货流程(Return Process)、订单执行流程(Order Execution Process)、债务催缴流程(Debt Press Process)、反馈流程(Feedback Process)、单据处理流程(Bill Process)、仓储流程(Store Process)^[3]。

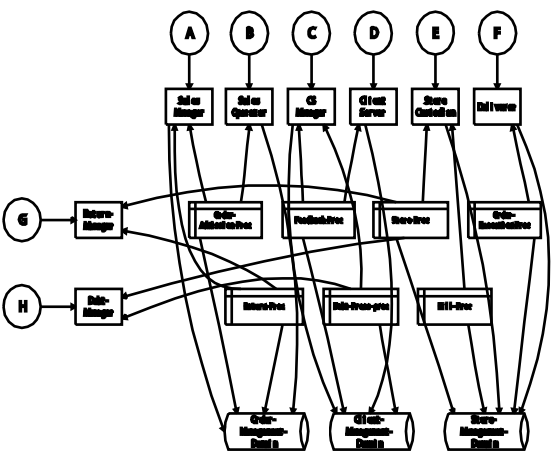


图 3 SSS 系统的 DAFMO 混合访问控制模型
(有关说明见附录)

色)，所以授权方法遵循着 DAFMAT 模型的设计。对于 DAFMAT 而言，虽然把请求分为一般、紧急、基于上下文，出于一种结构性的权变主义思路，资源的最终分配决定必须由人做出。这个人有权决定默认系统结果，还是由自己亲自分配资源^[3]。

本文假定一个用户 Gear 接到一个投诉要求退货还钱,具体操作由销售部经理 Alice 和仓库保管员 Edison 执行:

4.1 User1=Alice

funcrole=sales_manager
User2=Gear
taskrole=return_manager
process=return_proc
cv=returntype
pc=NR

cv_value='moneyback'

最后形成的授权函数表达式为:

Auth_Req(Alice,sales_manager,return_proc,returntype,'moneyback',NR)

依据第 3.1.1 节提到的逻辑运算，该请求最终变为一个基于上下文的授权请求:

Context_Auth_Req(Alice,sales_manager,Gear,return_manager,return_proc,returntype,'moneyback')

Process_FuncRole_TaskRole(return_proc,sales_manager,return_manager)&

Equals(cv,return)&Return_Assignment(user1,user2,cv_value)

4.2 User1=Edison

funcrole=sales_manager
User2=Gear
taskrole=return_manager
process=return_proc
cv=returntype
pc=NR

cv_value='moneyback'

最后形成的授权函数表达式为:

Auth_Req(Edison,store_custodian,return_proc,returntype,'moneyback',NR)

依据第 3.1.1 节提到的逻辑运算，该请求最终变为一个基于上下文的授权请求:

Context_Auth_Req(Edison,store_custodian,Gear,return_manager,return_proc,return,'moneyback')

3.1 逻辑驱动的授权引擎

以下代码用于将相关上下文谓词转化为一阶谓词菜单选项:

Process_FuncRole_TaskRole(process,funcrole,taskrole)&Equals(cv,return)&Return_Assignment(user1,user2,cv_value)→

Context_Auth_Req(user1,funcrole,user2,taskrole,process,cv,cv_value) (1)

Process_FuncRole_TaskRole(process,funcrole,taskrole)&Equals(cv,)&Debt_Assignment(user1,user2,cv_value)→

Context_Auth_Req(user1,funcrole,user2,taskrole,process,cv,cv_value) (2)

Process_FuncRole_TaskRole(process,funcrole,None)&Equals(cv,Installtion)&Return_Assignment(user1,user2,cv_value)→

Context_Auth_Req(user1,funcrole,process,cv,cv_value) (3)

Process_FuncRole_TaskRole(process,funcrole,None)&Equals(cv,Feedback)&Return_Assignment(user1,user2,cv_value)→

Context_Auth_Req(user1,funcrole,process,cv,cv_value) (4)

4 SSS模型验证

对于一般授权和紧急授权问题，系统职能就足够发挥作用，不需要参考标识横向联结的向量值(任务角

基于上下文谓词转化为一阶谓词选项：

```
Process_FuncRole_TaskRole(return_proc,store_custodian,return_manager)&
Equals(cv,return)&Return_Assignment(user1,,user2,cv_value)
```

5 总结

同 DAFMAT 相比，DAFMO 安排的正式的集成角色，专职特殊的任务。任务角色和任务团队的规模是由业务需要所决定的。如文中给出的 CRM 系统中的 SSS 模型，横向角色只有负责退货的经理和负责债务的经理。这两个角色关系到多个职能部门，还关系到企业的客户策略和市场策略，所以给予这两个角色很大的权变空间。模型的应用范围决定于组织结构是否匹配，不在乎技术和专业是否相同。当然，系统的弱点是就是组织结构的弱点，是需要人为的加以克服的。

本文的意义决不是对动态授权模型的又一次画蛇添足，而是想通过一个实例表明 DAFMAT 模型及其修正模型 DAFMO 广阔的应用空间，同时简单阐明它同矩阵化组织间的关系。整体上看，由 DAFMO 模型设计出来的信息系统聚合度高，耦合度低，系统效率得到极大的提升。

附录：

表 1 用户——职能角色映射

用户	角色	谓词逻辑
A	sales_manager	User_FuncRole(A,sales_manager)
B	sales_operator	User_FuncRole(B,sales_operator)
C	CS_manager	User_FuncRole(C,CS_manager)
D	client_server	User_FuncRole(D,client_server)
E	store_custodian	User_FuncRole(E,store_custodian)
F	deliverer	User_FuncRole(F,deliverer)

表 2 用户——任务角色映射

用户	任务角色	谓词逻辑
G	return_manager	User_TaskRole(G,return_manager)
H	debt_manager	User_TaskRole(H,debt_manager)

表 3 职能角色——域映射

职能角色	域	谓词逻辑
sales_manager	Order_mgmt_domain	Role_Domain(sales_manager,Order_mgmt_domain)
sales_operator	Order_mgmt_domain	Role_Domain(sales_operator,Order_mgmt_domain)
CS_manager	CS_mgmt_domain	Role_Domain(CS_manager,CS_mgmt_domain)
client_server	CS_mgmt_domain	Role_Domain(client_server,CS_mgmt_domain)
store_custodian	Store_mgmt_domain	Role_Domain(store_custodian,Store_mgmt_domain)
deliverer	Store_mgmt_domain	Role_Domain(d deliverer,Store_mgmt_domain)

表 4 程序——职能角色映射

程序	角色	谓词逻辑
Order-admission-proc	Sales_operator	Process_FuncRole(admission_proc,sales_operator)
	Sales_manager	Process_FuncRole(admission_proc,sales_manager)
Return-proc	Sales_manager	Process_FuncRole(Return-proc,Sales_manager)
Order-executions-proc	deliverer	Process_FuncRole(Order-executions-proc,deliverer)
Debt-press-proc	CS_manager	Process_FuncRole(Debt-press-proc,CS_manager)
Feedback-proc	Client_server	Process_FuncRole(Feedback-proc,Client_server)
	CS_manager	Process_FuncRole(Feedback-proc,CS_manager)
Bill-proc	Store_custodian	Process_FuncRole(Bill-proc,Store_custodian)

Store-proc	Store_custodian	Process_FuncRole (Store-proc, Store_custodian)
------------	-----------------	--

表 5 程序——任务角色映射

程序	任务角色	谓词逻辑
Return-proc	Return_manager	Process_TaskRole (return_proc,return_manager)
Debt-press-proc	Debt_manager	Process_TaskRole (debt_proc,debt_manager)
Store-proc	Return_manager	Process_TaskRole (store_proc,return_manager)
	Debt_manager	Process_TaskRole (debt_proc,debt_manager)

表 6 程序——域映射

程序	域	谓词逻辑
Order-admission-proc	Order_mgmt_domain	Process_Domain (Order-admission-proc, Order_mgmt_domain)
Return-proc	Order_mgmt_domain	Process_Domain (Return-proc, Order_mgmt_domain)
Order-executions-proc	Store_mgmt_domain	Process_Domain (Order-execution-proc, Store_mgmt_domain)
Debt-press-proc	CS_mgmt_domain	Process_Domain (Debt-press-proc, CS_mgmt_domain)
Feedback-proc	CS_mgmt_domain	Process_Domain (Feedback-proc, CS_mgmt_domain)
Bill-proc	Store_mgmt_domain	Process_Domain (Bill-proc, Store_mgmt_domain)
Store-proc	Store_mgmt_domain	Process_Domain (Store-proc, Store_mgmt_domain)

表 7 域——类型访问矩阵

域	对象类型/访问模式		
	Information Flow	Material Flow	Capital Flow
Order_mgmt_domain	C,U,D,V	V	C,U,D,V
CS_mgmt_domain	U,D,V	V	V
Store_mgmt_domain	V	C,U,D,V	U,V

访问模式：C-新建，U-更新，D-删除，V-浏览
职能角色依据情况赋予权限，而任务角色被赋予全部权限。

表 8 基于上下文的菜单选项

程序	上下文变量
Order-admission-proc	NONE
Return-proc	Returntype(moneyback, exchange,compensation)
Order-executions-proc	Installtion(plasma TV, Water Heater, Air conditioner)
Debt-press-proc	Paymenttype(Installment,pay on delivery)
Feedback-proc	Feedbackmeasure (questionnaire,phone)
Bill-proc	Billtype(inward, outward,transfer,spoil)
Store-proc	NONE

参考文献

1 Ramaswamy Chandramouli Computer Security Division, ITL, NIST, Gaithersburg, MD 20899 (mouli@nist.gov),(DB)A Framework for Multiple Authoriza- tion Types in a Healthcare Application System, 2000.1 – 7.

2 Crêteur, Yves Pochet Muriel,Organizational design and hospital performance.[1999-12-20]www.uclouvain.be/cps/ucl/doc/iag/documents

3 梁瑾,王咸伟,陈虹.客户关系管理系统的设计与实现. 电脑与信息技术, 2008,16(2).