开放分布系统中基于 RBAC 和策略实施的协调模型

张亚英，尤晋元
（上海交通大学分布计算技术中心，上海 200030）

摘 要： 本文提出了适用于开放分布系统的基于 RBAC 和策略实施的协调模型（RBPEC）. RBPEC 模型引入了协调组的概念，把需要进行交互才能完成整体任务的多个软件实体放在一个协调组中. RBPEC 协调模型把协调和访问控制两者结合起来，引入基于角色的访问控制为协调系统作安全性检查，系统的协调策略可以在分布环境下实施. RBPEC 协调模型具有安全和可扩展两大特点.

关键词： 协调；基于角色的访问控制 RBAC；策略；开放分布系统
中图分类号： TP311.5 **文献标识码：** A **文章编号：** 0372-2112 (2002) 12A-2000-04

A RBAC Based and Policy Enforcement Coordination Model in Open Distributed System

ZHANG Ya-ying, YOU Jin-yuan
(*Distributed Computing Technology Center, Shanghai Jiaotong University, Shanghai* 200030, *China*)

Abstract: An RBAC based and policy enforcement coordination model suitable in open distributed system is described. A set of software entities interacting with each other for a common global system task constitutes a coordination group. In this model coordination is combined with access control. Role based access control is introduced for security concerns and coordination policy is enforced in a distributed manner. The model has two characteristics of being secure and scalable.

Key words: coordination; RBAC; policy; open distributed system

1 引言

随着因特网和电子商务的发展[1]，不同领域的软件实体往往需要相互合作才能完成任务，这种合作必然涉及软件实体间的交互. 协调模型提供了描述软件实体间交互的框架[2,3]. 本文提出了一种基于 RBAC 和策略实施的协调模型 RBPEC（RBAC Based and Policy Enforcement Coordination），适用于开放分布系统（如 Internet）中软件实体之间的交互. 这里我们说的软件实体（简称实体）指的是软件组件，它往往与其他软件组件或周围环境进行交互，合作完成整体任务.

目前的协调技术主要考虑如何促使实体间的交互能顺利通畅地进行. 但事实上，基于系统安全性的考虑，往往需要有一定的访问控制措施来限制软件实体间的某些交互. 表面上看这两者是对立的，但实际上两者是紧密相连的，只有把两者结合起来考虑，才能充分表述开放分布系统中实体间的交互.

RBPEC 模型主要包括以下几部分：(1) 基于 RBAC[4] 机制，防止软件实体进行未经授权的访问；(2) 基于元组空间[5] 和协调代理的协调模型，软件实体间以及实体与元组空间之间的所有通信都通过它们对应的协调代理完成；(3) 协调策略可以分布式地实施.

2 RBPEC 模型总体框架

RBPEC 模型是针对一个协调组而言的. 协调组是为了完成共同的系统任务而需要进行交互的一系列软件实体的集合. 我们假设软件实体加入某个协调组，是因为它需要与这个协调组中的某些成员进行信息的交互.

RBPEC 模型的总体框架如图 1 所示. RBPEC 模型可以描述为一个五元组（ENTITY, GM, PROXY, CN, TS）. 其中 ENTITY 是协调组中软件实体的集合；GM 是协调组的管理员；PROXY 是协调组中协调代理的集合；CN 指通信网络；TS 是协调组中元组空间的集合.

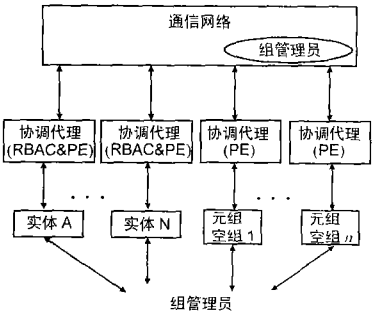

图 1 RBPEC 模型的整体框架.（整个协调组中只有一个组管理员，图中底部的组管理员实际上只是为了图形的清晰简洁附加上去的）

软件实体就是在整个协调过程中被协调的对象. 元组空间是用于存储实体间交换信息的公共数据空间.

每个协调组都有一个组管理员负责完成协调组的管理工作，组管理员的工作包括：(1) 软件实体和元组空间在组内的注册、角色分配和协调代理的分配；(2) 根据协调系统的实际情况制订角色分配和协调策略方案，并把方案广播到组内的每个协调代理.

收稿日期：2002-06-10；修回日期：2002-09-28
基金项目：国家自然科学基金（No.69973032）

每个软件实体和元组空间都有一个协调代理相对应,从逻辑上看,它位于软件实体/元组空间和通信网络之间(参见图1).事实上协调代理可以分布在网络的任意位置,而且多个软件实体/元组空间可共享同一个协调代理.软件实体的协调代理和元组空间的协调代理在功能上略有不同,软件实体的协调代理负责基于角色的访问控制检查和协调策略的实施,而元组空间的协调代理只负责协调策略的实施.元组空间是公共的信息存储空间,我们假设软件实体可以自由地访问元组空间,所以对元组空间并不需要进行访问控制.因此相对于两个软件实体之间的交互而言,软件实体与元组空间的交互(即软件实体访问元组空间)的过程要简单得多,所以本文下面的论述以两个软件实体间的交互为主.

3 RBPEC 的协调原理

图2是一个简单的协调组,包括实体 A 和 B、元组空间 ts、对应的协调代理、组管理员及所在的通信网络.图中省略了 A、B、元组空间与组管理员之间的连线.

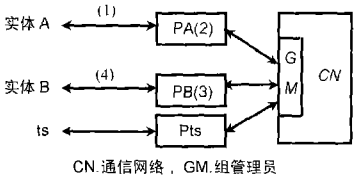

图2　一个简单的协调组

我们以实体 A 和 B 之间的交互为例来看 RBPEC 的协调过程,首先实体 A、B 和元组空间 ts 通过组管理员 GM 注册,组管理员分别分配相应的协调代理 P_A 和 P_B 和 Pts,赋予实体 A 和 B 相应的角色 roleA 和 roleB,并且把这些基本信息以表1的形式存储以便日后维护,元组空间可以被软件实体自由地访问,它不参与访问控制检查,它的角色设为 NULL.随后组管理员把整个协调组的角色分配信息和协调策略广播给协调代理,让协调组中的每个协调代理都有一份协调系统角色分配方案、角色的访问权限列表和协调策略的拷贝,这样协调策略和角色分配在组内的协调代理中是全局可见的.

表1　协调组管理员记录的维护信息

软件实体/元组空间名	协调代理名	赋予的角色
entity A	P_A	role A
entity B	P_B	role B
ts	Pts	NULL

接下来,软件实体 A 与 B 是怎样进行交互呢?这个过程分为4步(见图2中标注):(1)实体 A 向它对应的协调代理 PA 发出请求事件 *send*(A, *m*, B),其中 m 是 A 要发送的信息,B 是目标实体;(2)PA 根据组管理员提供的角色分配方案和角色的访问权限决定实体 A 是否有权访问 B;(3)经过访问控制检查后,如 A 获准访问 B,则接下来 P_A 将根据协调策略判断该信息能否继续前行,若可以则将此请求发送到目标实体 B 的协调代理 PB,这时 PB 将根据协调策略的要求将请求信息送至 B;(4)经过协调策略检查成功的信息最终抵达 B.

注意在目标实体 B 这边不需要进行角色的访问控制,因为协调组管理员在制订角色的访问权限时已经考虑到了这一点,在目标实体端无需访问控制的重复检查.

图2中所有的软件实体和元组空间如果共用一个协调代理的话,就是一个典型的集中式协调模型(CCM).只有在软件实体和元组空间的数目不多的情况下,这种方法是可行的.在 CCM 中,有一个专门的协调器处理整个协调系统中所有的协调活动,CCM 最大的缺点是这个协调器容易成为协调系统的瓶颈,这种结构也就不可能用于象 Internet 这样的开放分布环境中.为了克服这种缺陷,RBPEC 模型引入了可以分布在网络任意位置的多个协调代理,由这些协调代理处理实体间的交互活动.协调代理要做两项工作:基于角色的访问控制检查和协调策略的实施.

3.1 基于角色的访问控制

这里我们说的角色就是软件实体在协调系统中能执行的操作的集合.协调组管理员根据协调系统的实际情况决定实体该具有哪些角色.当实体向组管理员注册时,管理员就会给它分配角色.同时管理员会将软件实体名和它对应的角色送至该软件实体的协调代理,以便协调代理执行访问控制的检查.协调系统中实体的角色可以被重新分配,角色也可以被赋予新的访问权限或被撤销权限.

3.2 协调策略的实施

协调策略是一个四元组(M, CG, R, P),其中 M 是该协调策略下所有交互信息的集合;CG 是协调策略所在的协调组;R 是在协调过程中信息交互要遵守的一系列规则,我们用 Prolog[6] 的方法来描述规则;P 是信息交互过程中起指导作用的协调代理.

图3描述了协调策略的实施在实体 A 和 B 的信息交互过程中是如何完成的.

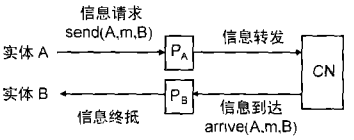

图3　协调策略实施的过程

为简单起见,我们在此只描述 A 发送信息给 B 这一过程中协调策略的实施,省略了访问控制检查,也即认为 A 有权限给 B 发送信息:(1)首先,A 把发送信息 *m* 到 B 的请求 *send*(A, *m*, B)传送给它的协调代理 PA,PA 根据协调策略中的规则检查这个请求能否继续传给 B,如可以,PA 把信息 *m* 转发到 B 的协调代理 PB;(2)然后,信息 *m* 到达 PB 时,在 PB 端生成一个事件 arrive(A, *m*, B),这时在 PB 端再次进行协调策略规则的检查,最后符合规则的信息 *m* 到达 B.

3.3 RBPEC 模型的特点

3.3.1 安全性　实体之间的通信首先经过基于角色的访问控制检查,然后再经过协调策略的约束.协调策略本身也可以把系统安全因素考虑在内.另外,协调组中的软件实体和元组空间完成注册以后并不需要经常与组管理员交流,只有当系统的角色分配方案或协调策略改变时,组管理员需要通知协调代理相应的改动.因此,即使协调组中的管理员因为故障不能正常运行,也并不会必然导致整个协调系统的瘫痪.

3.3.2 可扩展性　RBPEC 模型中,多个软件实体或元组空间可以共享一个协调代理,整个协调系统中的协调代理数目可以增加或缩减.一个软件实体可以同时加入不同的协调组,多个不同的协调组可以同时工作.角色分配和协调策略可以修

改和扩展.只要新加入的策略与协调组当前使用的策略不存在冲突,协调策略就可以加到协调组中.协调策略和角色管理的灵活性以及协调代理的可增减性使 RBPEC 模型能充分适用于开放分布环境中.

4 实例:分布式投标系统

一个安全的分布式的投标系统有以下几个步骤的通信:(1)当客户 c 请求服务 s 时,它把请求信息以元组(*request*(*c*),*service*(*s*))的形式写入元组空间 *ts*;(2)服务提供商 *sp* 把对服务请求的投标信息也写入元组空间,指明它提供服务 *s* 的有关信息,如价格、联系方法和提供商名称等.格式为(*offerFor*(*c*,*s*),*serviceProvider*(*sp*),*fee*(*f*),*contact*(*address*));(3)最后,客户就从投标的服务提供商中选择合适的提供商,通过投标元组中的相关信息与它联系.

为了确保系统中信息交互的安全性,首先,表示服务请求的元组和服务提供商的投标元组不能伪造;其次,客户 c 的服务请求元组对所有的服务提供商都是可见的,但对其他的客户是不可见的.并且,客户只能删除由它自己发出的服务请求元组;还有,对某个服务请求的投标元组只对提出该请求的客户是可见的,其他客户或服务提供商无权访问到此投标元组.

我们把这分布式的投标系统描述为一个协调组 $CG_{bid}=(ENTITY_{bid},GM_{bid},PROXY_{bid},CN_{bid},TS_{bid})$,$ENTITY_{bid}$包括该投标系统的所有客户和服务提供商,$TS_{bid}$是元组空间的集合,$PROXY_{bid}$是 $ENTITY_{bid}$和 TS_{bid}对应的协调代理的集合,GM_{bid}是协调系统的组管理员,CN_{bid}指通信网络.

整个投标系统中只有两种角色:客户和服务提供商.根据系统的限制条件得到每种角色被赋予的权限.客户只能读写自己的请求信息,无权读写其他客户的信息;服务提供商能读取任何客户的请求信息,但只能读写它自己的投标信息.整个系统的角色分配比较简单,因此我们就把它集成到下面的协调策略中.

```
R1: send(c, out(request(c'), service(s)), ts)
    :- c = = c', do(forward)
客户只能发出自己的请求信息
R2: send(c, in(request(c'), service(s)), ts)
    :- c = = c', do(forward)
客户只能读取和删除自己发出的请求信息
R3: send(sp, rd(request(c), service(s)), ts)
    :- member(sp, serviceProvider),
    do(forward)
服务提供者能读取所有的服务请求信息
R4: send(sp1, out(offerfor(c, s)fee(f)
        provider(sp2), contact(Address)), ts)
        :sp1 = = sp2, member(sp1, serviceProvider),
        do(forward)
服务提供者能发出自己的投标信息
R5: send(c, in(offerfor(c', s)fee(f), provider(p),
        contct(Address)), ts)
    :- c = = c', do(forward)
客户只能读取和删除属于自己的投标信息
R6: send(ts, -, -):- do(forward)
无组空间的任何请求都将被直接转发
R7: arrive(-, -, -):- do(deliver)
一旦有信息到达,就被直接发送
```

图 4 投标系统协调策略

协调策略的规则是用 Prolog 的规则来描述,投标系统协调策略的规则如图 4 所示,其中 member 是角色判断操作.

规则 *R*1 确保客户不能伪造其他客户的请求.*R*2 规定只有提出服务请求的客户能读取并删除该请求,*R*3 表示服务提供商可以读取任意客户的服务请求,但不能做删除操作.*R*4 表明服务提供商不能伪装成其他提供商.*R*5 表示与客户 *c* 的服务请求对应的所有投标元组,只能由客户 *c* 读取.在本例中,规则 *R*6 和 *R*7 之所以没有加入约束条件,是因为所有的约束条件在信息的发送端都已经完成了.规则中的 in、out、rd 的含义与 Linda 协调原语相同.

5 RBPEC 的效率

我们用实体间信息交互的相对开销来衡量 RBPEC 的效率.相对开销 $ro_{A,B}$定义为实体 *A* 和 *B* 在 RBPEC 模型中的消息传送时间 $RBPEC_{A,B}$与 *A* 和 *B* 直接消息传递所用的时间 $Msg_{A,B}$之差与 *A* 和 *B* 直接消息传递所用时间的比值.即

$$ro_{A,B}=(RBPEC_{A,B}-Msg_{A,B})/Msg_{A,B} \tag{1}$$

在 RBPEC 模型中,*A* 与 *B* 之间的信息交互要经过 *A* 和 *B* 的协调代理 P_A 和 P_B,*A* 到 *B* 的信息发送过程事实上有三步:$A->P_A$;$P_A->P_B$;$P_B->B$.整个过程所需的时间为

$$RBPEC_{A,B}=t_{com}^{A,P_A}+t_{eval}^{send}+t_{com}^{P_A,P_B}+t_{eval}^{arrive}+t_{com}^{P_B,B} \tag{2}$$

其中 t_{eval}^{send}和 t_{eval}^{arrive}表示协调代理本身处理 *send* 和 *arrive* 事件所需的时间.而 $t_{com}^{x,y}$表示从 *x* 到 *y* 的通信传输所需的时间.通常这个通信传输时间往往依赖很多实际的因素,比如采用的通信协议、信息的长度以及通信距离等等.这里我们的试验时间值,列于表 2.其中 t_{pipe}表示在同一台主机上的两个实体采用管道通信所用的时间;t_{LAN}表示通信实体位于局域网中,通信协议采用 TCP/IP 时所需的通信时间;而 t_{WAN}表示在广域网中需要的时间.t_{evalP}是在 RBPEC 中事件平均处理时间,用我们的测试结果值 2ms.(注:我们试验环境是在局域网环境中,一组软件实体和对应协调代理在 windows 2k,另一组软件实体和协调代理在 Solaris 7 平台上)

我们给出集中式协调模型 CCM 的相对开销 $ro_{A,B}'$作为参考,表 2 中的 t_{evalC}是 CCM 中协调器的事件处理时间,我们把它取为 t_{evalP}的两倍 4ms.因为在 CCM 中所有的信息交互都是通过这个协调器来完成的,协调器的负荷较重而且它的实现与 RBPEC 中协调代理相比要复杂的多.

表 2 通信时间值(ms)

t_{pipe}	t_{LAN}	t_{WAN}	t_{evalP}	t_{evalC}
0.1	5	50	2	4

$$\begin{aligned}ro'_{A,B}&=((t_{com}^{A,C}+t_{eval}^{C}+t_{com}^{C,B})-t_{com}^{A,B})/t_{com}^{A,B}\\&=((t_{WAN}+t_{evalC}+t_{WAN})-t_{WAN})/t_{WAN}=1.08\end{aligned} \tag{3}$$

RBPEC 的效率与协调代理的分布相关.鉴于 RBPEC 协调模型用于开放分布系统中,下面的效率分析是以在广域网中

的实体 A 和 B 之间的通信为例.

(a)实体使用本地代理

软件实体对应的协调代理都在本机上,实体与协调代理之间通过最直接的管道通信,结合上面的公式(1)(2),相对开销为

$$ro_{A,B} = ((2t_{pipe} + 2t_{evalP} + t_{WAN}) - t_{WAN})/t_{WAN} = 0.84 \quad (4)$$

(b)实体使用远程协调代理

当本地没有协调代理可用时,须利用远程的协调代理,远程的协调代理可能出现在本地局域网中,也有可能出现在广域网中.这时相对开销为

$$ro_{A,B} = ((2t_{WAN} + 2t_{evalP} + t_{WAN}) - t_{WAN})/t_{WAN} = 2.08 \quad (5a)$$

$$ro_{A,B} = ((2t_{LAN} + 2t_{evalP} + t_{WAN}) - t_{WAN})/t_{WAN} = 0.28 \quad (5b)$$

式(5a)对应于协调代理在广域网中,式(5b)对应于协调代理在局域网中.

(c)两个软件实体共享协调代理

假设软件实体 A 和 B 对应的协调代理都是 P_{AB},这时代理间的通信时间就变为 0,这时的相对开销为

$$ro_{A,B} = ((2t_{WAN} + 2t_{evalP}) - t_{WAN})/t_{WAN} = 1.08 \quad (6)$$

这与式(3)中 CCM 的相对开销是相同的.其实不难看出,RBPEC 模型的协调组中如果只有一个协调代理,组内所有成员都把它作为对应的协调代理,这就变为 CCM.

从上面的三种情况分析可见,要提高效率,必须尽可能地使用本地协调代理,但这样会造成整个系统的协调代理数目的增加,从而导致系统中组管理员维护工作的增加;而很多实体共用一个协调代理又会有 CCM 的缺陷,共用的协调代理容易成为系统的瓶颈.所以理想的方法是根据协调系统的规模,合理地控制协调代理的数目,并且让实体的协调代理尽可能地与实体本身位于同一局域网内,这样才能取得较高的效率,也不损失 RBPEC 的可扩展能力.

6 相关工作和结论

80 年代以来,国内外的学者已经提出了很多协调模型[7],但是当前的协调模型还没有具体研究协调和访问控制的关系,我们提出的 RBPEC 协调模型讨论了这一点.目前有一些协调模型也在关注这个问题,特别是 N H Minsky 等提出的 LGI[8]和 M Cremonini 等提出的 TuCSoN[9].

LGI 为分布在开放网络中的软件实体间的交互提供了协调机制,RBPEC 模型与 LGI 协调机制在引入协调代理方面是类似的,RBPEC 中的协调代理对应 LGI 的控制器,但与 LGI 不同的是,RBPEC 模型把协调和访问控制结合起来,引入了基于角色的访问控制,从而增强了协调系统的安全性.

TuCSoN 把协调规则嵌入到元组空间的行为规范中.它并没有在整个协调系统中加入象 RBPEC 中的协调代理这样的外部组件,但是它在安全性方面作了让步.

RBPEC 模型基于下面的假设.第一,信息在网络上的传输是安全的,即软件实体间的交互信息在网络的传输过程中不被篡改;第二,协调代理的实现是正确的,它可以正确地发送和接收信息;第三,协调组管理员对协调组内的每个成员来说都是可信赖的.

本文提出了协调模型 RBPEC,这个模型首先把协调与访问控制紧密结合,其次引入了基于角色的访问控制加强协调系统的安全性检查,再则协调策略的实施可以在分布环境中进行.具备上述特性的 RBPEC 协调模型有两个显著优点:安全和可扩展.

参考文献:

[1] Naftaly H Minsky, Victoria Ungureanu. A mechanisam for establishing policies for electronic commerce[A]. In the 18th International Conference on Distributed Computing Systems (ICDCS)[C]. ICPCS, 1998.

[2] D Gelernter, N Carriero. Coordination languages and their significance [J]. Communication of the ACM, 1992, 35(2): 97 - 107.

[3] T W Malone, K Crowston. The interdisciplinary study of coordination [J]. ACM Computing Surveys, 1994, 26(1): 87 - 119.

[4] David Ferraiolo, Richard Kuhn. Role based access control[A]. In Proceedings of 15th NIST-NCSC National Computer Security Conference [C]. Baltimore, MD: NIST-NCSC, 1992.

[5] N Carriero, D Gelernter. Linda in context[J]. *Communications* of the ACM, 1989, 32(4): 444 - 458.

[6] Carolton McDonald, Masoud Yazdani. Prolog Programming-a Tutorial Introduction[M]. Oxford, Boston: Blackwell Scientific Publications, 1990.

[7] G A Papadopoulos, F Arbab. Coordination models and languages[J]. Advances in Computers, Academic Press, 1998, 46(The Engineering of Large Systems): 329 - 400.

[8] Naftaly H Minsky, Yaron M Minsky, Victoria Ungureanu. Safe tuplespace-based coordination in multi-agent systems[J]. Jounal of applied artificial Intelligence, 2001, 15(1): 11 - 33.

[9] M Cremonini, A Omicini, F Zambonelli. Coordination and access control in open distributed agent systems: The TuCSoN approach[A]. In Proceedings of 4th International Conference (COORDINATION 2000) on Coordination Languages and Models[C]. Limassol, Cyprus: COORDIUATION, 2000. 99 - 114.

作者简介:

张亚英　女,1975 年出生于江苏省吴江市,目前在上海交通大学攻读博士学位,主要研究方向是分布计算,网格和对等计算,网络协议分析.

尤晋元　男,1939 年出生于江苏常州市,上海交通大学教授,博士生导师,主要研究方向是操作系统,分布计算.