一种基于本体的 RBAC 模型的研究与设计①

Research and Design of an Ontology-Based RBAC Model

傅　鹂　陈庆锋　(重庆大学 软件学院 重庆 400044)

摘　要： 为了描述和整合各种策略以应用于不同的领域,本文提出了一种通用的基于本体的 RBAC 访问控制策略机制(OntoRBAC)，它能很好地支持 RBAC96 模型，并能对其进行一定程度地扩展。利用本体这个工具,能够为访问控制策略的制定提供更精确、更强大的描述能力,它能够很好地描述分布式系统中的不同安全策略，同时可以实现从语义层次上对不同策略的集成。本文提出了一种广义的权限管理模型，该模型通过引入本体来对企业业务层面上与操作相关的信息进行概念建模，实现了一个对具有业务内涵的、范围相当广泛的一类广义操作进行权限定义和管理的合适的表达框架和管理机制，并建立了相应的形式化模型。最后，对下一步的工作做了必要的展望。

关键词： 本体 策略 模型 网络安全 RBAC

访问控制关注的是只允许授权用户去访问服务或资源，在学术界和产业界的访问控制应用系统中，那些限制系统组件行为的策略正越来越受到欢迎和青睐。在策略领域层面，用策略来引导实体行为已经被广泛地使用在安全、管理甚至是网络路由等方面，它还经常使用在灵活性要求很高的系统中作为代理，也经常使用在服务和权限变化比较频繁的系统中。授权和访问控制策略定义了高级别的规则，以指明在什么样的条件下主题被允许去访问目标，在管理企业广泛的应用和分布式系统层面，基于管理的策略正变成一种极具前途的解决方案。

1　前言

随着网络应用的逐步深入，网络安全问题日益受到广泛关注。基于角色访问控制(RBAC)的出现越来越显示出替代传统的 DAC 和 MAC 模型的优势[1]。但是，传统的 RBAC 模型中，用户-角色指派通常由管理员手工完成，在用户数量庞大的系统中，为用户指派角色的工作量将十分巨大，而且正确性很难得到保障，随着研究的不断深入，一种基于规则的 RBAC 模型(RB-RBAC)被提出且得到很好的实现，也解决了传统 RBAC 模型的部分缺陷，给整个系统的用户-角色指派工作减轻了不少负担。但由于 RB-RBAC 必须使用额外的方法定义属性值间的偏序关系和角色间的继承关系，而且无法表达分布式环境下的动态用户-角色指派，缺少角色-权限指派策略定义，此外，在 RB-RBAC 模型中，由于用户分配角色的动态性和授权规则的抽象性，使得管理员很难直观了解用户分配了哪些角色和权限，不同的授权规则是否包含了相同的用户[2]。而且，在定义授权策略的过程中，管理员通常需要掌握授权规则间的关系以及预见加入新规则会对系统造成哪些影响，并发现和解决否定授权规则引起的授权冲突问题。故在缺乏其它机制支持的条件下，实现 RB-RBAC 的系统中定义和维护授权策略十分困难。

本文首先简要介绍了传统 RBAC 访问控制模型，随后总结了传统访问控制模型面临新的应用时所暴露出的诸多问题，从而得出建立一种广义的访问控制模型的必要性。然后依次对本体的相关概念、基于本体的 RBAC 策略描述方法做了必要的叙述,最后提出了一种基于本体的访问控制模型,并尝试着用 OWL 描述语言对 RBAC 模型中的角色、用户及权限本体进行了简要描述。

① 基金项目:国家自然科学基金青年基金项目(60604007)
收稿时间:2008-10-23

2　传统的基于角色的访问控制模型

2.1 RBAC96 模型

目前，对 RBAC 模型已有广泛的研究。提出的模型主要有：美国国家标准与技术局研究小组的 RBAC/Web 模型，Ravi S.Sandhu 等人提出的 RBAC96 模型等等。其中，RBAC96 模型较全面系统地描述了 RBAC 模型的多层次、多方面的意义，现已经得到广泛的认可。该模型主要包括 RBAC0、RBAC1、RBAC2 及 RBAC3 四个不同的层次[2]，该模型的基本授权模型如图 1 所示：

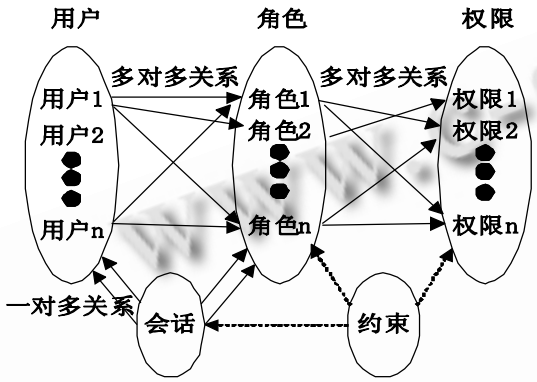

图 1　RBAC 基本授权模型

RBAC 模型与传统的访问控制相比，可以降低授权工作的繁琐程度。例如：用户数为 M，角色数为 L，权限数为 N。其中 M 的值可以很大，理论上没有上限，并且时常变化；因为角色是根据组织内部的职能岗位来设定的，所以 L 的值很有限，并且不受 M 的影响。N 也是有限的，同样不受 M 的影响。也就是说，角色和权限数量有限，且变化不大，因此将 N 种权限分配给 L 个角色，总共需要 L×N 次授权工作，且以后无需经常变动。对于每一个用户，只需要赋予其一个或多个角色。可见，在没有角色作为用户和权限中间桥梁的传统访问控制策略中，要做 M×N 次授权，其中 M 很大；而 RBAC 中，只需要 L×N＋M×L＝L×(M＋N)次，相对 M 而言 L 很小，这就大大减少用户授权的工作量[2]。

在传统的访问控制中，当某个权限发生改变时，需要对每一个拥有该权限的用户进行修改，而在 RBAC 中，则只需简单修改角色的权限，大大减少了授权更改的工作量。即使当组织内人员变动或者用户职能改变时，也只需简单的删除该用户原来的角色，重新分配新的角色即可。

综上所述，RBAC 模型摒弃了原有访问控制模型将用户与访问权限直接对应的做法，大大降低了用户权限变动所带来的复杂性和工作量，也减少了因各种变动对系统稳定性的影响。

2.2 传统访问控制模型面临新的挑战

传统的权限管理模型中，主体和客体是相互独立的，它们之间或者互不牵连，或者只通过一些共同的标记来进行联系。主体有主体的结构和标记，客体有客体的结构和标记，中间满足什么关系，就放行，否则就拦截，也就是说，只有主客体的安全相关属性才能用于授权决策中。但是，现代应用环境下的权限管理不同于传统的权限管理，需要面对若干新问题：

主客体营垒不甚分明，一个实体可能既是主体，又是客体。这样的主客体在传统的权限管理模型中无法表示[3]。

操作不仅是对文件和目录的简单的读、写和执行，而是一种广义的操作，通常都具有业务领域的内涵[3]。如对企业内部请假审批操作来说，必须要考虑主体的职位、客体的职位，请假的类型、天数，权限规定可审批的假期的类型、天数…等等。传统权限管理模型无法对这种带有语义信息的操作进行控制。

现代应用的访问控制有时需要考虑主客体的安全无关信息[4]。比如，对公共图书馆中某些材料的访问要根据访问用户的年龄来授予相应权限，传统的权限管理无法实现这种需求。

安全策略的表述有可能是跨应用的，策略的表达中涉及到一些公用的信息和结构，比如：时间、安全级别、职位等等。传统的权限管理模型没有提供这样的一种公共结构来表述这些共享信息[5]。此外，现代应用环境下还需要控制多种粒度的操作(包括基于用户的域名或 IP 地址、基于规则的操作等)，这样，必须需要多种权限管理层次，以实现各种粒度的权限控制。传统的权限管理模型同样无法解决这一问题。

可见，传统的权限管理模型已不能满足现代应用环境下权限管理的需要。为解决权限管理所面临的诸

多新问题，有必要设计一种新型的广义权限管理模型，用以解决对企业内部带有语义信息操作的控制以及跨应用复杂安全策略的设置问题。

3 基于本体的RBAC策略描述

3.1 本体论概述

本体论是对概念化的精确描述，最终目标是精确地表示那些隐含(或不明确的)信息，使得他们可被软件系统重用和共享。本体论是指包括一个领域中各类标准术语词汇，并对这些术语词汇进行准确定义，以及明确这些术语间的关系。本体论可以借鉴叙词不达意表的知识体系，借助语义相关和扩展记语言(XML)等信息技术，在增加术语相关性的基础上形成知识集成系统。从信息检索、情报语言角度看，这个概念解释更接近于我们要研究的内容。本体论的知识表示有多种，主要有语义网表示和描述逻辑表示，前者易于理解，国际 W3C 组织制定的互联网资源描述框架 RDF&CHEMA，就是基于语义网表示;后者表达力更强，由美国国防部主持的语义 Web 标注语言 DAML[6]就是采用了描述逻辑的表示技术。

3.2 基于本体的 RBAC 策略描述

RBAC 作为一种形式化、标准化,具有强大互操作能力的访问控制模型,成为了我们进行对访问控制策略描述的基础。同时借助了语义这个强有力的知识表达工具,对 RBAC 建立了一系列本体,作为策略描述的基本公理系统。

Entity (实体) :囊括了系统中可能涉及到的所有实体的抽象范畴,包括访问控制中的各种主体和客体对象[7];

Subject (主体) :访问控制过程中的主体对象,是访问权限的承载者、访问请求的发出者,在 RBAC 模型中又分为 Agent 和 Role 两个子类[8];

Agent (代理) : 是每一个访问请求的发出者以及会话(Session) 的创建者,可以被授予或自主地激活一定的角色，而根据具体的应用环境不同,可以由具体的用户或者是软件代理来充当,因此我们没有直接套用 RBAC96 模型里的User概念,而不失一般性地代之以 Agent ;

Role (角色) :RBAC 中的核心要素,依靠它将用户与权限联系起来,是 RBAC 模型中实施权限管理和进行访问控制判断的关键环节[8]。通过 juniorRoleOf 属性可以定义父角色与子角色间的关系,从而构造出实际应用系统中的角色层次树;

Resource (资源) :访问控制系统中主要的被保护客体对象,这里是一个抽象的范畴,根据访问控制所应用的不同环境,可以有更加具体的分类或表达形式,例如一个数据库中的表、视图,或者是 Web 应用中的一个网页或 Web 服务等[9];

Action (行为) :定义了系统中对资源可能进行的各种操作,这里的行为既可以是针对具体资源的访问操作(如:对数据库表的“修改”操作),也可以是通用性的安全管理方面操作(例如对某一权限的“授予”行为) ;

Privilege (权限) :描述了对某些特定对象执行某些操作的能力。由行为和操作对象两部分构成,操作对象就是访问控制中的客体资源;

Session (会话) :记录一个会话中的基本状态信息,如:请求者、已激活角色等;

Policy(策略) :一个访问控制策略的主体部分,由多条策略规则构成;

PolicyRule (策略规则) : 构成策略的基本单元,反映了 RBAC 模型中的授权关系,根据模型的特点又可以分为用户-角色授予规则(AgentRoleAssRule) 和角色-权限授予规则(RolePriAssRole) 两个子类。

Constraint (限制) :描述了 RBAC 模型中可能出现的各种限制的概念。这些限制可能涉及到角色、代理、策略规则等多种实体和对象。

4 基于本体的权限管理模型

4.1 模型描述

企业应用内部带有语义信息的广义操作要求现代企业的权限管理模型必须具有对这种语义信息的描述功能，同时，还能将这种语义信息用于操作执行时的权限判断。

本体是共享概念模型的明确的形式化规范说明，

它具有良好的概念层次结构和领域知识的共享性以及对逻辑推理的支持，故可考虑在权限管理模型中引入本体的概念，即使用本体这种工具为企业应用领域与操作相关的信息进行概念建模，借此反映操作的一些语义信息。当具体操作实现时，再将操作转化为对技术层面上带参数的服务的访问，并利用应用领域中的语义信息来进行访问控制[9]。图 2 是基于本体的权限管理模型。

从图 2 中可以看出，在基于本体的权限模型中，主体和客体都是本体的一部分，且它们都包含若干属性和属性值集合。此外，本体中还包含若干实体关系及约束。当主体对客体发出的操作请求被截获后，权限引擎根据相关规则以及这个操作所涉及的本体要素(包含操作所带的所有参数)之间是否满足策略所指定的约束条件，来决定该操作是被允许还是禁止。

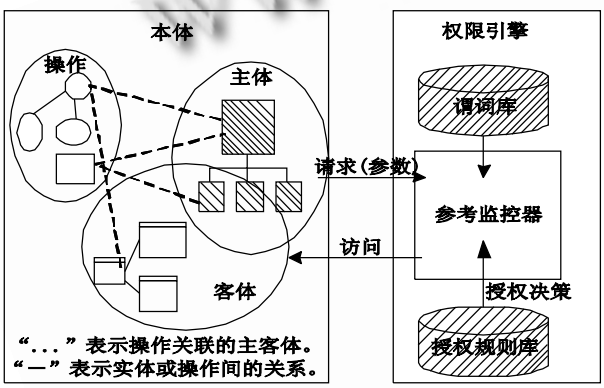

图 2 基于本体的权限管理模型

4.2 与传统权限管理模型的比较

传统的权限模型中，主客体间只是单纯的访问关系，不能表示它们之间的一些非访问关系，如报告关系、撤职关系等等，即传统的权限模型不能表示带业务语义信息的操作；传统的权限模型中，安全策略也只是规定了哪个主体可以访问哪个客体，访问粒度粗。为了细化访问粒度，有些模型增加了类似 BLP 模型中的安全标签这样的公共安全属性，来加强主体对客体访问的约束，但这种安全属性并不能涵盖所有的与策略表达相关的因素；另外，对于跨应用的策略来说，由于各个应用中所采用的信息的表示概念和术语并不相同，传统的权限模型很难对跨应用的安全策略进行表述。

基于本体的权限管理模型则把所有主体和客体都纳入本体体系中，主客体结构及它们之间的关系都是领域共享的，从而使得利用这种共享结构来统一表述各种复杂的带语义信息的操作以及跨应用的安全策略就成为可能；在基于本体的权限管理模型中，由于主体对客体的操作可以附带各种参数，且参数可引用主客体的所有属性，故基于本体的权限管理模型可以实现主体对客体细粒度的操作控制，即只有当操作所带的参数满足策略的约束条件时，才允许操作执行。

5 基于OWL的本体RBAC模型描述

5.1 OWL 介绍

OWL (Web Ontology Language)是 W3C 本体论工作小组提出的 Web 本体论语言规范,有 3 个子语言:OWLLite ,OWL DL ,OWL Full[10],这 3 种语言的语义表达能力是递增的。其中 OWL Lite 除了具有 RDFS 特性外,还具有简单的属性约束能力,比如描述属性特性(传递性、对称性、互逆性等) 以及对属性的基数进行约束;OWL DL 在 OWL Lite 的基础上引入了类型分割,它要求一个属性要么为对象属性(owl: ObjectProperty ,表示两个类的实例之间的关系) ;要么为数据类型属性(owl:DataType Property ,表示类实例和 XML datatype 之间的关系) ,其语义描述能力相当于描述逻辑,能够保证推理系统最大限度地计算出所有结论;OWL Full 包含了所有的 OWL , RDFS 词汇,比如类之间的操作(交、并、补) 等,此外它还允许在一个预定义的词汇表上添加词汇,能够提供最大限度的知识描述能力[10]。用户可以根据需要具体选择某种语言。

5.2 基于 OWL 的本体模型描述

5.2.1 用户本体

用户具有一系列的用户属性，而每种属性都是某种属性值的集合。可以为每个用户定义一个用户属性类，并相应定义一个属性的子属性来表达用户所具有的具体用户属性(可以根据应用环境确定),用户本体的 OWL 描述如下:

```
<owl:ObjectProperty
rdf:ID="hasUserAttribute">
<rdfs:subPropertyOf
rdf:resourse="#hasAttribute"/>
<rdfs:domain rdf:resourse="#User"/>
<rdfs:range rdf:resourse="#UserAttribute>
</owl:ObjectProperty>
<owl:Class rdf:ID="User">
<owl:subClassOf
rdf:resource="#PhysicalEntity"/>
</owl:Class>
<owl:Class rdf:ID="UserAttribute">
<owl:subClassOf
rdf:resource="#Attribute"/>
</owl:Class>
```

5.2.2 角色本体

角色可以看成是一些用户能执行的某些业务的集合。必须按照特定的要求定义角色：即根据业务视角定义角色，将角色作为一种进一步组织权限的机制而不是用户分组的机制。如系统中有 3 种角色 role1，role2，role3，并且角色 role3 继承了角色 role2 的权限，其描述描述如下：

```
<owl:Class rdf:ID="ROLE">
<owl:subClassOf
rdf:resource="#LogicEntity"/>
<owl:oneOf rdf:parseType="Collection">
<owl:Thing rdf:about="#role1"/>
<owl:Thing rdf:about="#role2"/>
<owl:Thing rdf:about="#role3"/>
</owl:oneOf>
<owl:subClassOf>\
<owl:Restriction>
<owl:onProperty
rdf:resource="#assignRole">
</owl:Restriction>
</owl:subClassOf>
</owl:Class>
<owl:Class rdf:ID="role3">
<owl:subClassOf rdf:resource="#role2"/>
</owl:Class>
```

5.2.3 权限本体

权限是在受保护的客体上执行某一动作的许可，它可以具有属性。权限应该具有通用的结构，我们将权限定义为一个逻辑实体， 具有权限属性。定义 PermissionAttribute 作为所有权限属性的父类并通过 hasPermissionAttribute 关联到权限，对象属性 performAction 表示权限相关的动作，其 OWL 描述如下：

```
<owl:ObjectProperty
rdf:ID="hasPermissionAttribute"/>
<rdfs:domain
rdf:resource="#Permission"/>
<rdfs:range
rdf:resource="#PermissionAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty
rdf:ID="performAction"/>
<rdfs:domain
rdf:resource="#Permission"/>
<rdfs:range rdf:resource="#Action"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Permission">
<owl:subClassOf
rdf:resource="#LogicalEntity "/>
</owl:Class>
<owl:Class rdf:ID="PermissionAttribute">
<owl:subClassOf  rdf:resource="#Attribute
"/>
</owl:Class>
```

6 结束语

为了解决企业应用内带语义信息的操作以及跨应用安全策略的设置和访问控制的问题，我们提出了一种基于本体结构的权限管理模型。我们使用本体来对业务层面上与操作相关的信息进行概念建模，实现了一个对具有业务内涵的、范围相当广泛的一类广义操作进行权限定义和管理的合适的表达框架和管理机

制，并建立了相应的形式化模型；另一方面，本体的使用将传统的访问控制对象扩展为带参数的、参数论域通过领域本体呈现出结构性关联(偏序结构)的、并且具有业务语义的服务，这样，扩大了访问控制的对象范围，丰富了相应的访问控制机制。

本文主要给出了一种基于本体的权限管理模型，并对这种广义的权限管理模型通过 OWL 进行了简要描述；运用该模型，通过 OWL 本体描述语言，用户可以构建自己领域内与权限相关信息的本体，从而利用该本体进行企业应用级的权限管理。总的来说，该模型虽然较传统的控制模型有了许多进步，但仍存在着许多值得研究的问题，例如：我们可以利用 SWRL (Semantic Web Rule Language)来描述我们所研究领域中的不同个体对象，从而简化访问控制模型的实现工作，这些工作都将在以后的实践中进行考虑解决。

参考文献

1 Woo T, Lam S. Authorizations in Distributed Systems: A New Approach. Journal of Computer Security,1993, 2(2&3):107－136.

2 Li DD, Hu SL, Bai S. A Uniform Model for Authorization and Access Control in Enterprise Information Platform. EDCIS, 2002:180－192.

3 王杰生,李舟军,李梦君.用描述逻辑进行语义 Web 服务组合.软件学报, 2008,19(4):967－980.

4 张东伟,赵津津,李鹏.基于语义网的知识管理研究.计算机与信息技术, 2008,(3):22－25.

5 都婧,封化民,何文才,孙茂增.基于 Ontology 的 Web 内容安全分析检测框架.西华大学学报(自然科学版), 2008,27(2):23－26.

6 王文璞,林木辉.基于本体的领域知识库构建方法研究.福建电脑, 2008,(8):131－132.

7 文坤梅,卢正鼎,吴杰文,李瑞轩,孙小林.基于描述逻辑的推理系统设计与实现.小型微型计算机系统, 2008,29(1):57－60.

8 文坤梅,卢正鼎,孙小林,李瑞轩.语义搜索研究综述.计算机科学, 2008,35(5):1－4.

9 何文才,都婧,封化民,孙茂增,张琼.基于 Ontology 的 Web 内容安全研究.网络安全技术与应用, 2008,(4):53－55.

10 王桢,程晟,程传业.基于 OWL 的电厂设备故障特征