第 34 卷 第 11 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.11 Computer Engineering June 2008
·博士论文· 文章编号：1000—3428(2008)11—0029—03 文献标识码：A 中图分类号：TP393

面向多客体的细粒度 RBAC 模型及应用

孔 芳[1,2]，朱巧明[1,2]
(1. 苏州大学计算机科学和技术学院，苏州 215006；2. 江苏省计算机信息处理技术重点实验室，苏州 215006)

摘 要：从综合信息管理系统以及软件系统集成的需求出发，分析了 NIST RBAC 模型的局限性，在此基础上对其模型进行扩充，提出 ERBAC 模型。该模型适用于多种客体且具体客体不确定的情况。文章还给出了 ERBAC 模型的具体定义和应用实例。
关键词：细粒度；多客体；访问控制；综合信息系统

Multi-object Oriented Fine Grain RBAC Model and Its Application

KONG Fang[1,2], ZHU Qiao-ming[1,2]
(1. School of Computer Science and Technology, Soochow University, Suzhou 215006;
2. Key Laboratory of Computer Information Processing Technology of Jiangsu Province, Suzhou 215006)

【Abstract】This paper analyses the requirements of comprehensive information management system and software integration. It proposes an extended model named ERBAC, which is based on the analysis of the shortcomings of NIST RBAC model. The extended model is applicable to the case of having multiple and unknown objects. And the definition and an instance of the ERBAC model are described.
【Key words】fine grain; multi-object; access control; comprehensive information management system

1 概述

随着信息技术，特别是网络技术的不断发展和广泛应用，各企事业单位、商业机构以及教育部门的信息化程度都大大增强。与工作效率大大提升同时出现的是一系列安全问题。近年来，基于角色的访问控制(RBAC)在信息系统安全控制方面得到了广泛的应用，其核心思想是在用户和权限之间引入角色的概念，为用户指派适当角色，为角色授予不同权限，用户通过激活角色行使相应的权限。

RBAC 的优点在于避免了直接为用户分配复杂的权限，简化了权限的管理。但随着信息化的推进，在进行软件系统集成或综合信息系统的设计时，RBAC 模型的局限性也日益显现。

本文在 RBAC2001 建议标准的参考模型(下称 NIST RBAC 模型)的基础上，结合综合信息系统以及软件系统集成的要求和特点，对 NIST RBAC 模型进行了扩充，给出了一个改进的模型。

2 RBAC 模型概述

2.1 RBAC 96 模型

文献[1]提出的RBAC 96 模型已经得到了广泛认可，该模型包含 4 个子模型：$RBAC_0$描述了任何支持RBAC的系统的最小要求，给出了客体、用户、角色、访问权限、会话、用户角色分配以及角色权限分配等基本概念。$RBAC_1$是对$RBAC_0$的扩充，增加了角色继承的概念。$RBAC_2$也是对$RBAC_0$的扩充，但与$RBAC_1$不同，$RBAC_2$加进了约束的概念。而$RBAC_3$是$RBAC_1$和$RBAC_2$的结合，将角色继承与约束结合起来就产生了继承之上的约束。模型讨论了几个最常用的约束：基数约束，互斥约束等。

在 RBAC96 的基础上文献[2]提出了 ARBAC97 管理模型。在该模型中将角色分成常规角色和管理员角色，两者是互斥的。管理员角色也有等级结构和权限继承。

2.2 NIST RBAC 模型

2001 年 8 月NIST发表了RBAC建议标准，该标准包含RBAC参考模型和功能规范 2 部分[3]。其中参考模型定义了RBAC的通用术语和模型构件，它包含 4 个层次：核心RBAC，定义了任何RBAC系统都应具备的要素，基本思想是通过角色建立用户和访问权限的多对多关系；层次RBAC，在核心RBAC上增加对角色层次的支持，而角色层次是一个严格意义上的偏序关系。有限层次RBAC则是指在偏序关系中加入一定约束，使层次结构趋于简单；静态职责分离(SSD)，用于解决角色系统中潜在的利益冲突的策略，而冲突源于用户被授予相互冲突的角色；动态职责分离(DSD)，与SSD类似，也可限制可提供给用户的访问权限，但实施的机制不同。DSD是在用户会话中对可激活的当前角色进行限制。另一方面，NIST RBAC的功能规范则负责将抽象的模型概念映射为可以提供管理操作、会话管理以及管理审查的功能需求。

2.3 NIST RBAC 模型的局限性

随着应用系统日趋复杂，以及各类软件系统集成工作的不断展开，NIST RBAC 模型的局限性也日益显现出来，主要体现在以下几个方面：

(1)在综合信息系统，或涉及多个软件系统集成的问题中，必然会面临角色过多的问题。例如高校内部的综合管理系统，涉及到学生、教师、教辅、财务、教务、校医院、宿舍管理处等众多人和部门，必然也会角色泛滥。在实际使用中，必然要求 RBAC 模型能够支持按类别、按部门，或者按照一定的依据划分成若干独立的单元，采用多级管理员体系，

基金项目：国家“863”计划基金资助项目(2006AA01Z147)；国家自然科学基金资助项目(60673041)
作者简介：孔 芳(1977－)，女，博士研究生，主研方向：中文信息处理，Web信息系统；朱巧明，教授
收稿日期：2007-06-05 **E-mail：**kongfang@suda.edu.cn

自上而下地逐级进行授权。

(2)随着计算机系统管理对象所涉及范围的不断扩大，客体的种类、特性也呈现出不确定性。各类资源都可能成为客体，就高校内部的管理而言，客体可以是某一应用系统中的一个功能，也可以是网络上的某一端口、IP 地址或者其他软硬件设备，或者可能是数字图书馆中相应的数字资源。客体的不确定性带来的问题是各类客体的操作权限各不相同，这必然要求 RBAC 模型能够支持功能操作与操作对象间的权限指派。

(3)在实际应用中，不同的用户即便使用相同的功能来操作相同或者不同的客体对象，也是存在着一定差别的，这就是所谓的级别问题。例如高校综合管理系统中，同样的成绩管理功能，甚至操作同一个年级的学生成绩。教务秘书的成绩管理既能对成绩进行各种形式的归类、排序、浏览，又能对数据进行更新、导入、导出。而年级辅导员的成绩管理功能仅包含浏览、排序，但无权更新。RBAC 模型必须细化对功能权限的控制粒度，提供为用户设置不同功能的不同级别的权限的方法。而具体的权限又随着客体的具体情况变化。

(4)在综合信息系统或集成系统中，客体是多种多样的，且都具有一定的周期特性。例如高校招生管理系统中的考生信息，在报考结束被录取后就成了新生数据，下一年报名期间又会产生新的考生信息。RBAC 模型必须进一步细化，能控制客体对象的粒度，同时体现周期特性。

3 模型的改进

针对 NIST RBAC 模型的局限性，一些研究者提出了对其进行改进的方法。文献[4]提出了适用于 Web 应用系统集成的 RBAC4WAS 模型，并基于该模型给出了一个面向 Web 应用集成的统一授权平台的解决方案。文献[5]借鉴 DTE 模型域和型的思想，提出通过引入主体和客体的属性参数以及虚拟权限来改进 RBAC 模型的解决方法。文献[6]将访问控制模型和基于任务的访问控制原理相结合，提出了一种基于角色和任务的访问控制模型。本文结合综合信息系统以及软件系统集成的要求和特点，给出一个改进的模型 (Extended RBAC, ERBAC)。

3.1 模型改进的基本思想

从上述对 NIST RBAC 模型局限性的分析可以看到，扩展模型 ERBAC 的定位是：

(1)面向不确定的多客体；

(2)适用于综合信息管理系统以及多个软件系统集成状况下，角色、权限非常多的情况；

(3)能对多种客体对象进行细粒度、多维度(分时间、空间、时序等)的访问控制。

3.1.1 面向不确定的多客体

客体多种多样，且具有不确定性和可扩充性(用户可以根据系统应用的实际需要，对客体对象进行维护)。在 ERBAC 模型中，使用了与将角色利用会话 Session 指派给用户这一过程相对称的模式，增加功能权限的指派过程。即在访问控制服务器端，新增一个(或一组，根据系统规模或实际的需求可以扩充)后台的代理监听器进程(或线程)。当有功能操作请求出现时，由代理监听器负责根据实际操作状况以及相应的访问权限获取相应的客体对象，并将合法的客体对象包装成统一的响应信息回复给发出请求的操作。

3.1.2 适用于复杂情况下的角色和权限的管理

随着企业信息化的不断推进，应用系统规模不断扩大，各类应用的功能、数据层的整合逐渐成为关注的焦点。系统规模的扩大以及软件系统集成需求的提出，在系统内部都会产生大量的角色、权限。虽然角色、权限众多，但在实际应用中，跨类型、跨部门、跨区域的角色却相对较少。因此，为了有效管理这些角色、权限，在 ERBAC 模型中作了如下 2 方面的改进：

(1)将角色层的角色分成类型角色和功能角色 2 类。类型角色对用户权限进行粗线条的切割，功能角色则对某一类型角色拥有的权限进行细粒度的分割。用户指派时，每个用户都将获得唯一的类型角色和多个功能角色。在这一扩展中，类型是一个抽象概念，它可以是企业的一个部门或一个子公司，可以是一类用户，也可以是客体资源的一个类别，在实际使用中可根据需要来设定类别。划分类型角色和功能角色有利于对角色进行分而治之的管理。顶层管理员只负责用户类型角色的分配，一旦分配完毕，相应用户的最高权限也设定了。接着由各类型的管理员设定自己类型下的功能角色。

(2)将权限层的权限分成类型权限和功能权限 2 种。类型权限又可细分成公共权限和最大权限 2 类。公共权限是指该类型的角色都拥有的一些基本权限，某种意义上是最低权限。而最大权限是指该类型的管理员所拥有的权限。而功能权限是某一具体功能在操作约束下，由代理监听器根据各种操作约束完成功能权限指派后得到的结果。

3.1.3 对多种客体对象进行细粒度和多维度的访问控制

类型权限只是对权限的有/无进行了粗线条的划分，而功能权限则根据客体对象的不同，操作约束的实际状况完成权限的细粒度控制。在对功能权限进行进一步控制时，首先对操作功能划分了操作级别。操作级别也是一个抽象概念，可以简单地用一个整数表示，其确切的含义可被代理监听器程序所理解。随着客体对象的不同，相同整数可能表示不同的含义，这一整数某种意义上规定了一类客体对象在这一级别下的最大功能权限。

客体对象是静态的数据、设备或者资源等，在交由某一功能进行操作时需要根据各类附加约束对客体对象进行包装。这些约束包括时间、空间以及功能操作的先后时序等。

3.2 模型的定义

根据 3.1 节给出的模型改进思想，得到了如图 1 所示的 ERBAC 模型。其定义如下：

(1)*USERS, ROLES, SESSION, TR, FR* 分别表示用户、角色、会话、类型角色、功能角色。

(2)*TP, FP, FERM, FOS, OBS, AGENTLISTENER* 分别表示类型权限、功能权限、访问权限、功能操作、操作对象、代理监听器实例。

(3) $ROLES = TR \cup FR$ ，角色集是类型角色集和功能角色集的并集。

(4) $\forall r \in ROLES \rightarrow \exists tr \in TR, fr \subseteq FR \wedge r = (tr, fr)$ ，任意角色 *r* 都是由某个类型角色 *tr* 和若干功能角色构成的功能角色子集 fr 的组合。

(5) $UA \subseteq USERS \times ROLES$ ，用户到角色的多对多映射：$assigned\_user(r) = \{u \in USER, r \in ROLE \mid (u, r) \in UA\}$ 。

(6) $PA \subseteq ROLES \times PERM$ ，角色到权限的多对多映射。由于角色分成类型角色和功能角色 2 种，因此这一映射过程包括类型角色到权限的映射和功能角色到权限的映射。

$assigned\_typerole(ptr) = \{tr \in TR, ptr \in TP \mid (tr, ptr) \in PA\}$ ，

$assigned\_funcrole(fptr) = \{fr \in FR, fptr \in FP \mid (fr, fptr) \in PA\}$

(7) $FPA \subseteq FOS \times OBS$，操作对象和功能操作之间的多对多关系，$assigned\_obs(f) = \{o \in OBS, f \in FOS \mid (o, f) \in FPA\}$。

(8) $user\_session(u : USER) \to 2^{SESSION}$，用户 u 和该用户激活的会话集对应的关系。

(9) $user\_session(s : SESSION) \to 2^{USER}$，会话 s 和该会话所属的用户集对应的关系。

(10) $session\_role(s : SESSION) \to 2^{ROLES}$，会话 s 和应当分配给该会话的用户角色集对应的关系：

$$session\_role(s_i) \subseteq \{r \in ROLES \mid (session\_user(s_i), r) \in UA\}$$

(11) $fos\_agentlistener(fos : FOS) \to 2^{AGENTLISTENER}$，功能操作 fos 和该功能操作激活的代理监听实例集的对应关系。

(12) $agentlistener\_fos(agtlst : AGENTLISTENER) \to 2^{FOS}$，代理监听器实例 $agtlst$ 和该监听器实例所属功能操作集的对应关系。

(13) $agentlistener\_obs(agtlst : AGENTLISTENER) \to 2^{OBS}$，代理监听器实例 $agtlst$ 和应当分配给该代理监听器实例集对应的关系：

$$agentlistener\_obs(a_i) \subseteq \{o \in OBS \mid (agentlistener\_fos(a_i), o) \in FPA\}$$

(14)$FUNC$, $LEVEL$, $DATA$, $PERIOD$ 分别表示操作功能集、操作级别集、客体对象集、周期集。其中操作功能集是指在系统中所涉及的功能或使用方式，例如防火墙的进入、端口的占用，或者报表审批、用户管理等功能菜单；操作级别集是指查询、添加、禁用等；客体对象集是指在系统中涉及的软硬件资源、数据对象等；周期集是指在系统中具体客体对象的周期，例如 17:00~21:00 开放某一资源等。

(15) $OPS = 2^{FUNC \times LEVEL}$，功能的访问许可集。

(16) $OBS = 2^{DATA \times PERIOD}$，代表系统中所有功能操作的客体对象的集合。

(17)OC：操作约束集，主要包括时间(非周期性的临时设定的时间段)、空间(不同的网段)和时序(完成操作的前后顺序等)。

(18)C：约束集，不同角色间、用户与角色动态绑定时的制约条件。其中，互斥约束指一个用户只能同时获得互斥角色中的一个；基本限制规定了某个角色可同时被分配的最大用户数；必备约束是指某一用户只有已经同时获得了另一些角色才可以获得当前这一角色；时间和空间的约束主要是获得该角色的时间段和网络段的约束条件。

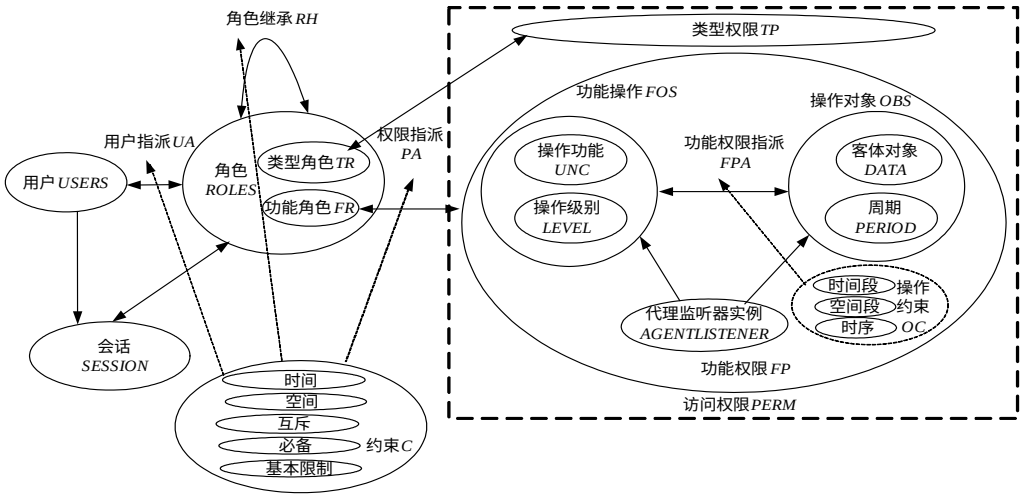

图 1　ERBAC 模型

3.3　应用实例

在实际应用中，ERBAC 模型与 NIST RBAC 模型相比能更好地适应客体对象的变化；同时由于细粒度访问控制的引入，能更好地对客体对象进行多维度的访问控制。下面以某高校研究生综合管理信息系统为例来简要介绍 ERBAC 模型的应用。

高校研究生综合管理系统负责管理各类研究生(包括全日制统招、专业学位、教师进修等多种类别)从招生报名开始，经历录取入校、制定培养计划，到最终获得学位学历的全过程。

在 ERBAC 模型中，根据用户群、机构部门作粗线条的分割，形成如图 2 所示的几种类型角色。

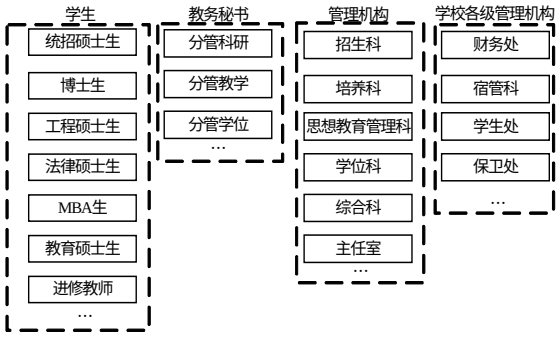

图 2　类型角色集

由顶层管理员为每种类型角色设定类型权限，包括公共权限和最大权限。由此形成了第 1 层的管理员(他们拥有所属类型角色的最大权限)，再由他们进行类型角色内的子类型角色和功能角色以及本类型公共权限的管理。

功能角色拥有一定的功能权限，而功能权限的指派过程是一个将功能操作与操作的客体具体绑定的过程。研究生综合管理系统中，客体对象是数据库中存储的数据，但实际使用中也可以是具体的硬件设备、端口等。利用功能权限配置模块设定功能操作与客体对象的绑定规则，形成 XML 形式的一个或一组规则文件。启动服务器端代理监听器实例时，不同实例读取各自的 XML 文件，形成 Hash 表常驻内存。当监听到某一权限的请求时，在经过约束合法性检验后，根据 Hash 表中的映射关系进行功能操作和客体对象的绑定。

4　结束语

本文从模型改进的基本思想、模型的定义和模型的应用实例 3 方面展开，对 ERBAC 模型进行了详细的阐述。目前该模型已成功地应用于研究生综合信息管理系统、数字化校园之共享数据中心、基于网格的统一资源管理系统等多个系统的设计与开发中，效果良好。ERBAC 模型对综合信息系统中以及软件系统集成状况下的统一访问控制系统的设计与开发具有一定的参考价值。

（下转第 34 页）