文章编号：1000-8608(2012)03-0452-06

实现细粒度访问控制的元模型研究

王　宁*，钞　柯，罗　嫔

（大连理工大学 管理科学与工程学院，辽宁 大连　116024）

摘要：从用户要求对数据对象达到细粒度访问控制的需求出发，借助元模型的概念，对基于OB4LAC的权限维度进行细粒度扩充，充分融合数据对象的时间期限、安全限制、上下文环境等属性特征，建立了关于数据对象和操作客体的细粒度元模型，并探讨了两者的粒度归并规则，提炼出功能权限集和功能菜单.由此为系统角色的界定和划分提供参照标准，更加方便了角色的统一管理，达到灵活的细粒度访问控制目标，有利于制定优化的访问控制策略.

关键词：访问控制；细粒度；数据对象；操作级别；元模型

中图分类号：TP315　　**文献标志码**：A

0　引　言

通过某种途径限制主体访问关键数据对象客体的访问控制技术是保障信息安全的关键.当前，RBAC(role-based access control)已经成为目前最为流行的一种访问控制策略.1996年，Sandhu等提出了RBAC96和ARBAC97(administration of RBAC，基于角色的访问控制管理)访问控制模型[1]，在访问控制研究领域取得了重大的突破.其核心思想是将访问权限与角色相联系，它包含一组用户集和角色集，将角色作为一个桥梁，沟通于用户和资源之间.由于实现了用户与访问权限的逻辑分离，基于角色的策略极大地方便了权限管理[2].

但是，传统的RBAC模型控制粒度较大，对软件系统更高的数据安全要求以及系统个性化需求不能很好地适应.细粒度访问控制与粗粒度访问控制的主要不同点是：细粒度访问控制过程除了需要考虑客体的类别，还会考虑操作客体、用户主体、资源等具体实例的细节问题，并且综合上下文环境限制与时间周期性约束，从而达到属性层或者数据资源元组层的控制.然而，粗粒度的访问控制过程只考虑到客体的类别，并不区分客体具体的实例[3,4].近年来国内外学者针对访问控制中的粒度问题也有了一些研究，文献[5]通过引入组件外权限和组件内权限，把访问控制粒度细化到组件内部业务数据的操作和属性层次，以实现组件重用.文献[6]结合RBAC模型思想和大型企业信息系统的实际需求，对核心RBAC模型进行了细粒度的扩充，在单位、功能、数据等多个维度对模型进行了细化，实现了大型信息系统的多级管理员体系，使得系统的功能及数据的控制更加灵活.文献[7]从Web实现方面对访问控制的细粒度问题进行了研究.这些针对访问控制中粒度的研究主要集中在某些环境下的具体实现上，还缺乏访问控制元模型方面的研究，而如何有效地划分、描述、控制访问控制粒度，进而达到细粒度的访问控制，即用户对信息系统中具体的操作功能权限或者每一个被管理的对象进行控制，将对访问控制的管理优化具有重大意义，这也正是目前相关组织机构对访问控制更深层次的要求[8,9].

细致全面地考虑操作权限所涉及的数据对象和操作客体之间的相互作用，以及二者之间的约束条件，是系统实现细粒度访问控制的关键所在.本文将从上述两方面的关键点来分别构建数据对象和操作客体的细粒度元模型，把数据对象资源

收稿日期：2009-04-04；　**修回日期**：2012-03-04.
基金项目：国家自然科学基金资助项目(91024029)；中央高校基本科研业务费专项资金资助项目(DUT10JR11).
作者简介：王　宁*(1973-)，男，博士，副教授，E-mail：wn@dlut.edu.cn.

归纳为一组概念以及概念间的联系，抽象出数据对象与操作客体的不同粒度以及粒度关联的元数据统一描述模型.并且在粒度细化的元模型基础上，按照粗细粒度访问控制的具体实际要求，定义数据对象与操作客体的归并规则，为之后的功能权限集的构成、角色定义以及之间的层次关系确定提供依据，从而制定优化的访问控制策略.

1　元模型的建立

元数据是信息资源管理和共享服务的基础，可以将其理解为数据的数据，为描述某一类别资源属性，提供关于数据对象定位、管理和精确检索服务的结构化数据[10].为了更好地解决信息系统研发过程中的异构现象，并且达到业务协同、信息系统共享以及应用集成的目标，需要建立元数据模型，在更高层面上定义元数据的数据结构、设计方法、格式、语义规则、语法规则、功能等内容[11].访问控制粒度的元模型则是用来描述访问控制粒度的数据模型[12].

1.1　信息资源的细粒度元模型

从当前的政务信息系统来看，政务相关的信息资源主要包括两个部分：政府服务资源与文本资源，主要分布在政府部门的文件系统、单、表、账、业务数据库、网页等数字化空间，大部分来自报表审批、公文流转等环节.

在现实的信息系统中，用户主体对不同的数据对象客体具有不同粒度的操作级别.数据对象，即信息资源，其粒度的粗细程度，是由政务系统服务的具体要求决定的，有以下几种粒度划分方法.

基于数据内容划分粒度：例如一个数据库存储了某个市级项目申报的具体情况，如果是用户自己申报的项目，那么针对该项目的各相关信息，用户都拥有查看与修改的权限；如果是其他用户申报的项目，用户就只拥有浏览项目名称、申报人等项目简介信息的权限.但是如果该用户的身份是审批者，那么就拥有查阅所有申报人的项目细则权限，并且可以进行审批批示，流转至下一位审批者.此外，用户有权关注的数据对象相异，例如财务人员处理公文流转相关的事项很少.

基于时间周期划分粒度：例如公文流转事项，一份公文只在某一规定的时间内有效，在该时间段之外，相关责任者查看或者批示该公文的权限则被收回；再如，财务相关人员在每个季度以及会计年度的规定日期编写会计报表，其相应的查看权限也会受限.

基于上下文环境划分粒度：以采购部门为例，采购人员首先要获取各部门提交的已通过审批的采购申请，然后才能编制采购清单；再如，公文流转过程中，A负责人先对其进行审批，之后才会流转到B负责人手中.

因此，访问控制模型需要对数据对象的控制粒度进一步细化，为用户提供设置详细的数据属性的具体方法，并且充分考虑数据对象的属性特性，如数据时间周期、数据内容以及数据类型等.

与以上数据信息资源有关的元数据都是通过“实体-关系-属性”这一面向对象三元组的形式来表示的，以此为基础来建立数据对象的细粒度模型.用于描述数据自身的元数据就是这个三元组中的“属性”，它主要包括一些与数据本身相关的信息，诸如时间周期、标识信息、发布信息、生成时间、质量信息之类的现有元数据标准.用于描述数据间关联的元数据就是三元组中的“关系”，可以将其理解为通过怎样的关系将各个数据集构成一个有机整体，这种关系包括组成关系、从属关系等.图1为数据对象的细粒度元模型.

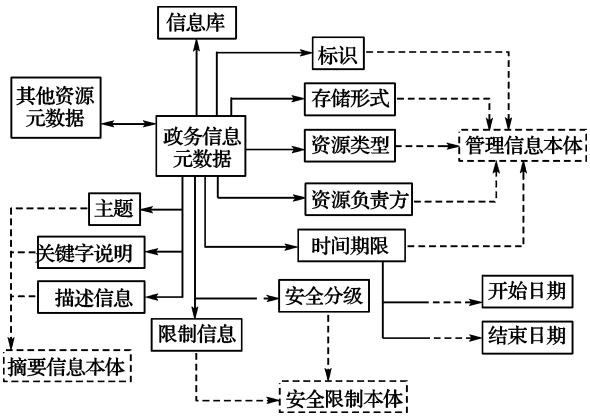

图1　数据对象的细粒度元模型

Fig.1　Fine-granularity meta-model about data object

这里可以把图1中的数据对象的细粒度元模型(ISMM)抽象成为一个六元组：

$$ISMM = \langle OBE, MS, AS, SS, OBOS, OBR \rangle$$

OBE 表示资源对象元模型实体集，obe_id、

obe_name 为其相应的用来区分数据对象元数据的标识.

MS(management set)是管理信息集,AS(abstract set)是摘要信息集,SS(security set)则为安全限制集,三者构成了三大类属性集合.

MS 表示基本信息管理属性集合,例如标识(sign)、资源类型(datatype)、资源负责方(principle)、资源存储形式(storage)和时间期限(beginday, endday). $\forall ms \in MS$, $ms = \langle sign, datatype, principle, storage, beginday, endday \rangle$.

AS 表示摘要信息相关的属性集合. 例如关键字说明(keywords)、主题(theme)、描述信息(describInformation)等. $\forall as \in AS$, $as = \langle keywords, theme, describInformation \rangle$.

SS 表示安全限制分级和其他限制信息. 诸如信息的使用范围(useRange)、元数据的安全等级(securityGrade)、信息的使用人员(users)、上下文环境限制(context)、互斥信息(mutexInfo)等都可以用 SS 来标识. $\forall ss \in SS$, $ss = \langle useRange, securityGrade, users, context, mutexInfo \rangle$.

OBOS 表示数据对象的本体信息集,OBOS 可以明确定义出信息资源领域内认同的相关词汇术语以及词汇间的关系,因而信息语义的异构问题能得以很好的解决. OBOS 主要包括三类本体信息:安全信息本体、摘要信息本体、管理信息本体,它从本体的内部属性以及属性之间的关联两方面来加以描述.

数据对象实体间以及属性间的关联可以通过 OBR 来描述. $\forall obr \in OBR$, 可以表示为 $obr(c_1, c_2), c_1, c_2 \in OBE$,比如实体资源 c_1 和 c_2 是互斥关系,表示为 obr(互斥)(c_1, c_2).

1.2 操作客体的细粒度元模型

在操作客体层面,需要将访问控制策略中关于功能操作权限的控制粒度加以细化,针对不同的系统功能,用户可以为其设置相异级别的操作权限. 为了真正体现用户在信息系统中拥有的功能权限差异性,将每个角色的操作权限进一步细化.

操作级别是操作客体粒度细化的主要方面,不同操作级别之间的关系不容忽视,例如同一数据对象是否能够同时赋予两个或者两个以上的操作级别. 针对不同的数据对象,操作级别的细化也有所不同:

针对财务记录,与财务报表编制相关的操作级别有借、贷、创建、删除账户等.

针对数据库表记录,诸如薪酬管理、用户管理等模块的数据维护,有插入、删除、修改记录等操作级别.

针对一般公文、文档,操作级别主要包括修改、删除、执行、查看等.

总之,无论是哪种数据对象,都可以总结出 5 种操作级别:增、删、改、查、禁用. 操作客体的细粒度元模型表示如图 2 所示.

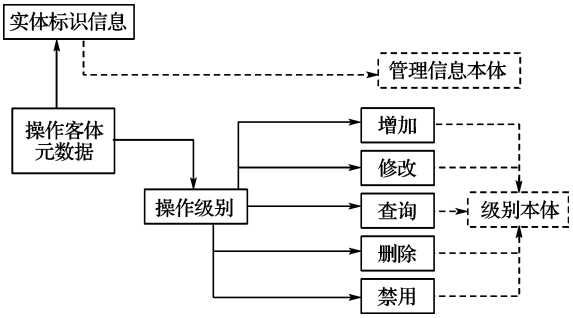

图 2 操作客体的细粒度元模型

Fig. 2 Fine-granularity meta-model about operation object

现将图 2 所示的操作客体的细粒度元模型(OOMM)抽象为一个四元组:

$$OOMM = \langle OPE, OPLS, OPOS, OPR \rangle$$

OPE 用来描述操作客体细粒度元模型实体集,$opeId$、$opeName$ 为相应的用来区分操作客体元数据的标识.

OPLS 用来描述操作级别集,包括增(add)、删(delete)、改(modify)、查(query)、禁用(disabled)5 种操作级别. 可表示为 $OPLS = \langle add, delete, modify, query, disabled \rangle$.

OPOS 用来描述包括操作级别本体和管理信息本体在内的操作客体本体信息集,刻画属性信息以及各个定义间的关系是它的重点之一.

OPR 用来描述操作客体以及属性之间的关系. $\forall opr \in OPR$,可以表示为 $opr(a_1, a_2), a_1, a_2 \in OPLS$,比如操作级别查询 a_1 和禁用 a_2 是互不相容的关系,表示为 opr(互斥)(a_1, a_2).

2　粒度归并规则

2.1　数据对象归并

数据对象的状态、属性、内部关联和约束关系在数据对象的细粒度元模型中都被综合考虑了，数据对象能够被细化到数据库中单条记录级别或者目录级别，从属性层或者元组级别层对数据项进行控制，或者将其归并为报表、文档级别，通过粗粒度控制整个数据表或者文件.

一些元组级别信息资源的细粒度应当根据具体的工作情况加以控制，如果粒度太细，或者这些对应资源更加适合粗粒度控制，都会在一定程度上造成系统冗余. 因此，为了提高系统工作效率，同时又能达到用户要求的细粒度访问控制，可以把数据对象归并到一个适应现实需求的粒度范围.

以数据对象的细粒度元模型标准作为依据，每个数据实体通过信息资源的属性以及关联关系进行描述，形成数据对象集的基础粒度，并且根据某一具体的标准将其归并为数据对象的个体(OB). 数据对象的安全分级、存储形式等属性都是粒度归并方法需要参考的因素. 本体中的粒度计算标准与概念分类是数据对象归并规则的制定依据，首先对信息粒进行定义，描述其形成，接着制定规则把信息粒归并为数据操作对象. 表示如下：

$g(i)$用来表示数据对象的基础粒度，把数据对象细粒度元模型中的管理信息集、摘要信息集、安全限制集分别赋予每一个实体概念，那么数据对象的基础粒度与单个实体相对应.

(1) $MS, AS, SS \in ISMM$，$MS, AS, SS \rightarrow OB \rightarrow g(i)$.

(2) $WT[g(i)] = F(\alpha(secRank), \beta(keywords), \gamma(datatype))$

$$sim(g(i), g(j)) = \frac{\phi}{\phi + d(WT[g(i)], WT[g(j)])}$$

其中 WT 用来表示信息粒的权重，其赋值函数为 F，α、β、γ 为函数的赋值因子，分别表示安全限制分级(secRank)、关键字(keywords)、数据类型(datatype)；$sim(g(i), g(j))$ 用来描述信息粒 $g(i)$ 与 $g(j)$ 的相似度；$d(WT[g(i)], WT[g(j)])$ 表示 $g(i)$ 与 $g(j)$ 的距离，通过权重与麦考斯基距离获取；ϕ 是按照具体情况设置的动态因子.

通过以上方式能够获取对应的相似度值，如果指标范围是确定的，计算所得相似度在该范围以内，则将其归并到同一数据对象个体($obs(i)$)中；否则不予归并. 那么数据对象集可以用如下形式表示：

$$OBS = \langle obs(1), obs(2), \cdots, obs(i) \cdots, obs(n) \rangle$$

2.2　操作客体归并

参照操作客体细粒度元模型标准，操作客体与操作级别相对应，操作客体集则是由操作级别中各元素遵循一定的组合方式归并而形成的，表示如下：

$$Length(OPLS) = n$$

$$m = (c_n^1 + c_n^2 + \cdots + c_n^n) - \eta = 2^n - 1 - \eta$$

这里，n 表示操作级别集 $OPLS$ 中包含 n 个元素，m 表示归并之后的元素个数. 操作级别的元素组合子集有 $2^n - 1$ 个，不满足实际具体情况的元素组合个数为 η，那么二者之差即为 m 的值. 例如：禁用操作级别与修改操作级别的组合就不满足实际情况.

所以，操作客体集表示如下：

$$OPS = \langle ops(1), ops(2), \cdots, ops(i), \cdots, ops(m) \rangle$$

3　基于控制粒度的权限-角色指派

操作客体与数据对象是权限的两个重要元素，上文提出的细粒度元模型从这两个层面实现了权限的细化. 在把权限合理地分配给相应的角色过程中，为了更加清晰地界定角色，更好地与现实应用条件衔接，可以增加两个中间过渡层次：功能权限集(Function)和功能菜单(Menu)，从而实现对信息资源的细粒度访问控制，为权限完美授权路线提供支持. 权限-角色分配流程如图 3 所示.

3.1　功能权限集

操作客体集(OPS)和数据对象集(OBS)的互操作组合成权限集(P). 权限就是对数据对象的操作许可，例如对财务报表的操作、公文的审批和工资单的计算汇总等. 以图 4 采购为例，有对下个季度的部门采购计划表的增删改查，对当月采购单的执行、状态处理等权限. 这些都是细化的基础权限，表示如下：

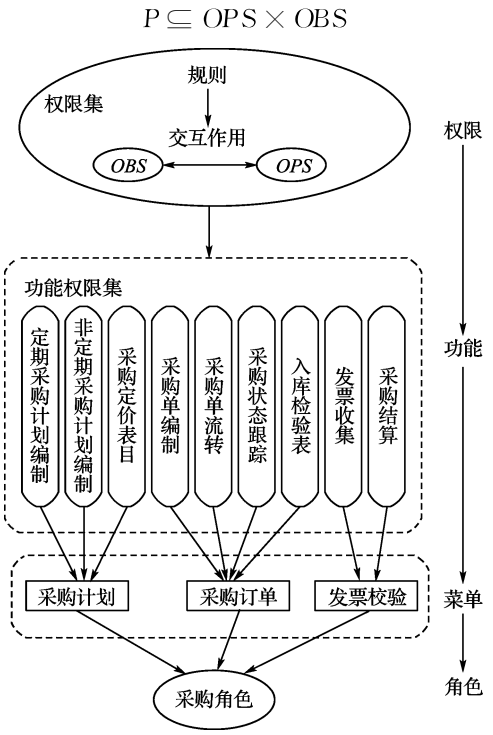

图 3 “权限-角色”分配流程

Fig. 3 Permissons transmitting path of "Permission-Role"

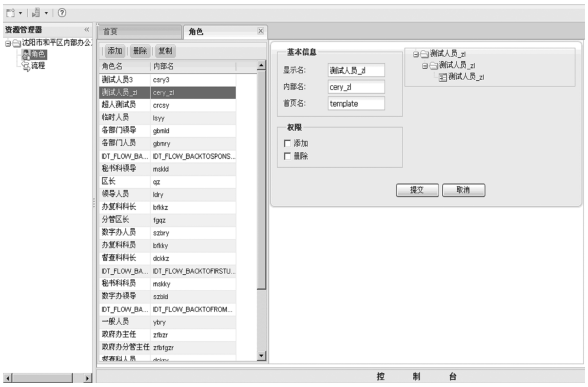

图 4 协同办公系统中访问控制管理的角色指派界面

Fig. 4 The page of permission-role of access control management in cooperation office system

功能权限集可以理解为基础权限的集合，它描述的是政府部门在现实工作中实现某个具体功能所需要的权限集合，一个功能权限对应多个基础权限. 例如非定期采购计划编制作为一个功能权限，它包含了预算控制、审核临时购买清单、修改采购计划等多个基础权限. 这里把功能权限定义为一个四元组：

$$FUNC = \langle FID, OPS, OBS, RTR \rangle$$

功能权限通过 FID 来标识，操作级别的映射用 OPS 来描述，OBS 则描述了数据对象的映射，诸如时间周期、上下文环境、安全级别等从数据对象属性中抽象出来的相关限制条件可以用 RTR 来描述. 针对相异的数据对象以及相异的现实环境设置，其限制条件也会有所不同.

3.2 角色界定

在实际应用系统中，功能菜单与功能权限集中的元素相对应，根据不同的应用情况，功能菜单所具有的功能权限大小与权限范围也会有所不同. 通常情况下，参照相应的工作任务对功能菜单进行设置. 例如图 4 中，采购工作包含了发票校验、采购订单、采购计划这 3 个功能菜单.

功能菜单组合又与角色相对应. 在功能菜单-角色指派过程中，必须遵循安全级别限制原则、互斥权限原则、最小特权等分配原则. 根据粒度控制与业务的要求，能够把一个功能菜单分配给一个角色，也能够把多个功能菜单的组合指派给一个角色. 例如图 4 中，将 3 个功能菜单组合指派给了采购角色.

指派函数表示为

$$FUNC \rightarrow MENU$$

$$F(MENU, CONSTRAINTS) \rightarrow Roles$$

与传统 RBAC 模型相比，本文将角色定义为一组权限的集合，淡化角色的组织概念，按照粒度优化原则，从操作客体与数据对象的粒度层面来分析，角色的界定通过细化的功能权限集与功能菜单集来实现，进而实现组织岗位与角色的相互指派. 在传统的访问控制模型中，角色直接与岗位相对等，容易与组织机构产生冲突，而本文提出的模型正可以解决该问题，一旦岗位职能改变，只需为其指派其他细化的角色，这样既可以统一管理，又方便操作，更能适应现有的需求.

基于上述模型和方法，在作者参与开发的沈阳市和平区协同办公系统中，应用了访问控制细粒度元模型，建立了信息资源与操作客体的元数据库，进行了数据对象与操作客体的归并，实现了细粒度的访问控制，并取得了良好的应用效果.

4 结 语

针对基于角色的访问控制模型，从权限的数

据对象和操作客体两个维度建立了细粒度元模型，根据一定的规则进行粒度粗细的归并控制后，两者相互作用组合成功能权限集，由此构成功能菜单的组成部分，为角色的界定和层次划分提供参照标准，根据实际需要的设置，可以达到数据元组级别或属性级别的细粒度访问控制，使得系统的功能和数据的控制更加灵活，便于角色的管理，一方面达到信息安全访问的保障，另一方面为用户提供多方位的细粒度访问途径. 本研究可以为实际应用系统开发的访问控制策略制定提供方法上的借鉴. 对于模型中的数据对象粒度归并规则，需要进一步研究优化的粒度计算模型，在现实应用中，如果数据对象与操作客体的粒度越细、数目越多，那么系统就会更加频繁地调用访问控制决策模块进行权限的判定，这就需要更快的响应速度，所以粒度归并的优化和权限的最优设置仍是下一步研究的重点.

参考文献：

[1] SANDHU R S, COYNE E J, FEINSTEIN H L, *et al*. Role-based access control models [J]. **IEEE Computer**, 1996, **29**(2):38-47

[2] OH S, BYUN C, PARK S. An organizational structure-based administration model for decentralized access control [J]. **Journal of Information Science and Engineering**, 2006, **22**(6):1465-1483

[3] FERBER J, GUTKNECHT O. A meta-model for the analysis and design of organizations in multi-agent systems [C] // **Proceedings of the 3rd International Conference on Multi Agent Systems**. Washington D C: IEEE Computer Society, 1998

[4] SCHOINAS I, FALSAFI B, LEBECK A R. **Fine-grain Access Control for Distributed Shared Memory** [M]. California: ACM, 1994

[5] 廖　尔. 细粒度角色访问控制模型及其应用[J]. 河北省科学院学报, 2009, **26**(3):55-58

[6] 司　炜，曾广周，盛　琦，等. RBAC模型的细粒度扩充及应用[J]. 计算机科学, 2006, **33**(4):277-279

[7] 朱佃波. Web信息系统中统一细粒度访问控制的研究[D]. 苏州：苏州大学, 2008

[8] 廖俊国，洪　帆，肖海军，等. 细粒度的基于角色的访问控制模型[J]. 计算机工程与应用, 2007, **43**(34):138-140

[9] 赵　遐，怀进鹏. 基于XML的多粒度访问控制系统[J]. 计算机工程与应用, 2002, **38**(21):155-159

[10] 黄宏斌，张维明，邓　苏，等. 面向语义信息共享的元数据模型的研究与实现[J]. 计算机科学, 2008, **35**(4):124-128

[11] 高国伟，王延章. 政务信息资源目录体系中的政府组织元模型研究[J]. 情报杂志, 2008(6):151-154

[12] 韩志扬. 企业计算环境下细粒度访问控制模型XFGAC研究[D]. 青岛：山东科技大学, 2006

Research on meta-model to realize fine-grained access control

WANG　Ning*,　CHAO　Ke,　LUO　Pin

(School of Management Science and Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: Aiming at users′ demanding for fine-grained access control to protected information resources, the term "permission" in OB4LAC is expanded to be more finely granular in the way of meta-model. Attributes of information resources, such as time period, security constraints and context situation, are taken into account to establish fine-grained meta-model of data object and operation object, of which merging rules are researched so that the set of function permissions and function menus can be fine-grained. As such, roles can be wisely identified and classified accordingly, which is helpful for managing roles and achieving goals of fine-grained access control, as well as providing methods for establishing optimal access control policy.

Key words: access control; fine-grain; data object; operation level; meta-model