面向对象数据库的隐授权安全机制研究

向 宏,鹿 琪,胡海波,桑 军,蔡 斌
XIANG Hong,LU Qi,HU Hai-bo,SANG Jun,CAI Bin

重庆大学 软件学院,重庆 400044
College of Software Engineering,Chongqing University,Chongqing 400044,China
E-mail:lara_luqi67@yahoo.com.cn

XIANG Hong,LU Qi,HU Hai-bo,et al.Research on implicit authorization mechanism of object-oriented database.Computer Engineering and Applications,2010,46(2):121-124.

Abstract: Due to the special principle of data structure and manipulate principle,OODB has a particular safe protection and authorization methods.This paper first introduces the feature of security mechanism of OODB,and then based on the discussion on subject hierarchy,object hierarchy and operation hierarchy to focus on the authorization check process and authorization mechanism of OODB.By implement an instance combined with other authorization methods to validate the improvement of flexibility of implicit authorization mechanism.

Key words: Object-Oriented Database(OODB);safe protection;authorization mechanism;implicit authorization;authorization check

摘 要:面向对象数据库(OODB)独特的数据组织与操纵原理,决定了其具有不同于传统关系数据库的安全保护和授权模式。介绍了OODB安全授权机制特点,通过对OODB安全授权模式的主体、客体及访问方式三个层次的讨论,重点对授权检验流程及这三个层次上的隐授权机制进行了研究,通过尝试在实例中结合其他几种授权方式的应用,表明了隐授权机制的有效性和灵活性。

关键词:面向对象数据库;安全保护;授权机制;隐授权;授权检验
DOI:10.3778/j.issn.1002-8331.2010.02.037 **文章编号**:1002-8331(2010)02-0121-04 **文献标识码**:A **中图分类号**:TP311.132

1 引言

随着计算机网络技术的日益发展,数据库系统面临随之而来的数据集中及多用户多服务存取的需求,数据库的安全问题也成为日益复杂和严峻的课题[1]。OODB将面向对象的概念融入到数据库系统中,在数据库系统的发展史上是一个意义重大的进步[2]。因而自面向对象数据库产生以来,其数据安全性就吸引了广大的学者广泛研究和关注。随着OODB在各个领域日益广泛的应用,对其安全授权机制的研究也具有重大的意义。

对OODB安全性的研究中,授权机制和访问控制一直都是受到十分关注的内容。1994年,Fernandez等人提出了类层次结构的授权策略的概念,并进一步将广泛适用于关系数据库的五个维度改进为三个维度[3]。随后,Bertino[4]等人在其提出的三维模型上加入了有效性时间限制,提出了暂时授权模型(Temporal Authorization),但其有效性和灵活性不高。Demurjian[5]等则尝试将用户-角色的安全性应用于OO模型。而Thomas等人基于对主体的结构层次划分来对多级安全OODB建模,并提出主体层次间消息传递机制,将人们的研究推向了新的高度[6]。Bertino[7]等人随后又将强弱授权结合起来,有效地提高了灵活性,从而掀起了新的研究热潮。

在比较和总结各描述模型优缺点的基础上,该文进一步补充和发展了OODB授权机制中的主体、客体及访问方式层次结构,将隐授权方向扩充到了主体和操作层次,着重对隐授权控制策略进行了研究。

2 OODB的安全机制特点

面向对象数据库独特的数据模型结构,决定了其安全机制的如下特点。

2.1 客体复杂性

不同于传统的RDB中的关系表,面向对象数据库的基本存取单位是对象。每个对象都有其唯一的标识符,在数据库中具有唯一性和独立性。此外,OO数据模型是由类层次结构组成,超类和子类、类与实例之间有着密切的关联,这都使得其安全机制中客体层次及其相互关系变得多而复杂。

2.2 操作多样性

关系模型中的关系仅由数据属性组成,而OO数据模型中的对象(或类)是由属性与方法共同封装而成的[8]。OODB的操作已经不仅仅局限于传统的增删改查,用户还可以根据实际需要来定义用户方法,通过执行方法来实现对DB的各种操作。因此其

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z445)。
作者简介:向宏(1962-),男,教授,主要研究方向:网络与信息安全;鹿琪(1984-),女,硕士研究生,主要研究方向:信息安全与数据库技术;胡海波(1977-),男,博士生,主要研究方向:信息安全;桑军(1968-),男,副教授,主要研究方向:网络与信息安全;蔡斌(1979-),男,博士生,主要研究方向为信息安全。
收稿日期:2008-07-24 **修回日期**:2008-09-27

安全机制不仅要考虑属性数据的权限还要考虑到方法的权限。

2.3 自身安全性

面向对象数据模型自身已经具备了一定的安全保护措施：由于 OO 模型中的封装特性，OODB 的状态和行为被封装在各自的对象中，用户通过显式的消息传送来执行指定的用户或系统方法，进行存取或调用[9]。面向对象的数据模型自身提倡的数据封装性和消息传输机制也已经通过 public/private 类型提供了对于属性和方法的执行限制和存取保护。

2.4 分级保护

面向对象数据库可提供几种不同的安全机制[10]，如：子图机制、视图机制、授权机制和对象模型机制等，以实现 OODB 在数据库级、子图级、视图级、类级以及对象级的安全保护。

虽然 OODB 的视图机制和模型机制等在一定程度上提供了很好的保护机制，但没有提供指定对象对指定用户的存取机制，不能提供合适的精度。OODB 的授权机制基于对客体和操作的复杂层次结构，将强弱授权和肯否定授权结合起来，灵活准确地完成了 OODB 的存取访问控制。

3 安全授权模式

面向对象数据库的授权控制模式可由一个三元组(S,O,Op)定义，访问主体 S、访问客体 O 及访问方式 Op，三个元素之间的关系如图 1 所示。若有(S,O,Op)=true(或 1)，意味着主体(通常为用户或控制组)S 拥有在客体(或客体集)O 上的操作权限 Op，否则无操作权。

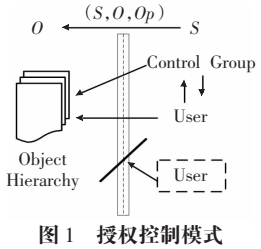

图 1　授权控制模式

鉴于上章提到的 OODB 安全授权机制的特性，对授权控制模式的各元素的层次结构进行如下的扩充和改进。

3.1 主体层次

根据用户 u 及其所拥有的角色，主体可以组织为不同的控制组 G_k(control Group)。访问主体可以是用户也可以是组。而组的成员可以是用户也可以是别的组[7]，这些元素共同组成一个不循环的有向图，访问主体的层次就体现在这些用户和控制组的层次关系上。一个控制组可由对相同客体对象有相同操作权限的若干用户或控制组组成，记作“u 直接属于 G_k”:$u\in G_k$，或者“u 间接属于 G_k”:$u\in_1 G_1\in_2\cdots\in_n G_k$，如图 2 所示。

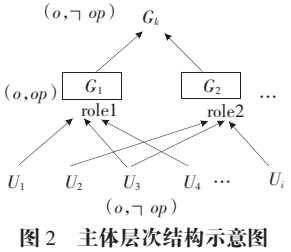

图 2　主体层次结构示意图

3.2 客体层次

对授权客体进行层次划分能够有效地简化 OODB 授权库，降低授权模型的复杂性。OODB 授权客体层次包含了所有的授权客体，并加入了类的方法(Method)进行了扩充[11]。

根据授权客体粒度的不同，可以将客体分为从数据库系统级到对象属性级几个不同的层次，这些层次共同组成一个带根的有向无环结构，如图 3 所示。层次较高的客体授权会隐含地包含层次较低的客体授权。

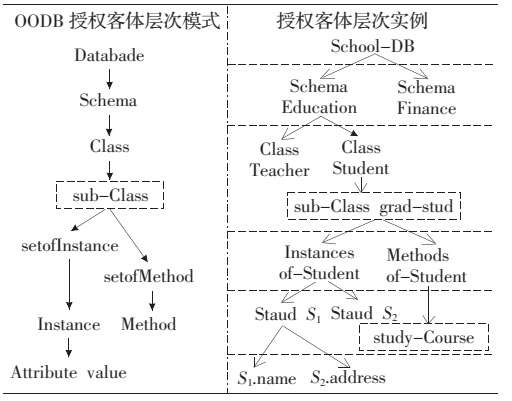

图 3　授权客体层次结构及层次实例示意图

3.3 访问方式(操作)

主体对客体的访问方式也涉及到操作权限的层次关联性。操作权限关联性使得主体在拥有某种操作时也隐含地拥有其他一些操作。在操作权限的关联关系中，写授权蕴含着读授权(可表达为 read<write)，定义新类型授权蕴含着对定义的读授权(可表达为 define<read define)。

OO 模型的类层次中又包含了方法的授权机制，方法一般具有调用、修改、建立三种操作[8]，其层次关系可表达为：call<modify<create。

4 授权控制机制

在层次化的面向对象数据库系统中，授权机制应该能足够灵活地支持复杂多变的访问需求[2]。它必须既能够简洁灵活地对某一类访问主体的权限进行合理化的统一控制，又要精确地定位不同的访问主体在某一特定客体层次上的操作特性。OODB 的授权机制利用肯定授权(Positive Authorization)和否定授权(Negative Authorization，用“¬”表示)来控制权限的授予或取消，而结合强授权(Strong Authorization)和弱授权(Weak Authorization，用“[]”表示)来满足特殊情况下的灵活性要求。

4.1 授权检验

当主体提出访问请求时，首先需要进行授权检验，根据授权检验的结果决定是否允许当前的访问操作[11]。

4.1.1 基本思想

假定主体 S 要对客体 O 进行访问操作 Op，就需要对主体 S 所拥有的各种权限进行检验，并根据检验结果决定允许或者拒绝当前请求的操作。根据 Bertino 等在文献[7]中的观点，一般授权检验过程的基本思想为由强授权到弱授权，由肯定到否定，完成对特定主体访问权限的限制检查，如图 4 所示。

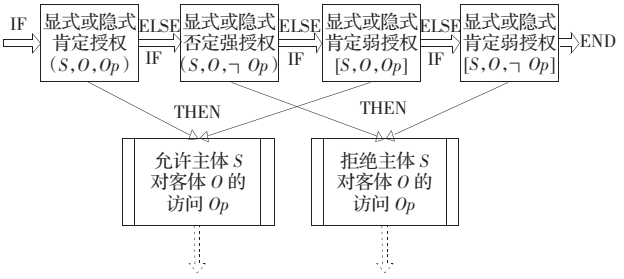

图 4　授权检验基本思想

4.1.2　检验流程

根据上述检验思想,基于主体的层次模型(图2),检验流程从特定用户开始,并根据检验的返回结果,决定结束检验过程,或是继续对上一主体层进行检验。设定用户 u 请求对客体 o 的访问操作 op,具体的检验流程 check_auth 步骤可简述如下,见图5。

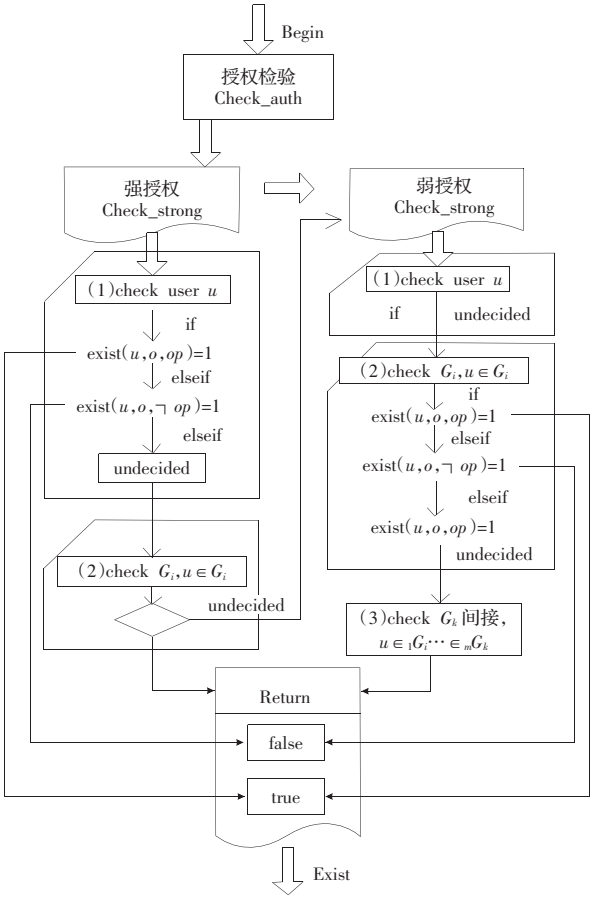

图5　授权检验流程示意图

(1)首先对用户 u 进行强授权检查:

```
begin check_strong(u):
  if(exist(u,o,op)=1)then
    return true;exit check_auth;
  elseif(exist(u,o,¬ op)=1)then
    return false;exit check_auth;
  else go to(2);
```

(2)对用户 u 直属的组 G_i 进行强授权检查

```
check_strong(G_i),u∈G_i:
  if(exist(G_i,o,op)=1)then
    return true;exit check_auth;
  elseif(exist(G_i,o,¬ op)=1)then
    return false;exit check_auth;
  else
    exit check_strong;
```

(3)随后对用户 u 进行弱授权检查:

```
begin check_weak(u):
  if(exist[u,o,op]=1)then
    return true;exit check_auth;
  elseif(exist[u,o,¬ op]=1)then
    return false;exit check_auth;
  else go to(4);
```

(4)对用户 u 直接属于的组 G_i 进行弱授权检查:check_weak(G_i),$u\in G_i$

```
  if exist([G_i,o,op]=1)then
    return true;exit check_auth;
  elseif exist([G_i,o,¬ op]=1)then
    return false;exit check_auth;
  else go to(5);
```

(5)逐层向上检查 u 间接所属的组 G_j,$u\in {}_1G_i\cdots\in {}_mG_j\cdots\in {}_nG_k$,($m<j<n$):

```
  While(exist(G_j,G_i∈G_j))
    redo check_weak(G_i);—步骤(4)
  exit check_weak;
  exit check_auth;结束授权检查。
```

4.2　隐授权

隐授权利用已有的授权和授权规则推导出新的授权,即新的授权隐含在已有的授权和授权规则中[2]。隐授权一定程度上提高了授权效率,降低了授权模型的复杂性。

根据隐授权的传递方向,可分为向上类(Upset)隐授权和向下类(Downset)隐授权两种。隐授权意味着授权的继承,以下分别就主体、客体和访问方式三个方面进行详细说明[8,11]。文中用符号"<"和">"表示客体层次和访问方式层次上的高低关系。

4.2.1　主体隐授权

从授权主体角度来看,隐授权与主体层次(Subject Hierarchy)有紧密的联系。

(1)主体层次上的 Upset 隐授权:若存在控制组 G_i 上的显式授权(G_i,O,Op),则对 G_k,隐授权(G_k,O,Op)成立,其中 $G_i\in {}_nG_k$,可理解为包含关系;若存在控制组 G_i 上的显式否定授权(G_i,O,¬ Op),则对 G_j,隐否定授权(G_j,O,¬ Op)成立,其中 $G_j\in {}_mG_i$。

(2)主体层次上的 Downset 隐授权:主体层次上的控制组可以实现对整个组内成员的隐授权。若存在控制组 G_k 上的显式授权(G_k,O,Op),则对 u_i,G_i,$u_i\in G_k$ 且 $G_i\in {}_iG_k$,有隐授权(u_i,O,Op)与(G_i,O,Op)成立。

4.2.2　访问方式隐授权

若存在 Op 上的显式肯定授权(S,O,Op),则有(S,O,$Op\_$)成立,其中 $Op\_<Op$($Op\_$ 表示位于 Op 下层的访问方式);若存在 Op 上的显式否定授权(S,O,¬ Op),则有隐授权(S,O,¬ Op^-),其中 Op^->Op($Op\_$ 表示位于 Op 下层的访问方式)。可见,在访问方式所在的层次上属于 Downset 隐授权。

4.2.3　客体隐授权

客体层次中某层节点的权限可以自动传递到下层,也可影响到上层客体。由于授权客体自身的灵活性和特殊性,隐式授权的方向与具体的访问方式与关[8]。

(1)客体层次上的 Upset 隐授权:若存在主体 S 在客体 O 上的显式肯定授权(S,O,Op),则有(S,O^-,Op),其中 O^->O(O^- 表示客体层次结构中位于 O 上层的客体);若存在客体 O 上的显式否定授权(S,O,¬ Op),则有(S,$O\_$,¬ Op),其中 $O\_$>O($O\_$ 表示客体层次结构中位于 O 下层的客体)。例如,若有(s,obj,$write$),且对象 obj 是 an instance of 类 C,则有隐授权(s,C,$read$)成立;此外,由(s,C,¬ $read$),也可得到(s,obj,¬ $read$)。

(2)客体层次上的 Downset 隐授权:若存在主体 S 在客体 O 上的显式肯定授权(S,O,Op),则有(S,$O\_$,Op),其中 $O\_<O$;若存在客体 O 上的显式否定授权(S,O,¬ Op),则有(S,O^-,

¬ *Op*),其中 $O^{-}>O$。例如,若有(*s*,*C*,*read*),且 *c* 是 a subclass of 类 *C*,则有(*s*,*c*,*read*)成立;若有(*s*,*obj*,*write*),且 a_i 是 an attribute of 对象 *obj*,m_i 是 a method of 对象 *obj*,则(*s*,a_i,*write*)和(*s*,m_i,*execute*)成立。另设 *mic_obj* 是 a part-of 复杂对象 *obj*,则有(*s*,*mic_obj*,*write*)成立。

表 1 主体、客体及访问方式层次上的隐授权传递

	显授权→	Upset 隐授权	Downset 隐授权
S	(G_i,o,op)	$(G_k,o,op),G_i\in{}_nG_k$	$(u_i,o,op),u_i\in G_i$
	$(G_i,o,\neg\ op)$	$(G_j,o,\neg\ op),G_j\in{}_mG_i$	$(G_j,o,op),G_j\in{}_mG_i$
O	(s,o,op)	$(s,o^{-},op),o^{-}>o$	$(s,o_{-},op),o_{-}<o$
	$(s,o,\neg\ op)$	$(s,o_{-},\neg\ op),o_{-}<o$	$(s,o^{-},\neg\ op),o^{-}>o$
Op	(s,o,op)	—	$(s,o,op_{-}),op_{-}<op$
	$(s,o,\neg\ op)$	—	$(s,o,\neg\ op^{-}),op^{-}>op$

4.3 冲突和重写

由于授权在各层次结构上的向下或向上隐传递,可能会造成同一主体在相同客体上的两个授权冲突。从授权主体的层次上进行冲突控制,通常有如下优先的原则:

(1)同一主体在相同客体上的同种操作,弱授权自动被强授权重写,隐授权自动被显授权覆盖。

(2)对两个弱授权之间的冲突,控制组 G_i 内成员的授权总是优先于对 G_i 的授权[7]。

(3)从直接属于的组 G_i 的传递的授权总是优先于从间接授权的组传递的授权。

(4)两个强授权之间有冲突,发生在后的强授权会视为无效。

综上,隐授权从某种程度上简化了面向对象数据库的授权机制,并在主体、访问方式和客体三个不同的角度上完成了授权的传递。但它在一定程度上缺乏灵活性,不能充分满足实际应用需要。而弱授权(Weak Authorization)方式具有可以被重写(overriding)的特点,因此可将其用于隐授权机制下的例外处理,还可由弱授权引出否定授权来共同定位授权失效的层次和节点,完成对例外情况的灵活控制。

5 实例验证

鉴于上述的授权机制的研究和总结,本章尝试加以实现和验证。如图 6 所示,定义类 Student 及其子类 grad_stud,并实现子类的两个实例对象:grad_stud1 和 grad_stud2。

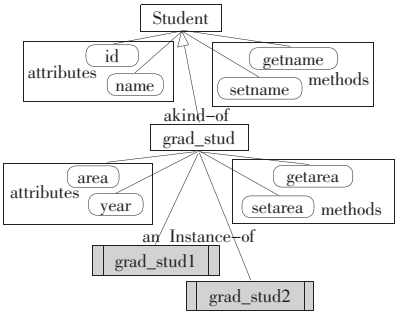

图 6 实例中的类层次结构图

基于图 2 中的主体层次进行授权机制的下述验证。

(1)假设存在对控制组 G_1 的显式强授权(G_1,*grad_student*,*update*):

GRANT update ON grad_student TO G_1;

由于 $U_1,U_3\in G_1$,则 U_3 可获得由 G_1 传递的隐式强授权(U_3,*grad_student*,*update*),拥有了对子类 *grad_student* 的属性、成员函数及实例对象的读写权。此外,由于它继承了 *Student*类,还会获得对父类属性 *id* 和 *name* 的读权限。

如果另外显式地对 U_3 进行否定授权(U_3,*grad_student*,¬ *read*):

NONGRANT read ON grad_student TO U_3;

根据组内成员授权优先,则 U_3 失去对类 *grad_student* 的可读权限。

(2)条件仍同(1),又若同时存在 G_1 上的显式肯定授权(G_1,*grad_student*,*update*)以及 G_k 上的显式否定授权(G_k,*grad_student*,¬ *update*)。由于 $U_1\in G_1$,且 $U_1\in{}_1G_1\in{}_2G_k$,根据直接从属的控制组授权优先,U_1 可获得隐授权(U_1,*grad_student*,*update*)。

(3)U_1 自动获得在子类 *grad_student* 上的 *update* 权限后,对该类的所有实例对象都获得读写权限。在现实情况中,通常某一类下面有大量的实例存在,对某一个类中所有实例的授权会显得过于笼统和粗糙。假定在某一特殊情况下,不希望 U_3 拥有对其中某一特定实例(如 *grad_stud*1)的读写权限,但是对剩余的其他所有实例(如 *grad_stud*2 等)都保留该权限[11],就需要在 *grad_student* 上对 U_1 使用弱授权操作[U_1,*grad_student*,*update*],并对 *grad_stud*2 使用否定读写授权操作(U_1,*grad_stud*2,*update*)来完成该灵活要求。

WEAKLY GRANT update ON grad_student TO U_1;

NONGRANT update ON grad_stud2 TO U_1;

Oracle 的对象体系遵从面向对象思想的基本特征。在 Oracle 的对象体系中定义图 6 的实例结构,并应用角色来控制图 2 中的控制组及其成员的权限,用命令 grant 与 revoke 实现授权的肯定和否定,对上述实例进行验证,可得表 2 的结果。

表 2 实例验证结果(表中斜体表示隐含授权)

S	Student.attr	grad_student	grad_stud1	grad_stud2
G_k	—	(¬ *update*)	¬ *update*	¬ *update*
G_1	*read*	(*update*)	*update*	*update*
U_3	—	(¬ *read*)	—	—
U_1	*read*	[*update*]	*update*	(¬ *update*)

可见,隐授权机制基于安全模式中的各个层次结构进行权限的传递和冲突的解决。此外,结合弱授权及否定授权可调整特定主体在客体层次上的隐授权[3],可增强授权机制的灵活性和可用性。

6 结语

OODB 中引入了面向对象的概念,使得其安全性保护机制更为复杂。授权机制提供合适的精度对指定用户进行授权访问,与其他几种安全机制共同确保了对 OODB 的安全保护。隐授权基于安全授权模式的主体层次、客体层次及访问方式层次完成对授权的传递,有效地简化了授权模型。此外,隐授权与其他几种授权方式结合,可以灵活有效地完成对 OODB 的授权保护。隐授权机制在简化授权模型复杂性的同时,也可能会带来同一主体在相同客体上的授权冲突,而在授权冲突的解决方面还有待更深入的研究和总结。

参考文献:

[1] Oki Y,Chikaraishi T,Shimomura T,et al.A design method for data integrity in object-oriented database systems[C]//Proceedings of IEEE International Conference on Information Engineering,Singapore,1995:204-209.

(下转 144 页)