面向对象数据库的安全性建模*

杨继国

(北京大学计算机系 北京 100871)

摘要 文章提出了面向对象数据库中对安全性进行建模的一种方法.该方法以多级数据模型为基础,定义了若干种安全性约束,同时引入了一种图示方法,能够对应用安全性模型进行直观地表示.

关键词 数据库,安全性,面向对象.

中图法分类号 TP311

随着网络化的发展,安全性已成为越来越多的应用系统的共同需求.在传统的关系数据库RDB(relational database)系统中,基于多级关系(Multilevel Relation)的强制存取控制机制已经得到了广泛的认可[1],在实践中也已应用到一些商用数据库中.面向对象数据库OODB(object-oriented database)系统正在从理论逐步走向应用.近年来,随着OODB研究的逐渐深入,安全性问题也成为OODB的重要研究课题.很自然地,多级数据模型被看成解决OODB安全性的一个重要途径,已经有相当多的文献和实验系统作出了将Bell-LaPadula模型扩展到OODB的尝试,一些商业OODB也正在或准备将它应用到其产品中.可以预计,支持多级数据模型将是实现OODB安全性的一个主流.

然而,随着OODB安全性研究的发展,随之而来的一些问题也逐渐产生,却很少有研究者注意到.其中之一就是OODB的安全性建模问题,即如何把现实世界中的安全性要求映射到数据库的安全性模式,在数据库支持的安全模型中表达实际系统的安全性语义.由于面向对象方法已经被公认为比传统方法更能自然地表达现实系统,在语义表达上比RDB更强有力,这也正是OODB的优势之所在.因此,解决好OODB的安全性建模问题是OODB走向实用的一个重要方面.本文在这个方面进行了一些探索,提出了一种OODB安全性建模的基本方法,并通过一个实例,对该方法作了进一步的说明.本文所提出的方法并不基于某个特定的OODBMS,但要求OODBMS能够支持多级数据模型.

1 安全性建模的基本框架

安全建模所要解决的基本问题是,如何将应用系统的安全性语义用数据库所支持的安全性模型表达出来,使得数据库模式能够符合现实世界的安全性要求.在这个过程中,应使设计者能够方便、直观地表达应用系统的安全语义.我们认为,基于多级数据模型的强制存取控制是OODB实现安全性的主要途径,因此假定所使用的OODBMS支持多级数据模型.由于在多级数据模型中,安全性语义是由附加在对象的属性和方法上的安全性约束(Constraint)来实现的,因此,我们的问题可以粗略地用图1表示.

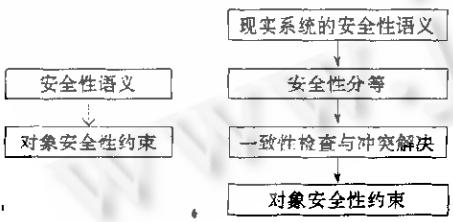

图1

图2

这个方法的本质就是在利用面向对象的建模技术,对应于现实世界的各种安全性,引入若干种安全性约束分类,进行安全性分等(Classifying),将现实系统中的安全性语义表达成数据库系统支持的安全性模型.需要注意的是,由于在表达和转换过程中可能会引起数据库安全性语义的不一致性,因此,进行一致性检测和发生冲突情况下的解决是必要的.整个框架如图2所示.

从图中我们看到,这种安全性建模首先是通过一系列的安全性分类约束,将现实世界的对象和它们的属性和方法赋予恰当的安全性标识(Label).一致性检查是指在上一步的基础上,对于各个对象及其属性、方法的安全性标识可能出现的不一致(冲突)进行检测,并通过一定的方法来解决这种冲突,从而维护数据库模式的一致性,最终形成完整的、一致的对象安全性约束,即对象安全性模式.可以看出,图的上半部分是

---

* 作者杨继国,1972年生,博士生,主要研究领域为数据库.
本文通讯联系人:杨继国,北京 100871,北京大学计算机系
本文1997-03-15收到原稿,1997-06-19收到修改稿

OODB提供方法，由应用系统设计者(建模者)来表达完成，而一致性检查和冲突解决应当主要由OODB系统来完成(当然不排除在某些方面需要用户的介入，但主要应是系统的工作).

在这里，我们主要研究的是图的上半部分，即通过各种安全性分类约束，将现实系统的安全性语义表达成对象安全性模型. 一致检查与冲突的解决实际上是比较复杂的，是一个需要进一步研究的问题. 目前这方面的研究还不多.

2 安全性分等

2.1 基本概念

在OODB中，最基本的概念是对象. 在这里，对象可以是现实世界中任何有意义的实体，每个对象有一个唯一的对象标识(OID)；具有相同的结构和行为的对象组成类，类中的每个对象为类的一个实例. 一个对象类可以定义成$O(A_1,\ldots,A_n,M_1,\ldots,M_m)$，其中$A_i(i=1,\ldots,n)$为对象的属性，定义在域$D_i$上；$M_j(j=1,\ldots,m)$为对象的方法，其返回值定义在域$D_j$上.

现在考虑引入安全性后，在多级数据模型中，对象类被扩展为多级对象类，可以表示为$O^m(A_1,C_1,\ldots,A_n,C_n,M_1,C_{n+1},\ldots,M_m,C_{n+m},CO)$，其中$C_i(i=1,\ldots,n)$为属性$A_i$对应的安全标识；$C_{n+j}(j=1,\ldots,m)$为方法$M_j$对应的安全标识；$CO$为整个类的安全标识. 在这里，安全标识事实上由两部分组成：一个定义在一定区间(通常为U(Unclassified)<Co(Confidential)<S(Secret)<TS(Top Secret))内的安全级别(SL)以及所对应的范畴(Category). 为了简单起见，我们只考虑前者(安全级别)，即假定只有一个范畴.

对象间的关系称为关联(Association)，关联按所参加的对象数目可以分成一对一的、一对多或多对多的等等. 在面向对象方法中还要考虑两个重要的结构概念：聚集(Aggregation)和特化/概化(Specialization/Generalization). 聚集是指一个对象可由其他对象组成，体现了part-of的语义；特化/概化是指子类可以继承父类的属性和方法，再增加自己的属性和方法，体现了is-a的语义，子类为父类的特化而父类为子类的概化. 这样，特化/概化定义了一个继承的层次.

2.2 约束类别

OODB安全性建模的关键问题是，将现实世界中的对象的安全性在多级数据模型中表示出来. 前面我们已经提到，多级数据模型是赋予对象的各个属性和方法以适当的安全标识(级别)，以达到强制存取控制的目的. 因此，主要问题也就归结到如何将现实对象O附加上适当的安全性约束，成为多级对象O^m，如图3所示. 这个过程被称之为安全性分等(Classifying). 需要注意的是，分等不仅仅是简单地赋予对应的属性和方法以一个安全标识，还要考虑到对象之间的关联以及上面提到的聚集和特化/概化关系. 另外，我们下面还会看到，不仅对象本身的属性和方法需要安全性分等，对于OODB的查询结果，也有安全性的问题.

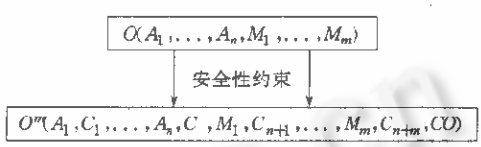

图3

我们首先明确一下分等的具体含义. 在这里，分等具有3个方面的含义：对属性分等、对方法分等和对对象实例分等. 对属性分等是指区分属性的值的安全级别；对方法分等并不是指对方法的返回值分等，而是指对方法本身的标识. 这就意味着，当一个具有一定安全级别的主体(例如用户或者用户所启动的事务)在访问对象时，他只能意识到那些安全级包含于(小于)主体安全级的方法的存在，从而才有调用(激活)该方法的可能. 一旦方法被激活，该方法便继承了主体的安全级，在随后的访问中就要根据这个安全级受到相应的控制，例如，按照Bell-LaPadula模型的不准向下写(No Write-down)和不准向上读(No Read-up). 显然，在这个含义下，我们不考虑对外界没有接口的方法，只有对和用户有接口的方法分等才有意义；对对象实例分等是指，对一个对象实例的属性和方法的安全级确定一个最小值和最大值.

将一个对象O转换成多级对象O^m的方法是，根据应用系统的安全性语义，将各种安全性约束附加于对象. 假定我们所关心的安全级集合为SL，下面我们给出所支持的约束种类.

定义1(简单约束). 假定K为对象类O的属性和/或方法的一个集合，即$K\subseteq\{A_1,\ldots,A_n,M_1,\ldots,M_m\}$. 定义简单约束$C_S$为以下安全分等形式：对于$O$的任一个属性或方法$E_i$，若$E_i\in K$，则$E_i$具有安全级别$C(C\in SL)$；若$E_i\notin K$，则不影响$E_i$的安全级别，记为$C_S(O(K))=C$.

简单约束C_S是一种简单的分等方式，它对一个对象中的若干属性和/或方法统一附加上一个相同的安全级别C，而保持其他属性和方法的安全级别不变.

定义 2(谓词约束). 假定 E_i 为对象类 O 的一个属性或方法,定义在域 D_i 上,$E_i \in K$(K 的定义同前);P 为定义在 E_i 上的一个谓词. 定义谓词约束 C_P 为以下安全分等形式:对于类 O 的任何实例 o,若 o 的一个属性或方法 $E_i \in K$,且对于谓词 P 结果为真,则 E_i 被赋予安全级别 C;否则($E_i \in K$ 或 P 不满足),E_i 的安全级别保持不变,记为 $C_P(O(K), P; E_i\ \theta\ a) = C$ 或 $C_P(O(K), P; E_i\ \theta\ E_j) = C(\theta \in \{=, \neq, <, >, \leqslant, \geqslant\}, a \in D_i, i \neq j, C \in SL)$. 这里,一个谓词还可以通过逻辑操作,与其他谓词进行组合.

谓词约束通过衡量满足定义在对象属性或方法上的一个谓词来给这些属性或方法赋予安全级别.

定义 3(复合约束). 假定 O, O' 为两个有关联或特化/概化关系的类,即 O 的任一实例 o 的存在都依赖于 O' 的一个实例 o' 的存在,$K \subseteq \{A_1, \ldots, A_n, M_1, \ldots, M_m\}$,$P(O')$ 为定义在 O' 上的一个谓词,其含义同谓词约束中一样. 定义复合约束 C_C 为以下安全分等形式:对于 O 的任何实例 o,若 o 的一个属性或方法 $E_i \in K$,若在与 o 相关的 O' 的实例 o' 中,$P(o')$ 为真,则被 E_i 赋予安全级别 C;否则($E_i \in K$ 或 $P(o')$ 不满足),E_i 的安全级别保持不变,记为 $C_C(O(K), P(O')) = C(C \in SL)$.

复合约束从形式上看很类似谓词约束,只不过在谓词约束中所衡量的谓词是作用在对象本身的,各个对象间是独立的;而在复合约束中,谓词是作用于与当前对象有依赖关系的其他对象上的.

定义 4(级别约束). 假定函数 $level(E_i)$ 为返回 E_i 所对应安全级别 c_i 的函数,$K \subseteq \{A_1, \ldots, A_n, M_1, \ldots, M_m\}$. 定义级别约束 C_L 为以下安全分等形式:对于 O 的任何实例 o,若 $E_i \in K$,则 E_i 被赋予安全级别 $level(E_i)$,即 $c_i = c_i$;否则,安全级别保持不变,记为 $C_L(O(K)) = level(E_i)$.

级别约束是通过对属性或方法赋予同一对象中的其他属性或方法的安全级别来进行分等的. 在现实应用中,很多情况下对象的所有实例中的一些属性或方法都具有相同的安全级别,级别约束在表达这种情况时非常有用.

以上的约束类别都是对对象本身的属性和方法进行安全性分等的,另一个需要安全性考虑的是查询的结果. 对查询结果作单独的安全性分等是必须的,因为在有些情况下,查询结果的安全性级别同查询中所涉及的对象的安全级是不同的. 虽然在目前的面向对象数据库系统中,导航式查询还是它所支持的主要查询方式,但由于多年来关系型数据库的结构化查询语言 SQL 在市场上所获得的成功,多数人相信 OODB 支持结构化查询只是时间早晚的问题,一些国际标准化组织也正在讨论 OSQL 的标准问题. 不管怎样,OODB 的查询应该是支持结构上的组合的,那么就可能会存在查询结果的安全性问题. 下面两个约束种类就是针对查询结果的安全性而提出的.

定义 5(聚集约束). 假定 $count(O)$ 为返回某个查询 Q 所返回的实例数的聚集函数,并属于对象 $O(A_1, \ldots, A_n, M_1, \ldots, M_m)$,$K \subseteq \{A_1, \ldots, A_n, M_1, \ldots, M_m\}$. 定义聚集约束 C_A 为以下安全分等形式:若访问属性集 K 的查询 Q 所返回的 O 的实例数超过 $n (n \in N)$,即 $count(O) > n$ 时,该查询的结果被赋予安全级别 $C (C \in SL)$,记为 $C_A(O, (K, count(O) > n)) = C(C \in SL, n \in N)$.

聚集约束区分返回的实例数目是否达到指定的数目来给查询结果分等. 在有些情况下,一个返回多个实例组合的查询结果与只返回由单个实例组成的查询结果安全性要求是不同的,这就是所谓的聚集问题. 聚集约束对聚集问题提供了一个直观的表达方法.

定义 6(推理约束). 假定 OI 为一个可能的逻辑推理 I 所涉及的所有对象的集合,$O_1, O_2, \ldots, O_n$ 为 OI 中的一些对象,它们具有多级数据表示 $O_i(A_{i1}, C_{i1}, \ldots, A_{im}, C_{im}, M_{i1}, C_{i(n+1)}, \ldots, M_{im}, C_{i(n+m)}, C_iO_i)$,$K_i \subseteq \{A_{i1}, \ldots, A_{im}, M_{i1}, \ldots, M_{im}\}$,$i, j \in \{1, \ldots, n\}$. 定义推理约束 C_I 为以下安全分等形式:对于任何包含全部 $K_i, \ldots, K_j$ 的查询,其查询结果被赋予安全级 C. 记为 $C_I(O_i(X_i), \ldots, O_j(X_j)) = C$. 特殊情况下,$i = j, O_i = O_j$.

推理约束用于模型中无法直接表达的推理关系的安全性要求,它能限制用低安全级的数据来推导出高安全级的数据. 在一些情况下,一些数据能够通过相当复杂的推理来得出安全性要求不同的其他数据,而这种关系在模型中又没有直接的方法表示,或者我们无法预知这个过程(事实上,在有些情况下,推理的过程还会加入数据库应用领域以外的其他知识),使用推理约束就可以表达这种安全性要求.

以上我们定义了 6 种安全分等方式. 其中前 4 种(简单约束、谓词约束、复合约束和级别约束)都是对单个对象进行安全性约束的. 这种情况下,一个对象类就可以由它的状态(属性)、行为(方法)和安全表示(约束)来决定. 后两种约束(聚集约束和推理约束)是针对查询结果的安全性的. 由于查询结果可以由多个对象的信息组合而得到,因此,用单个对象的约束有时候是无法表达某些安全性约束的. 在下节中,我们介绍一种图示方法,用于直观地表示这些约束种类,并通过一个实例演示安全性建模的方法.

3 图示方法

上一节我们引入了安全性建模的几种约束类别,用以表达应用系统中的安全性语义. 为直观起见,我们通过一种

图示方法来表示这些约束.由于安全性建模是建立在对象结构建模的基础之上的,所以,这个图示还有一些对象模型的表示.其基本的表示方法是借鉴 J. Rumbauh 等人的 OMT 方法中的图示(见文献[2]).说明如图 4 所示.

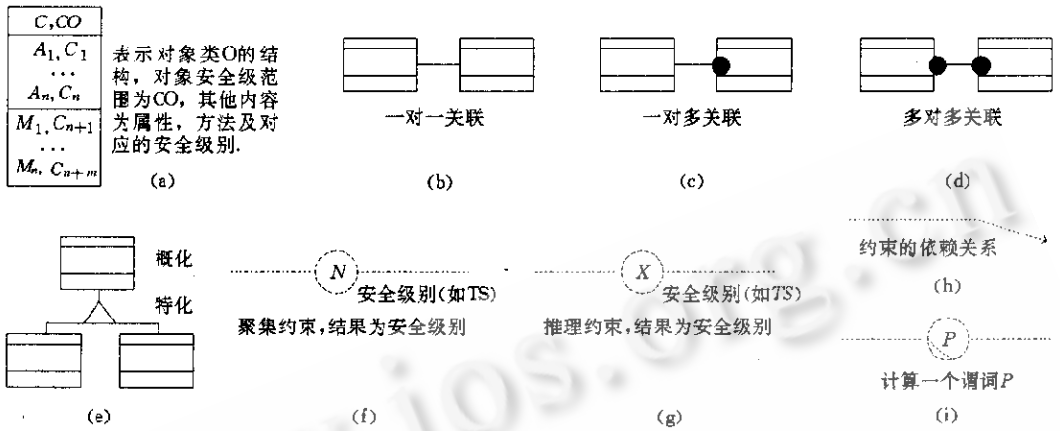

图4

在这些图例里,(a)～(e)为对象的结构表示.简单约束 C_S 由于形式简单,可以直接在对象结构(a)中通过标出方法或属性的安全级直接表示.(f)和(g)分别对应着聚集约束 C_A 和推理约束 C_I,圆圈后,线的下面标出的是查询结果的安全级别.其余的复合依赖 C_C、谓词依赖 C_P 和级别依赖 C_L 由(h)和(i)组合表示.

4　范　例

在这个简单的例子中,某公司有一批雇员,进行着若干项目,雇员被分配在各项目中工作.雇员的信息有员工号、姓名、部门、工资和专长,他们可以请求分配哪种工作;项目信息有名称、项目目的、客户和预算资金,还可以显示该项目的详细计划以及终止项目;雇员被分配到项目中工作时有分配日期和分工等信息.安全性要求雇员的工资不能为级别低于 S(即 secret)的人访问;项目只有级别为高度机密(TS)的人才可以终止,目的为"新产品研制"的项目的截止日期和预算为高度机密(TS),同时,这个项目的雇员分配情况也是 TS 级;对于所有的分配情况,安全级低于 Co(confidential)的人只能查询最多 3 条记录.最后,由于可能从雇员所在部门及项目的目的推断出雇员的分配情况,因此应防止安全级低于 Co 的人这样做.

由于安全性建模是建立在对象建模的基础上的,因此,我们首先进行对象结构的建模.由于它不是本文的重点,故这里只列出结果.关于 OO 方法建模见参考文献[2].这个例子比较简单,其结构如图 5 所示.其中有雇员和项目对象类,它们之间的多对多关联——分配联系也被作为一个对象类.

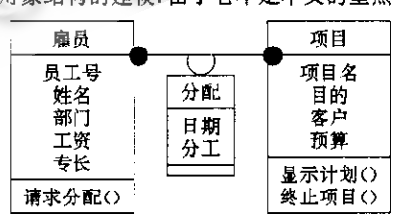

图5

在对象模型的基础上,我们考虑安全性约束.首先将所有对象类及其属性、方法的安全级置为缺省安全级 U(unclassified),再根据安全性语义,我们可以用以下安全性约束来表达:

(1) 在雇员对象类的工资属性上,有简单约束 C_S(雇员({工资}))=S;

(2) 在项目对象类的终止项目方法上,也存在简单约束 C_S(项目({终止项目()}))=TS;

(3) 谓词约束 C_P(项目({预算,显示计划()}),P:目的="新产品研制")=TS,限制了安全标识低于 TS 的主体访问目的为"新产品研制"的项目的预算及显示计划;

(4) 复合约束 C_C(分配({*}),项目(P:目的="新产品研制"))=TS,为目的为"新产品研制"的项目的分配对象赋予安全级 TS;

(5) 在分配对象上有一个聚集约束 C_A(分配,({*},*count*(分配)=3))=Co,使得安全级别低于 Co 的主体一次最多只能查询到 3 条分配记录;

(6) 推理约束 C_I(雇员({部门});项目({目的}))=Co,防止安全级低于 Co 的人用雇员和项目的信息来推论出分

配的信息.

图6是加上安全性约束后的结果,其中P为谓词(目的="新产品研制").

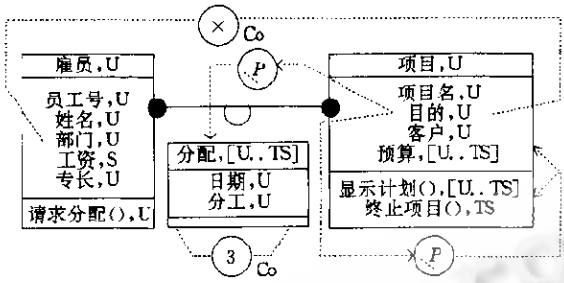

图6

5 总结与讨论

随着信息系统的网络化,安全性已成为许多应用系统的共同需求.很多系统采用诸如防火墙等技术来保护企业/单位内部的信息,但是防火墙只能防护来自外部的入侵,而对于防火墙内部的非法数据访问则无能为力.数据库系统所提供的安全性能够较好地从数据访问的低层控制对敏感信息的存取,应用前景非常广阔.然而,数据库系统的信息表示形式与现实世界存在一定的差异,对于应用系统的安全性要求,如何简单、直观地去表达是一个值得研究的课题.本文就这个问题作了一些初步的探索,提出了一种安全性建模的方法,并介绍了一种图形符号表示.虽然这个方法是针对面向对象数据库的,但对于关系型数据库也适用.这是因为,这种方法的本质是利用面向对象的设计方法,而对最终的实现系统没有要求必须支持面向对象,正如我们完全可以用OO方法设计一个系统,而最终用非OO语言实现一样.不过,本文的基础是多级数据模型,无论是OODB,还是RDB,多级数据的支持都是必要的.

本文所提出的方法是用于设计阶段的.一个应用系统的安全性语义和其他语义一样,是包含在应用本身中的.在信息系统中,要体现、维护这些安全性语义,必须将它们转化成为软件系统中的表示.采用本文所提出的方法,可以很直观、方便地做到这一点.现实世界的安全要求被表示成OODB中特定的几种安全约束,用一种直观的图示方法表示出来.本文所提出的安全性约束可以在支持多级数据模型的DBMS中很方便地支持和实现,使得设计和维护应用所要求的安全性变得简单、方便;同时,设计和开发人员在设计和实现应用系统时,也可以根据这些约束和图示,以直观、有效的办法来体现应用的安全性语义.这样,对于安全性要求较高的应用系统,使用本文提出的方法可以有效地支持设计和开发工作,减轻设计和维护的负担,从而提高软件的生产效率.

由于OODB对于安全性的支持还没有一个统一的标准(其实OODB本身的规范就还没有形成),各数据库厂商对安全性的支持都各有一套,如何最终实现应用系统的安全性要求,目前的做法还各不相同.但本文的方法是应用于设计阶段的,因此,对于各种实现方法都是适用的.关于数据库安全性建模的研究,目前还不多见,这个领域还有许多问题尚待解决.例如本文前面所提到的一致性检查和冲突解决问题等.

参考文献

1 Bertino E, Jajodia S, Samarati P. Database security: research and practice. Information Systems, 1995,20(7):537~556
2 Rumbauh J *et al*. Object-oriented modeling and design. NJ: Prentice Hallm Englewood Cliffs, 1991

Security Modeling in Object-oriented Database

YANG Ji-guo

(*Department of Computer Science Beijing University Beijing* 100871)

Abstract In this paper, a method to model the security semantic in object-oriented database system is presented. The method is based on multilevel data model, and defines several kinds of security constraint. A kind of diagram to illustrate application's security model is also introduced in this paper.

Key words Database, security, object-oriented.