

基于对象代理模型的异构多数据库集成^{*}

Integration of Multiple Heterogeneous Databases Based on Object Deputy Model

彭智勇¹ 何炎祥^{1,2} 罗 慧² 赵明霞² 罗 义² 叶 娟²

(武汉大学软件工程国家重点实验室 武汉430072)¹ (武汉大学计算机学院 武汉430072)²

Abstract Object-oriented approach is thought as a good method to integrate multiple heterogeneous databases under distributed environment because the object-oriented data model is semantically rich and can represent various types of data. However, traditional object-oriented data model is not enough flexible so that it is still difficult to resolve various syntactic and semantic conflicts. In this paper, we propose an object deputy approach which uses object deputy model as the common data model. The object deputy model is the extension of the traditional object-oriented data model with the new concepts of deputy objects and deputy classes. It is not only semantically rich but also very flexible and can facilitate the creation of the federated schema.

Keywords WWW, Information integration, Object deputy model

1. 引言

随着计算机和互联网的普及,网上信息源呈爆炸式增长。电子商务、电子政务、虚拟企业等新的计算机应用都需要将分散在网上的多个不同信息源集成起来进行统一处理。特别是经济的全球化,使得政府、企业、银行等组织纷纷将其机构分散起来管理。各分支机构通常都有自己的信息资源,为了便于决策,有必要将分散的信息源集成起来进行分析。另外,各种组织之间的兼并和重组也带来信息集成的巨大需求。因此,有必要研制一个软件系统用于集成网上信息源,构造虚拟数据库,提供统一处理界面。这是一个由来已久非常重要而复杂的研究课题。十几年以来,人们尝试了各种方法,开发了许多系统,例如有 MRDSM^[1], Superviews^[2], Pegasus^[3], UniSQL/M^[4], InfoSleuth^[9], Infomaster^[10], TSIMMIS^[11]。但据我们调查所知,到目前为止还没有形成有影响力的产品,基本上仍处于理论研究阶段。

实现异构多信息源集成的关键在于找到一个合适的公共数据模型。该模型必须语义丰富,能够表示不同信息源的各种数据类型;足够柔软,易于解决信息源之间的差异和冲突。人们尝试各种数据模型去实现异构多数据库的集成。例如,MRDSM 使用关系数据模型(Relational Data Model),Superviews 使用函数数据模型(Functional Data Model),TSIMMIS 使用对象交换模型(Object Exchange Model),InfoSleuth 使用本体论模型(Ontologies),

Pegasus 使用面向对象数据模型(Object Oriented Data Model),UniSQL/M 使用对象关系模型(Object Relational Model)。人们普遍认为面向对象数据模型是目前较好的公共数据模型^[6,12]。因为它具有丰富的数据语义表示能力,并能够通过对象操作方法解决数据之间的差异和冲突。然而面向对象数据模型的柔软性较差,以致异构型信息源之间的融合还比较困难。这主要表现在视图机制难以实现,缺乏自下而上的抽象机制(由子类定义父类)。

我们将使用对象代理模型(Object Deputy Model)作为公共数据模型。对象代理模型^[7]是我们针对面向对象数据模型柔软性较差等问题提出的一个新的数据模型。关系数据模型非常柔软就在于它将数据表示成简单的关系表,关系表可以通过关系代数进行分割和重组,变换其表现形式以满足不同数据库应用的需要。面向对象数据模型将数据和操作封装成对象,对象本身难以分割和重组,因此其柔软性较差。我们将代理对象引入到面向对象数据模型,使得一个对象可以有多个代理对象,多个对象可以共有一个代理对象。代理对象除了继承对象的属性和方法,还具有自己独有的属性和方法。代理对象的属性和操作通过代理类来定义,代理类可以通过我们提出的对象代理代数的操作导出。也就是说对象虽然难以直接地分割和重组,但通过其代理对象可以间接地进行。所以,对象代理模型除了具有面向对象数据模型的丰富数据语义表示能力之外,还具有类似关系数据模型的柔软性,能够满足公共数据

^{*})该研究得到中国博士后科学基金,教育部博士点基金项目(编号:20010486029)资助。

模型的要求。

在这篇论文中,我们将介绍对象代理模型,讨论如何基于对象代理模型实现网上异构多数据库的集成。

2. 对象代理模型

对象代理模型是为了解决传统面向对象数据库模型柔性差等问题而提出的一个新的数据库模型。它同传统面向对象数据库模型一样,将客观实体描述成对象。对象包括属性和方法,具有相同属性和方法的对象用类来描述。下面是对象和类的定义。

定义1 每个对象有一个标识符、属性集和方法集,具有相同属性和方法的对象用类来表示。作为对象模式,类包括类名、外延和类型。外延是属于该类的对象集,类型是对象的属性和方法的定义。类C可以表示成:

$$C = \langle O, A, M \rangle$$

(1)O 是类C的外延, $o \in O$ 是类C的一个实例。

(2)A 是类C属性定义的集合。对于 $(T_a: a) \in A$,a 和 T_a 分别表示一个属性的名字和类型。对象o的属性a的值表示为 $o.a$ 。对于每个属性 $T_a: a$,有两个基本方法: $read(o, a)$ 用于读取 $o.a$; $write(o, a, v)$ 用于将 $o.a$ 替换成新值v,它们可表示成:

$$read(o, a) = \uparrow o.a$$

$$write(o, a, v) = \rightarrow o.a := v$$

这里, $=$, $\uparrow$ 和 $:=$ 分别表示操作调用、值返回和赋值。

(3)M 是类C方法定义的集合。对于 $(m: \{T_p: p\}) \in M$,m 和 $\{T_p: p\}$ 分别表示方法名和该方法的参数集,p 和 T_p 分别表示参数名和类型。方法应用可表示成

$$apply(o, m, \{p\})$$

因为应用各种各样,对象满足某个应用的要求,不一定能完全满足另外一个应用的要求。我们通过定义代理对象对其进行某种扩充和改变,新的应用可以使用代理对象来满足其需要。针对不同的应用,可以定义不同的代理对象。

代理对象继承对象的属性和方法,同时可以增加应用需要而源对象没有的属性和方法。代理对象既可以作为对象的视图,又可以作为对象的角色。对象的角色随时间的变化可以看成对象的移动。因此,代理对象可将这三个概念统一起来处理。

一个对象可以有多个代理对象,多个对象可以共享一个代理对象,也可以定义代理对象的代理对象。代理对象的模式通过代理类来描述。下面是代理对象和代理类的定义。

定义2 代理对象基于对象或其他代理对象来

定义,后者称为源对象。代理对象的模式表示为代理类,它包括名字、外延和类型。代理类可由源对象所属类导出,后者称为源类。假定 $C^s = \langle \{o^s\}, \{T_a^s: a^s\}, \{m^s: \{T_p^s: p^s\}\} \rangle$ 是一个源类,其代理类 C^d 可定义为:

$$C^d = \langle \{o^d \mid o^d \rightarrow (o^s \vee [\dots \times o^s \times \dots] \vee \{o^s\}) \rangle,$$

$$sp(o^s) \vee cp([\dots \times o^s \times \dots]) \vee gp(\{o^s\}) = true \rangle,$$

$$\{T_a^d: a^d\} \cup \{T_{a+}^d: a_{+}^d\},$$

$$\{m^d: \{T_p^d: p^d\}\} \cup \{m_{+}^d: \{T_{p+}^d: p_{+}^d\}\} \rangle$$

(1) $\{o^d \mid o^d \rightarrow (o^s \vee [\dots \times o^s \times \dots] \vee \{o^s\}) \rangle, sp(o^s) \vee cp([\dots \times o^s \times \dots]) \vee gp(\{o^s\}) = true \rangle$ 是 C^d 的外延, $o^d \rightarrow (o^s \vee [\dots \times o^s \times \dots] \vee \{o^s\})$ 表示 o^d 是单个对象 o^s 或多个对象组合 $[\dots \times o^s \times \dots]$ 或一个对象集合 $\{o^s\}$ 的代理对象。 sp 、 cp 或 gp 分别为选择、组合或集合谓词。

(2) $\{T_a^d: a^d\} \cup \{T_{a+}^d: a_{+}^d\}$ 是 C^d 属性定义的集合,其中

(a) $\{T_a^d: a^d\}$ 是继承 C^s 的 $\{T_a^s: a^s\}$ 的属性集,其继承通过以下的切换操作进行。

$$read(o^d, a^d) \Rightarrow \uparrow f_{T_a}^s \rightarrow_{T_a}^d (read(o^s, a^s))$$

$$write(o^d, a^d, v^d) \Rightarrow write(o^s, a^s, f_{T_a}^d \rightarrow_{T_a}^s (v^d))$$

(b) $\{T_{a+}^d: a_{+}^d\}$ 是 C^d 增加的属性集,其基本操作如下:

$$read(o^d, a_{+}^d) \Rightarrow \uparrow o^d.a_{+}^d$$

$$write(o^d, a_{+}^d, v_{+}^d) \Rightarrow o^d.a_{+}^d := v_{+}^d$$

(3) $\{m^d: \{T_p^d: p^d\}\} \cup \{m_{+}^d: \{T_{p+}^d: p_{+}^d\}\}$ 是 C^d 方法定义的集合,其中

(a) $\{m^d: \{T_p^d: p^d\}\}$ 是继承 C^s 的 $\{m^s: \{T_p^s: p^s\}\}$ 的方法集,其继承通过以下的切换操作进行。

$$apply(o^d, m^d, \{p^d\}) \Rightarrow \uparrow apply(o^s, m^s, \{f_{T_p}^d \rightarrow_{T_p}^s (p^d)\})$$

(b) $\{m_{+}^d: \{T_{p+}^d: p_{+}^d\}\}$ 是 C^d 增加的方法集,其基本操作如下

$$apply(o^d, m_{+}^d, \{p_{+}^d\})$$

在关系数据模型中,关系代数可以用于导出各种形式的关系表。同样在对象代理模型中,我们有对象代理代数用于代理类的导出,它包括以下六个操作:

1)选择操作 它根据选择谓词选取源类的对象,创建其代理对象作为导出代理类的实例。

2)扩充操作 它导出代理类,其实例为源类的实例的代理对象,它被扩充新的属性和方法。

3)屏蔽操作 它导出代理类,其实例为源类的实例的代理对象,它部分继承源类的属性和方法。

4)联合操作 它将多个源类的实例的代理对象联合起来,作为所导出单个代理类的实例。

5)组合操作 它根据组合谓词,将多个源类的实例组合成单个代理对象,作为所导出单个代理类

的实例。

6)集合操作 它根据集合谓词,将源类的实例分成不同集合,为每个集合创建代理对象,作为所导出代理类的实例。

使用以上对象代理代数的操作,我们可以导出各种类型的代理类,用于对象视图定义、多样角色描述、数据模式进化等等。以下叙述表明对象代理代数也能在数据集成过程中发挥重要的作用。

3. 对象代理集成

数据集成就是解决来自不同数据源的数据之间的差异,将其转换成统一形式进行处理。数据集成需要一个公共数据模型,该模型必须语义丰富,能够表示不同信息源的各种数据类型;足够柔软,易于解决信息源之间的差异和冲突。对象代理模型既具有面向对象数据模型的丰富语义表现能力,又具有类似关系数据模型的柔软性,非常适合作为公共数据模型用于数据集成。

基于对象代理模型的数据集成称作对象代理集成,它是一种间接的集成,即对象通过代理对象间接地改变其形式达到与其他对象的统一。异构型数据库之间存在各种差异,主要差异可分成以下几类,我们将介绍基于对象代理模型的解决方案。

(1)同名异名及类型差异

在异构型数据库之间,存在同名或异名问题,即不同属性和方法采用相同的名字或者相同属性和方法采用不同的名字。另外,相同属性的值的类型可能不同。例如,日期属性有的采用字符串,有的采用数字表示。因为代理对象可通过切换操作继承对象的属性和方法,在继承过程中,切换操作能够改变属性和方法的名字和类型,所以名字和类型的差异可通过代理对象的切换操作来解决。

(2)同类对象的分布不同

同样类型的对象可能分布在不同的数据库中,在集成过程中,它们应归为一个类。如果它们所具有的属性和方法不完全相同,我们可以通过扩充操作对不足的进行补充,或者通过屏蔽操作对多余的进行隐蔽,然后再通过联合操作将它们归为一类。

(3)对象组成的分布不同

同一对象的组成可能分布在不同的数据库中,在集成过程中,它们应组合在一起。例如,一个人既在一个学校读书,又在一个公司工作。他的属性(名字、年龄、成绩、收入等等)分布在不同的数据库中,为了集成它们,可通过组合操作将它们合并成单个代理对象。其组合谓词可表示名字相同等条件。

4. 集成系统实现

实现基于对象代理模型的数据集成系统包括以

下五个方面:

(1)信息源描述

信息源可能是结构、半结构或无结构。它们所能提供的操作也是各种各样的。另外,有些数据可公开,有些数据不可公开。信息源描述包括它的数据模型、所能提供的数据和操作。我们将采用面向对象的方法描述信息源。

(2)包装器生成

包装器用来将信息源的数据模式转换成统一的数据模式。由于信息源各异,其包装器将不同。包装器生成将基于上述信息源描述来实现。只要用户给出信息源描述,系统就能自动或半自动地生成相应的包装器。

(3)合成器实现

合成器将基于对象代理模型来实现,它用来克服来自不同信息源的数据之间的差异和冲突,根据应用需要将数据集成起来。它可以通过虚拟视图或实体视图来实现。虚拟视图不需额外空间但查询效率不高,实体视图能高效查询但需额外空间且维护费用较高。对象代理模型既能提供虚拟视图又能实现实体视图,因此合成器能根据应用需要进行选择。

(4)集成化查询

一旦数据被集成起来,其查询就应分解或重写。所产生的子查询作用在信息源上,其结果汇总之后再返回给用户。集成化数据查询可以通过视图实体化机制和最优查询规划选择等方式提高其速度。

(5)集成化修改

修改集成化数据既要保证各个信息源内的数据一致性,又要保证信息源之间的数据一致性。数据集成系统将通过协调各个信息源的并发控制机制来保证数据修改的一致性。

总结 公共数据模型在异构多数据库集成中发挥着非常重要的作用。本文在分析传统面向对象数据模型所存在问题的基础上,提出了基于对象代理模型的数据库集成方法。其相关的实现技术有待进一步研究和开发。

参考文献

- 1 Wong K K, Bazex P. MRDSM: A relational multidatabase management system. In: Proc. Third Int. Seminar Distributed Data Sharing System, Mar. 1984
- 2 Motro A. Superviews: Virtual Integration of Multiple Databases. IEEE Trans. on Software Engineering, 1987, SE-13 (7): 785~798
- 3 Ahmed R, et al. The Pegasus heterogeneous multidatabase system. IEEE Computer, 1991, 24(12)
- 4 Kim W, et al. Classifying schematic and data heterogeneity in multi-database systems. IEEE Computer, Dec. 1991. 12~18
- 5 Pitoura E, Bukhres O, Elmagarmid A. Object Orientation in Multidatabase Systems. ACM Computing Surveys, 1995, 27(2): 141~195

(下转第255页)

查询的特点来决定。

本文用邻接链表表示 G_{join} 图,链表的每个节点需要记录边的 w 和 f 权值。用链表存储每个连接属性的 $profit$ 值,同时也用链表存储最后为每个关系分配的存储方式。关系存储方式选择算法描述如下:

输入:数据库模式 $D\{R_1, R_2, \dots, R_m\}$, 连接查询 $Q(q_1, q_2, \dots, q_n)$, 每个连接操作出现的概率 $(p_1, p_2, \dots, p_n)$

输出:关系的存储方式 $DS\{SR_1(A), \dots, SR_m(K)\}$

```
1 构造存储  $G_{join}$  的邻接表
2 while (E 中的边不为空) {
3   将 profit 链表置空
4   for(E 中每条边  $(vi, vj)$ )
5     if  $f(vi, vj)$  不在 profit 链表中
6       计算  $profit(f(vi, vj))$  的值, 并将其插入到 profit 链表中
7   找到 profit 值最大的属性  $A_k$ 
8   在邻接表中查找所有满足  $f(vi, vj) = A_k$  的  $vi$  点, 将  $vi$  点从邻接表中删除, 将其加入到存储分布方式的链表中, 令分布方式为 Hash 或 Range-Partition, 分布属性为  $A_k$ 
}
```

算法时间复杂性分析 设 $|V|=m, |E|=n$ 。算法的第1步需要时间复杂度为 $O(m+n)$, 算法的第3步需要遍历 profit 链表, 时间复杂度为 $O(n)$, 第4步需要循环 n 次, 第6步需要遍历邻接表, 复杂度为 $O(m+n)$; 第4~6步的复杂度为 $O(n(m+n))$; 第7步的复杂度为 $O(n)$, 第8步的复杂度为 $O(m+n)$; 最外层的 while 循环最坏要循环 n 次, 因此算法的时间复杂度为 $O(n^2(m+n))$ 。

结论 在基于群机系统的并行数据库中, 关系的存储方式对于数据库系统的查询效率有着很大的影响。连接操作是数据库系统中经常使用并且时间开销很大的一种数据操作, 在并行数据库中, 如果关系分布方式选择得合理, 在执行连接操作时就可以避免在处理机间交换数据, 大大提高连接的执行效率。本文提出了一个基于连接代价图的关系分布方式选择算法, 当给定一个数据库模式和一组连接查询时, 该算法可以为数据库中每个关系分配一种存储方式, 有效地提高这组连接查询的执行效率。

参考文献

- 1 李建中, 孙文骞. 并行关系数据库管理系统引论. 科学出版社, 1998. 57~71
- 2 Ghandeharizadh S, Dewitt D J. Performance Analysis of Alternative Declustering Strategies. In: Proc. of the Sixth Intl. Conf. on Data Engineering, 1990
- 3 Li Jianzhong, Srivastava J, Rotem D. CMD: A Multi-dimensional Declustering Method for Parallel Database Systems. In: Proc. of Intl. Conf. on Very Large Data Bases, 1992
- 4 Li Jianzhong, Srivastava J, Rotem D. CMD: A Multi-dimensional Declustering Method for Parallel Database Systems. [Technical Report, TR-91-59]. Department of Computer Science, University of Minnesota, 1991
- 5 Information Integration System. In: Proc. Of ACM SIGMOD Intl. Conf. 1997. 539~542
- 6 Peng Z, Kambayashi Y. Deputy Mechanisms for Object-Oriented Databases. In: Proc. of IEEE 11th Intl. Conf. on Data Engineering, 1995. 333~340
- 7 Bayardo R J, et al. Infosleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. In: Proc. of ACM SIGMOD Intl. Conf. 1997. 195~206
- 8 Genesereth M R, Keller A M, Duschka O M. Informaster: An
- 9 Li C, et al. Capability Based Mediation in TSIMMIS. In: Proc. of the ACM SIGMOD Conf. 1998. 564~566
- 10 Josifovski V, Risch T. Integrating Heterogeneous Overlapping Databases Through Object-Oriented Transformations. In: Proc. of the 25th VLDB Conf. Edinburgh, Scotland, 1999. 435~446

(上接第258页)