基于模式集成语义的查询处理*

石祥滨 张 斌 于 戈 郑怀远

(东北大学计算机系 沈阳 110006)

摘要 在采用面向对象模型作为公共数据模型的多数据库系统中,基于模式集成语义的查询处理不仅要实现针对集成模式查询到针对输出模式查询的转换,而且要从语义上尽可能减少回答用户查询所需数据,保证对象引用的正确性.为了达到这个目标,提出了一些新的概念及基于模式集成语义的查询处理规则和路径表达式的查询处理方法.

关键词 数据库,多数据库系统,查询处理,面向对象.

中图法分类号 TP311

多数据库系统中,异构性、自治性,查询处理是十分复杂的.现有的研究[1~4]把多数据库系统中的查询处理主要分为以下任务:(1) 查询修改,根据模式集成语义将针对集成模式的查询转换为针对输出模式的查询;(2) 查询计划生成,基于代数重写规则和代价模型,为经过查询修改后的查询产生优化的查询执行计划;(3) 局部查询处理,完成局部查询转换并取得局部查询结果.

事实上,查询修改不是简单地将针对集成模式的查询转换为针对输出模式的查询,在这个过程中,还要根据模式集成语义尽可能减少回答用户查询所需数据.从这个意义上讲,查询修改还具有一定的优化作用.因此,严格地讲,查询修改是一个基于模式集成语义的查询处理过程,本文将这个过程称为基于模式集成语义的查询处理.目前,关于这个问题的研究[1,3,4]只能处理一些简单的查询.随着面向对象数据模型作为公共数据模型的日益普遍,导航查询等对象特征的查询处理成为在异构分布环境下需要解决的重要问题.

SCOPE/CIMS是我们为满足CIMS环境下信息集成需求设计和实现的基于CORBA(common object request broker)的面向对象的多数据库信息集成平台系统,它采用了对象数据库管理组ODMG(object database management group)提出的对象数据库标准ODMG-93[5]规定的对象数据模型和对象查询语言OQL(object query language)作为公共数据模型和全局查询语言.为了在SCOPE/CIMS中重点解决导航查询的查询处理,我们发展了文献[6,7]中提出的语义标识符和限定路径表达式的概念,并定义了一些新的概念,包括:对象引用表、对象合并操作、基于语义标识符的集合操作、限定投影路径表达式等.基于这些概念,本文定义了模式集成操作,基于模式集成语义的基本查询处理规则和路径表达式的查询处理规则及查询处理方法,不仅保证了针对集成模式查询到针对输出模式查询转换的正确性,而且从语义上尽可能地减少了回答用户查询所需数据,保证了在路径表达式中对象引用的正确性.

1 基本概念

本节首先定义一些基本概念.本文所采用的符号约定如下:$A(C_i)$表示类C_i的属性集;$O(C_i)$表示类C_i的实例集或外延.

对于待集成模式中语义相关的类C_1和C_2,在集成模式中应定义它们的集成类C_i.为了回答集成用户对C_i的查询请求,多数据库系统的查询处理器应实现类C_1和C_2的数据集成.目前广泛采用的数据集成方法是外连接方法[1~3],但是外连接方法对于导航查询的处理能力是非常有限的.[6]因此,在文献[6]中采用了基于标识符的数据集成方法.这种方法在数据集成环境下具有更强的能力,并且可以简化数据集成.因此,我们根据SCOPE/CIMS中模式集成的特

---

* 本文研究得到国家863高科技项目基金资助.作者石祥滨,1963年生,博士生,副教授,主要研究领域为分布数据库,面向对象数据库,信息集成技术,软件工程.张斌,1964年生,博士,副教授,主要研究领域为信息集成技术,面向对象数据库,WEB数据库,软件工程.于戈,1962年生,博士,教授,博士导师,主要研究领域为并行处理技术,分布数据库,面向对象数据库,信息集成技术.郑怀远,1931年生,教授,博士导师,主要研究领域为分布数据库,面向对象数据库,信息集成技术.

本文通讯联系人:石祥滨,沈阳 110006,东北大学计算机系

本文1997-10-05收到原稿,1998-03-09收到修改稿

点,发展了这种想法,定义了以下概念.

定义 1. 语义标识符:表示为 soid,与对象标识符 oid 一样具有唯一标识一个对象的作用,soid 可以是关键字或对象状态的函数.引入语义标识符的目的是为了标识类 C_1 和类 C_2 的等价对象,因此,要求类 C_1 和类 C_2 中的等价对象的 soid 必须相同.很显然,soid 只有在类 C_1 和 C_2 中的实例集存在重叠时才是必要的,C_1,C_2 的实例集是否存在重叠不是通过比较 C_1,C_2 的实例集,而是通过语义确定的.

定义 2. 对象引用表:为了满足其它对象对类 C_1 和 C_2 的集成类 C_i 中对象的引用需要,我们定义了对象引用表.类 C_1 和 C_2 的对象引用表的每一行是一个三元组 $\langle source\_oid, soid, oid\rangle$,其中 $source\_oid$ 是原类 C_1 和 C_2 中对象的对象标识符(为简化讨论,假定来自不同局部场地的对象不存在对象标识符冲突),$soid$ 为类 C_1 和 C_2 中对象的语义标识符,oid 为集成类 C_i 中对象的对象标识符.有关对象引用表的作用可在第 4 节的例子中得到具体的说明.

定义 3. 对象合并操作:对类 C_1 和 C_2 的等价对象在集成类 C_i 中应合并为一个对象,这种合并在不同的查询处理上下文有不同的要求,为此,我们定义了不同的对象合并操作.如果 $o_1\in C_1, o_2\in C_2, o_1.soid=o_2.soid$,则 o_1 和 o_2 的合并按以下 3 个操作进行:

(1) 状态合并操作:定义为 $s(o_1,o_2)$.该操作只合并对象状态,不合并对象标识符,即对象标识符对查询结果不产生任何影响.合并后对象状态为 o_1,o_2 按一定的方法(如文献[1]中的聚集函数;文献[8]中的判定函数等)合并后的状态.对于对象状态的合并方法,本文不作过多讨论.

(2) 对象标识符合并操作:定义为 $i(o_1,o_2)$.该操作不合并对象状态,只合并对象标识符.合并后,对象状态选择 o_1 的状态,合并后,对象标识符 $oid=o_1.oid$ 或 $oid=o_2.oid$.该操作具有不可交换性.

(3) 状态标识符合并操作:定义为 $s_i(o_1,o_2)$.该操作既合并对象状态,也合并对象标识符.合并后,对象状态为 o_1,o_2 按一定的方法合并后的状态,合并后,对象标识符 $oid=o_1.oid$ 或 $oid=o_2.oid$.

在后两个操作中,$oid=o_1.oid$ 称为倾向于类 C_1 的对象标识符,$oid=o_2.oid$ 称为倾向于类 C_2 的对象标识符.

以上对象合并操作的定义主要基于以下的观点:①在回答用户查询时对象标识符不是必需的;②保证导航查询(对象引用)的正确性;③对象标识符合并操作比状态标识符合并操作所需数据量小,并且在特定情况下可简化导航查询(参见第 4 节).

定义 4. 基于语义标识符的集合操作:为实现基于不同语义的数据集成,我们根据前面定义的 3 种对象合并操作定义了基于语义标识符的集合并、交操作.基于语义标识符的集合并操作,表示为 $O(C_1)\bigcup_{soid,mop}O(C_2)$,$mop\in\{s,i,si\}$,具体定义为

$$O(C_1)\bigcup_{soid,s}O(C_2)=EO(C_1)\cup EO(C_2)\cup OO(C_1,C_2,s)$$

$$O(C_1)\bigcup_{soid,i}O(C_2)=EO(C_1)\cup OO(C_1,C_2,i)$$

$$O(C_1)\bigcup_{soid,si}O(C_2)=EO(C_1)\cup EO(C_2)\cup OO(C_1,C_2,si)$$

其中

$$EO(C_1)=\{o_1\mid o_1\in O(C_1)\wedge\neg\exists\ o_2(o_2\in O(C_2)\wedge o_1.soid=o_2.soid)\}$$

$$EO(C_2)=\{o_2\mid o_2\in O(C_2)\wedge\neg\exists\ o_1(o_1\in O(C_1)\wedge o_1.soid=o_2.soid)\}$$

$$OO(C_1,C_2,mop)=\{mop(o_1,o_2)\mid o_1\in O(C_1)\wedge\exists\ o_2(o_2\in O(C_2)\wedge o_1.soid=o_2.soid)\}$$

若类 C_1,C_2 的实例不存在重叠,则 $O(C_1)\bigcup_{soid,s}O(C_2)$ 和 $O(C_1)\bigcup_{soid,si}O(C_2)$ 简化为 $O(C_1)\cup O(C_2)$,$O(C_1)\bigcup_{soid,i}O(C_2)$,简化为 $O(C_1)$,此时有无语义标识符对操作结果无影响.

基于语义标识符的交操作定义为 $O(C_1)\bigcap_{soid,mop}O(C_2)=OO(C_1,C_2,mop)$,$mop\in\{s,i,si\}$.该操作不能简化为传统的集合交操作.

对于不能简化的基于语义标识符的集合并和交操作,通常需创建类 C_1 和类 C_2 的对象引用表,但不是必须的.这与查询的上下文有关(见第 4 节).其中 $EO(C_1)$ 部分的每个实例在对象引用表中建立一个如下形式的元组:$source\_oid=o_1.oid, soid=o_1.soid, oid=o_1.oid, o_1\in O(C_1)$;$EO(C_2)$ 的实例在对象引用表中建立的元组的形式类似;$OO(C_1,C_2,mop)$ 部分的实例在对象引用表中建立如下形式的两个元组:(1) $source\_oid=o_1.oid, soid=o_1.soid$;(2) $source\_oid=o_2.oid, soid=o_2.soid$,两个元组的 oid 相同,$oid=o_1.oid_1$ 或 $oid=o_2.oid_2, o_1\in O(C_1), o_2\in O(C_2), o_1.soid=o_2.soid$.

2 模式集成

本节我们首先将与本文相关的模式结构、模式一致化问题作简单的回顾(有关模式集成理论和方法的详细讨论见文献[9,10]),然后根据基于语义标识符的集合操作定义模式集成操作.

2.1 模式结构

在多数据库系统中,所采用的模式结构决定了查询转换的层次.在SCOPE/CIMS中,我们采用了文献[9]中提出了的5级模式结构,即局部模式、成员模式、输出模式、集成模式、外模式.集成模式与输出模式有直接对应关系的类称为基类,在集成过程中建立的新类称为虚类.本文所讨论的基于模式集成语义的查询处理的最基本的目标就是将针对集成模式的查询转换为针对输出模式的查询.

2.2 模式一致化

假定类C_1为某个待集成模式中的类,为了消除类C_1与其它模式中语义相关的类之间的存在的冲突(冲突及解决策略的详细讨论见文献[10,11]),需对类C_1进行同构处理,主要的同构操作可形式化地描述为以下两个函数.

$nf:I\times I\times\ldots\times I\to I$;$df:D_1\times D_2\times\ldots\times D_m\to D_{m+1}$,$m\geqslant 1$,其中$I$表示标识符的集合,$D_i$表示类$C_1$属性的域.

函数nf为性名操作函数,这个函数可以将一个属性的属性名改变为另外一个名字,将一组属性名对应为一个新的属性名.

函数df为性值转换函数,它由原属性的值或属性组的值导出原属性或新属性在类C_1中的新值.

2.3 模式集成操作

SCOPE/CIMS中所采用的全面的模式集成操作见文献[10],本文只讨论合并、概括、特化操作.

(1) 合并操作:C_1,C_2为语义上等价的类,C_3为合并后形成的新类,是一个虚类.如果在定义C_3时提供了C_1,C_2的语义标识符函数,则表明C_1,C_2的实例集存在重叠,否则,表明C_1,C_2的实例集不存在重叠.$A(C_3)=A(C_1)\cup A(C_2)$;$O(C_3)=O(C_1)\cup_{sid,mop}O(C_2)$.

(2) 概括操作:C_1,C_2为已存在的类,该操作建立C_1,C_2之间的共同超类C_3,C_3为一虚类.关于实例集是否重叠的约定同上.$A(C_3)=A(C_1)\cap A(C_2)$;$O(C_3)=O(C_1)\cup_{sid,mop}O(C_2)$.

(3) 特化操作:C_1,C_2为已存在的类,该操作建立C_1,C_2之间的共同子类C_3,C_3为一虚类.由于C_1,C_2的实例集必然存在重叠,因此,在定义C_3时必须提供C_1,C_2的语义标识符函数.$A(C_3)=A(C_1)\cup A(C_2)$;$O(C_3)=O(C_1)\cap_{sid,mop}O(C_2)$.

3 基于模式集成语义的基本查询处理规则

假设对一集成模式中类C_1,C_2通过前面介绍的集成操作形成的虚类C_i的一个合取查询Q_1为$\pi_{\exp(A_1)}\sigma_{pred(A_2)}O(C_i)$,$A_1\subseteq A(C_i)$,$A_2\subseteq A(C_i)$,$\exp(A_1)$为投影表达式,$pred(A_2)$为谓词表达式.$Q_1$中不存在对象导航问题.

规则1. 不重叠合并查询处理规则

如果$A_1\cap A(C_1)=\varnothing$或者$A_2\cap A(C_1)=\varnothing$,即投影表达式或谓词表达式与类$C_1$的属性无关,则对于$O(C_1)$的查询不会产生查询结果,将$O(C_i)$替换为$O(C_2)$.对于类$C_2$有类似的规则.

规则2. 不重叠合并查询处理规则

如果$A_1\cap A(C_1)\neq\varnothing$并且$A_2\cap A(C_1)\neq\varnothing$,但$\exists\ a(a\in A_2\wedge a\notin A(C_1))$,即谓词表达式中存在与类$C_1$的属性无关的谓词项,则对$O(C_1)$的查询也不会产生查询结果(合取查询的特点),同样将$O(C_i)$替换为$O(C_2)$.对类C_2有类似的规则.

规则3. 不重叠合并查询处理规则

在不能运用规则1,2的情况下,将$O(C_i)$替换为$O(C_1)\cup O(C_2)$.

规则4. 重叠合并查询处理规则

如果$A_1\cap A(C_1)=\varnothing$并且$A_2\cap A(C_1)=\varnothing$,尽管投影表达式和谓词表达式与类$C_1$的属性无关,但由于对象合并,不能简单地运用规则1(规则8的解释与此类似),将$O(C_i)$替换为$O(C_2)\cup_{sid,1}O(C_1)$.对于类C_2有类似的规则.

规则5. 重叠合并查询处理规则

在不能运用规则4的情况下,将$O(C_i)$替换为$O(C_1)\cup_{sid,\varnothing}O(C_2)$.

规则6. 不重叠概括查询处理规则

在任何情况下,将$O(C_i)$替换为$O(C_1)\cup O(C_2)$.

规则7. 重叠概括查询处理规则

在任何情况下,将$O(C_i)$替换为$O(C_1)\cup_{sid,\varnothing}O(C_2)$.

规则8. 特化查询处理规则

如果$A_1\cap A(C_1)=\varnothing$并且$A_2\cap A(C_1)=\varnothing$,则将$O(C_i)$替换为$O(C_2)\cap_{sid,1}O(C_1)$.对于类$C_2$有类似的规则.

规则 9. 特化查询处理规则

在不能运用规则 8 的情况下，将 $O(C_i)$ 替换为 $O(C_1)\cap_{sid,n}O(C_2)$.

规则 10. 基于语义标识符的集合操作的简化规则

如果查询 Q_1 作为一个独立的查询或查询 Q_1 作为另外一个查询 Q 的子查询时，Q 中不存在对 C_i 的引用，则不需要对象标识符的合并(对象标识符对查询结果无任何影响)，$O(C_1)\cup_{sid,n}O(C_2)$ 简化为 $O(C_1)\cup_{sid,n}O(C_2)$，$O(C_1)\cup_{sid,it}O(C_2)$ 简化为 $O(C_1)$.

4 基于模式集成语义的路径表达式的查询处理

在文献[7]中，为了有效地求解路径表达式，定义了限定路径表达式的概念.限定路径表达式比通常的路径表达式具有更强的表达能力，可以表达分支的路径表达式.为了在多数据库系统的查询处理中应用这个概念，我们根据下推投影技术对这个概念作了扩充.扩充后的路径表达式称为限定投影路径表达式，其形式如下

$$(x_1,O(C_1))[pj_1;p_1]\cdot(a_1,x_2)[pj_2;p_2]\cdot(a_2,x_3)[pj_3;p_3]\cdots(a_{n-1},x_n)[pj_n;p_n],$$

其中 x_i，$1\leqslant i\leqslant n$ 是一个变量，表示类 C_i(基类或虚类)的一个对象(称 x_i 的迭代范围为 $O(C_i)$)；a_i 是一个复杂属性(单值或多值)；p_i 是一个谓词，限定类 C_i 的对象，可以为空，pr_i 是对类 C_i 的对象的投影表达式，pr_i 为空时，投影表达式为 x_i；对于每个变量 x_i，有一个隐含的投影表达式项 a_{i+1}.限定投影路径表达式的语义是一个元组的集合$\{[pj_1,pj_2,\cdots,pj_n]\}$.

对于限定投影路径表达式的基于模式集成语义的查询处理，按以下几个阶段进行.

阶段 1. 迭代范围替换　替换阶段的任务是将针对集成模式的查询转换为针对输出模式的查询.具体作法是：对于每个变量 x_i 的迭代范围，运用前面的基本查询处理规则进行替换和简化其迭代范围.经替换处理后每个变量 x_i 的迭代范围为按如下规定的表达式 e_i.

(1) 一个基类的外延；

(2) 由基类的外延和集合并、交，基于语义标识符的集合并、交操作组成的表达式.

阶段 2. 分解　分解阶段的任务是将一个限定投影路径表达式运用下面的分解规则等价地分解为两个限定投影路径表达式的并.

规则 11. 分解规则

如果路径表达式 path 中变量 x_i 的迭代范围为 $e_{i1}\cup e_{i2}$，则可将路径表达式 path 转换为两个与 path 具有相同形式的路径表达式 $path_1$，$path_2$ 的并，$path_1$，$path_2$ 中变量 x_i 的迭代范围分别为 e_{i1} 和 e_{i2}.

阶段 3. 删除　删除阶段的任务是运用下面的删除规则，删除不能提供查询结果的限定投影路径表达式.

规则 12. 删除规则

若 x_i 的迭代范围为 e_i，x_{i+1}的迭代范围为 e_{i+1}，对于 e_i 中出现的每一个基类 C_{ij}，在 e_{i+1}中不存在与 C_{ij}来自同一输出模式并存在引用关系的基类，则该路径表达式无查询结果.

规则 13. 删除规则

如果 x_i 的迭代范围为 $e_{i1}\cup_{sid,it}e_{i2}$，$x_{i+1}$的迭代范围为 e_{i+1}，对于 e_{i1}中出现的每一个基类 C_{ij}，在 e_{i+1}中不存在与 C_{ij}来自同一输出模式并存在引用关系的基类，则该路径表达式无查询结果.

阶段 4. 对象引用表的确定　对象引用表的作用是为了保证导航查询的正确性，但对象引用表作为附加的存储并非在所有场合都是必需的，如果不需要对象引用表就能完成导航查询，则可简化查询，提高查询处理的效率.对象引用表的确定按以下规则进行.

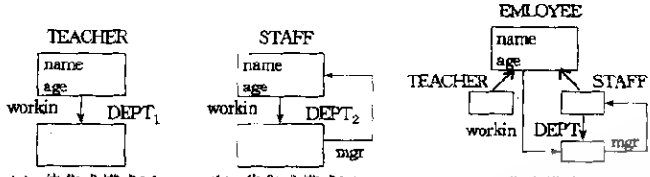
规则 14. 对象引用表的确定规则

如果 x_{i-1}的迭代范围为 e_{i-1}，$i\geqslant 2$，x_i 的迭代范围为不能简化的 $e_{i1}\cup_{sid,it}e_{i2}$，$e_{i1}\cap_{sid,it}e_{i2}$，$e_{i1}\cup_{sid,it}e_{i2}$或 $e_{i1}\cap_{sid,it}e_{i2}$对于 e_{i-1}中出现的每一个基类 $C_{i-1,j}$，在 e_{i2}中不存在与 $C_{i-1,j}$来自同一输出模式并存在引用关系的基类，则 x_{i-1}代表的对象对 x_i 代表的对象的引用不需要对象引用表，等价对象的合并选择倾向于 e_{i1}的对象标识符；否则，x_{i-1}代表的对象对 x_i 代表的对象的引用必须通过对象引用表，此时，等价对象的合并与选择倾向于 e_{i1}的对象标识符还是选择倾向于 e_{i2}的对象标识符无关.

阶段 5. 属性替换　前面处理后，如果变量 x_i 的迭代范围为一个基类的外延，直接将出现的属性替换为在模式一致化阶段定义的属性名操作函数和属性值转换函数；否则，在下推投影和谓词表达式后进行替换.

下面通过一个例子来说明前面的查询处理规则和查询处理方法.假设两个待集成模式如图 1(a)、1(b)所示，集成

模式如图1(c)所示. EMPLOYEE 由 TEACHER 和 STAFF 经概括操作形成,DEPT 由 $DEPT_1$ 和 $DEPT_2$ 经合并操作形成,限于篇幅,略去了可能存在的不必要的冲突.



(a) 待集成模式M_1　(b) 待集成模式M_2　(c) 集成模式M

图1

需要处理的限定投影路径表达式 path 为

$$(e,O(EMPLOYEE))[;e.age>20]\cdot(d,workin)\cdot(m,mgr)[m.name;]$$

下面我们在不同的假定下分析对这个路径表达式的查询处理.

假设 1. TEACHER 和 STAFF,$DEPT_1$ 和 $DEPT_2$ 的实例均不存在重叠. 按规则 6 可将 $O(EMPLOYEE)$ 替换为 $O(TEACHER)\cup O(STAFF)$,按规则 1 可将 $O(DEPT)$ 替换为 $O(DEPT_2)$,按规则 11 可将该路径表达式 path 分解为两个路径表达式 $path_1$,$path_2$,e 的迭代范围分别为 $O(TEACHER)$,$O(STAFF)$,但规则 12 可将 $path_1$ 删除.

假设 2. TEACHER 和 STAFF 的实例存在重叠,$DEPT_1$ 和 $DEPT_2$ 的实例不存在重叠. 按规则 7 和规则 10 可将 $O(EMPLOYEE)$ 替换 $O(TEACHER)\cup_{soid_1} O(STAFF)$,按规则 1 可将 $O(DEPT)$ 替换为 $O(DEPT_2)$. 从假设 2 我们可以看出这样的问题:(1) 在合并 TEACHER 和 STAFF 的等价对象的 workin 属性时,如果倾向于 TEACHER 的 workin 属性,则该路径表达式无查询结果,与属性合并(状态合并)相关的问题限于篇幅,本文不作讨论;(2) O(TEACHER)独有的实例无查询结果(无引用路径),即 $O(TEACHER)\cup_{soid_1} O(STAFF)$ 中可能存在部分不产生查询结果的数据. 但如果像文献[1]中那样作过细的实例划分,尽管可完全避免检索不产生查询结果的数据,可是会产生大量的子查询,在生成查询计划时将产生无效率的查询计划.

假设 3. TEACHER 和 STAFF 不存在重叠,$DEPT_1$ 和 $DEPT_2$ 的实例存在重叠. 按规则 6 可将 $O(EMPLOYEE)$ 替换为 $O(TEACHER)\cup O(STAFF)$,按规则 4 可将 $O(DEPT)$ 替换为 $O(DEPT_2)\cup_{soid_2} O(DEPT_1)$,按规则 11 可将该路径表达式分解为两个路径表达式 $path_1$,$path_2$,e 的迭代范围分别为 $O(TEACHER)$,$O(STAFF)$. 对于 $O(DEPT_2)\cup_{soid_2} O(DEPT_1)$,按规则 14,$DEPT_1$ 和 $DEPT_2$ 的重叠对象的对象标识符在 $path_1$ 中选择倾向于 $DEPT_1$ 的对象标识符,在 $path_2$ 中选择倾向于 $DEPT_2$ 的对象标识符,此时,很明显不需要建立对象引用表即可完成对象导航. 从这种情况我们可以看出这样的问题:在待集成模式 M_1 中,TEACHER 的实例无法引用 $DEPT_2$ 的 mgr 属性,但由于 $DEPT_1$ 和 $DEPT_2$ 的实例存在重叠和数据集成,这种引用成为可能,然而在 $path_1$ 中 TEACHER 独有的实例对 $DEPT_2$ 独有实例的引用不会产生查询结果(无引用路径).

假设 4. TEACHER 和 STAFF,$DEPT_1$ 和 $DEPT_2$ 的实例均存在重叠. 按规则 7 和规则 10,可将 $O(EMPLOYEE)$ 替换为 $O(TEACHER)\cup_{soid_1} O(STAFF)$,按规则 4 可将 $O(DEPT)$ 替换为 $O(DEPT_2)\cup_{soid_2} O(DEPT_1)$. 按规则 14,$O(DEPT_2)\cup_{soid_2} O(DEPT_1)$ 必须建立对象引用表. 从这种情况我们可以看出这样的问题:(1) 数据集成后,EMPLOYEE 的实例的 workin 属性既包括 $DEPT_1$ 对象的 oid,也可能包括 $DEPT_2$ 对象的 oid,如果不使用对象引用表,将无法通过 workin 属性的属性值链接 EMPLOYEE 的实例与所引用的 DEPT 的实例;(2) 对象引用表与等价对象合并后选择对象标识符的倾向性无关.

5 结束语

本文所讨论的基于模式集成语义的查询处理只是 SCOPE/CIMS 查询处理中需要解决的问题之一,其它问题,如,异构分布环境下导航查询的优化问题等是我们正在进行的研究工作.

参考文献

1 Dayal U. Query processing in a multidatabase system. In: Kim W, Reiner D S, Batory D S eds. Query Processing in Database System. New York: Spring-Verlag, 1985. 81～108

2 Chen A L P. Outjoin optimization in multidatabase system. In: Stanley Su ed. Proceedings of the 6th International Conference on Data Engineering. Los Alamitos, Califinia: IEEE CS Press, Feb. 1990. 211～218

3 Meng W, Yu C. Query processing in multidatabase system. In: Kim W ed. Modern Database Systems. New York: ACM Press, 1995. 551～572

4 Chen Arbee L P *et al*. Schema integration and query processing for multiple object databases. Integrated Computer-Aided Engineering, 1995,2(1):21～34

5 Object Database Management Group. The object dabase standard, ODMG-93, release 1.2, 1996

6 Papakonstantinou Yannis *et al*. Object fusion in mediator system. In: Vijayaraman T M *et al* eds. Proceedings of the 22nd International Conference on Very Large Database. India, Polo Alto, Califinia: Murgan Kaufmann, 1996. 413～424

7 Georges Gararin *et al*. Cost-based selection of path expression processing algorithms in object-oriented databases. In: Vijayaraman T M *et al* eds. Proceedings of the 22nd International Conference on Very Large Database. India, Polo Alto, Califinia: Murgan Kaufmann, 1996. 390～401

8 Vermeer Mark W W, Apers Peter M G. The role of integrity constraints in database interoperation. In: Vijayaraman T M *et al* eds. Proceedings of the 22nd International Conference on Very Large Database. India, Polo Alto, Califinia: Murgan Kaufmann, 1996. 425～435

9 Vaggelia. Object orientation in multidatabase systems. ACM Computing Surveys, 1995,27(2):141～195

10 张斌.基于面向对象的大规模多数据源集成技术的研究与实现[博士论文].东北大学,1996
(Zhang Bin. Study and pratice on integration theories of large scale multiple sources based on object-oriented [Ph.D. Dissertation]. Northeastern University, 1996)

11 张斌,石祥滨,郑怀远.面向对象的多数据库技术.计算机科学,1996,26(5):33～37
(Zhang Bin, Shi Xiang-bin, Zheng Huai-yuan. Object oriented multidatabase techniques. Computer Science, 1996,26(5):33～37)

Query Processing on Schema Integration Semantics

SHI Xiang-bin ZHANG Bin YU Ge ZHENG Huai-yuan

(*Department of Computer Science Northeastern University Shenyang* 110006)

Abstract In a multidatabase system with an object-oriented data model as a common data model, query processing on schema integration semantics should not only implement the translation from a query against integrated schema to queries against export schema, but also reduce the datum used for answering for user's query as more as possible and ensure the correctness of object reference in semantics. To achieve this object, in this paper, the authors present some new concepts, query processing rules and query processing methods for path expression on schema integration semantics.

Key words Database, multitatabase system, query processing, object-orientation.