基于带根连通有向图的对象集成模型及代数*

王 宁 徐宏炳 王能斌

(东南大学计算机科学系 南京 210096)

摘要 提出一种便于异构数据源集成的公共数据模型——OIM对象模型.它基于带根连通有向图,图中可出现环路,因而能自然地描述复杂对象与其成员对象间的引用关系和WWW上HTML文件间的链接关系.它的每个对象含有描述符,特别适合于描述那些没有显式模式或模式无法预知的数据对象.OIM对象代数提供对象并、差、选择、投影、粘贴及切削6种操作.此关系代数具有更大的灵活性,可作为查询分解和优化的形式化基础.

关键词 异构数据源,半结构化数据,数据集成,对象模型,对象代数.

中图法分类号 TP311

数据集成的研究始于70年代中期,以多库系统的研究[1,2]为主.然而,多库系统仅能集成数据库系统中的数据.随着计算机网络的普及和WWW(world-wide-web)的出现,人们要求数据集成系统不仅能集成数据库系统中的数据,也能集成非数据库系统的数据,例如,文件系统、电子邮件、电子表格、WWW上的HTML文件等.集成这样一些没有显式模式或模式无法预知的数据,是传统的数据集成技术难以胜任的.

有些多库系统(如CORDS[3],MIND[4])被称为具有可扩展性,允许新数据源的加入.这种可扩展性基于各数据源,经过包装具有标准界面,然而,从采用的公共模型看,它们的可扩展性是有限的.CORDS采用关系模型,任何加入CORDS的数据源在与其他数据源进行数据交互时,必须转换成表格式数据.显然,并非任何数据源的数据均可用表格式数据表示.MIND采用传统的O-O模型,O-O模型的描述能力虽然强于关系模型,但表示自描述数据仍显得力不从心.

要实现一个可扩展的异构数据源集成系统,除为各数据源定义标准界面外,尚需寻找一种通用的、便于异构数据源集成的公共数据模型.这种数据模型既要能够自然地描述各种数据源的数据,又要能够方便地表示来自各种异构数据源的异构数据,因而必须具有很强的描述能力.从这种意义上说,多库系统中常用的标准模型[5],如关系模型、E-R模型乃至传统的面向对象模型都显得不合适.因为在这些模型中,元数据和数据是分开存放的,不仅难于表示那些没有预知模式的数据源中的数据,在表示从各个异构数据源集成的数据时,也显得不够方便.

本文提出一种便于异构数据源集成的公共数据模型——基于带根连通有向图的对象集成模型(以下简称OIM对象模型),并以OIM对象代数作为查询语言的数学基础.OIM对象模型基于带根连通有向图,因而能自然地描述复杂对象与其成员对象间的引用关系和WWW上HTML文件间的链接关系.它的每个对象含有描述符,特别适合于描述那些没有显式模式或模式无法预知的数据对象.OIM对象代数的操作对象是OIM对象(一种带根连通有向图),它提供对象并、对象差、对象选择、对象投影、对象粘贴及对象切削6种操作.与关系代数[6]相比,OIM对象代数有更大的灵活性,它不仅能操作同构的OIM对象,也能操作异构的OIM对象.它无需遵循关系代数的"并兼容"原则,无论两个OIM对象是否相容,均可执行对象的并、差、粘贴运算,尤其适合异构多数据源的集成,这是John Grant等人提出的多关系代数[7]所无法比拟的.

本文引入带根连通有向图的有关概念,在此基础上,建立OIM对象模型,进而讨论OIM对象代数.

1 带根连通有向图

下述定义建立在有向图[8]的基础上.

* 本文研究得到国家自然科学基金资助.作者王宁,女,1967年生,博士,工程师,主要研究领域为数据库系统,分布对象技术.徐宏炳,1947年生,副教授,主要研究领域为数据库系统.王能斌,1929年生,教授,博导,主要研究领域为数据库,信息系统.

本文通讯联系人:王宁,南京210003,蔡家巷24号电力部电力自动化研究院系统所

本文1997-08-12收到原稿,1997-12-08收到修改稿

定义 1. 在一个有向图 $G(V,E)$中,如果存在一个入度为零的结点 r,r 到除自身外的任一结点都是可达的,则称 G 为带根连通有向图,r 为有向图 G 的根.

定理 1. 一个带根连通有向图 G 仅有一个根.

证明略.根据定理 1,一个带根连通有向图可记作 $G(r,V,E)$,其中 r 是根.

定义 2. 在带根连通有向图 $G(r,V,E)$中,如果 G 中存在一条边$\langle v_i,v_j\rangle$,则称 v_j 是 v_i 的亲子.对于某个结点 v_m,如果不存在亲子,则称 v_m 是叶结点,否则称为非叶结点.

定义 3. 在带根连通有向图 $G(r,V,E)$中,路径用结点的序列表示成 $p_i=(v_{i1}:v_{i2}:\ldots:v_{ik})$,$k\geqslant 2$,其中 v_{i1}称作路径 p_i 的始点,记作 $begin(p_i)$,v_{ik}称作路径 p_i 的终点,记作 $end(p_i)$.如果 $begin(p_i)=r$,则称 p_i 为原始路径.p_i 所经过的结点的集合$\{v_{i1},v_{i2},\ldots,v_{ik}\}$称作 p_i 的结点集,记作 $nodes(p_i)$.

定义 4. 带根连通有向图 $G(r,V,E)$中,如果存在两条路径 $p_i=(v_{i1}:v_{i2}:\ldots:v_{im})$,$p_j=(v_{i1}:v_{i2}:\ldots:v_{in})$,$n\leqslant m$,则称 p_j 是 p_i 的前端路径,p_i 的前端路径的集合记作 $front(p_i)$.

定义 5. 带根连通有向图 $G(r,V,E)$中,如果存在两条路径 $p_i=(v_{i1}:v_{i2}:\ldots:v_{im})$,$p_j=(v_{j1}:v_{j2}:\ldots:v_{jn})$,当 $v_{im}=v_{j1}$时,G 中一定存在路径$(v_{i1}:v_{i2}:\ldots:v_{im}:v_{j2}:\ldots:v_{jn})$,该路径称作 p_i 和 p_j 的连接,记作 p_ip_j.

定义 6. 如果带根连通有向图 $G(r,V,E)$中存在一条路径 p_i,p_i 的闭包是一个路径的集合,记作 p_i^+.$p_i^+=\{p_ip_j|p_j$ 是 G 中以 end(p_i)为始点的路径$\}$.

定义 7. 在一个带根连通有向图 $G(r,V,E)$中,对于任一 $v_k\in V$,从 v_k 出发的最大连通子图是一个有向图.记作 $MCG(G,v_k)$.$MCG(G,v_k)=(V',E')$,其中 $V'=\{v_i|v_i\in V\wedge v_k$ 到 v_i 可达$\}$,$E'=\{\langle v_i,v_j\rangle|v_i\in V'\wedge v_j\in V'\wedge\langle v_i,v_j\rangle\in E\}$.

下面,用 $v(MCG(G,v_k))$和 $e(MCG(G,v_k))$分别表示 $MCG(G,v_k)$的结点集和边集.

2　OIM 对象模型

在传统的数据模型中,元数据和数据是分开存放的,这种模型适合于具有一定模式结构的数据源,例如,数据库系统管理的数据.对于非数据库系统管理的数据,数据模式往往隐含于数据类型说明中,一般无显式的数据模式.为了统一地表示来自各种异构数据源的数据,OIM 对象模型的每个对象都含有描述符.

定义 8. 在 OIM 对象模型中,一个对象用四元组$\langle OID,n,t,c\rangle$表示,其中 OID 是对象标识符,n 表示对象名,t 表示对象类型,c 表示对象值,该四元组称为对象描述子.

为了描述来自各种数据源的数据,一个对象描述子的四元组$\langle OID,n,t,c\rangle$中,t 除了可表示基本数据类型(如 integer,char,float,string 等)外,还可表示集合数据类型(如 set,list,bag 等)、可变长数据类型(text,BLOB)和引用类型(ref).如果一个对象的类型是引用类型,表示该对象由其他对象聚集而成,它的值是所引用对象标识符的集合.

定义 9. 一个 OIM 对象 O 是一个带根连通有向图,表示成 $O(r,V,E)$.其中结点表示对象,边表示对象之间的引用关系.根结点 r 是一个聚集对象,它是引用类型的,V 是该聚集对象及其所有引用对象的集合,E 是对象之间引用关系的集合.即 $E=\{\langle v_i,v_j\rangle|v_i$ 是 V 中对象 O_i 的标识符$\wedge v_j$ 是 V 中对象 O_j 的标识符$\wedge$对象 O_i 引用对象 $O_j\}$.

定义 9 沿用了传统 O-O 模型的一些概念,然而,对于不同的 OIM 对象,对象之间的引用关系有着不同的语义.例如,关系数据库的一张表与表中的各个元组之间、一个元组与各个属性之间以及 HTML 文件间的超文本链均可看成对象之间的引用关系.

定义 10. 在 OIM 对象 $O(r,V,E)$中,从任一 $v_k\in V$ 出发的最大连通子图 $MCG(O,v_k)$都称作 O 的子对象.当 O 中的结点 v_n 是 v_m 的亲子时,$MCG(O,v_n)$称作 v_n 的亲子对象,v_m 的所有亲子对象的集合称作 v_m 的亲子对象集,记作 $son(v_m)$.根 r 的亲子对象集也称作 O 的亲子对象集.

定义 11. 设有 OIM 对象 $O_1(r_1,V_1,E_1)$和 $O_2(r_2,V_2,E_2)$,对于任一 $v_m\in V_1$ 和 $v_n\in V_2$,在 v_m 和 v_n 所表示对象的描述子中,如果对象名和对象类型分别相等,则称 v_m 和 v_n 是同类结点.

定义 12. 设有 OIM 对象 $O_1(r_1,V_1,E_1)$和 $O_2(r_2,V_2,E_2)$,$p_i=(v_{i1}:v_{i2}:\ldots:v_{i(m-1)}:v_{im})$是 O_1 中的一条路径,$p_j=(v_{j1}:v_{j2}:\ldots:v_{j(n-1)}:v_{jn})$是 O_2 中一条路径,如果 $m=n$,并且 v_{i2}与 v_{j2},v_{i3}与 v_{j3},$\ldots$,$v_{i(m-1)}$与 $v_{j(n-1)}$均是同类结点,则称 p_i 与 p_j 是同类路径.

定义 13. 设有 OIM 对象 $O_1(r_1,V_1,E_1)$和 $O_2(r_2,V_2,E_2)$.O_{1i}是 O_1 的从 $v_{1i}\in(V_1-\{r_1\})$出发的子对象,用有向图 $O_{1i}(V_{1i},E_{1i})$表示,O_{2j}是 O_2 的从 $v_{2j}\in(V_2-\{r_2\})$出发的子对象,用有向图 $O_{2j}(V_{2j},E_{2j})$表示.如果 V_{1i}与 V_{2j}之间存在一一对应的映射 Ψ,满足:

(1) 对于任意 $v_m \in V_{1i}$, v_m 与 $\Psi(v_m)$ 是同类结点;

(2) $(v_m, v_n) \in E_{1i}$ 当且仅当 $(\Psi(v_m), \Psi(v_n)) \in E_{2j}$,

则称 OIM 对象 O_1 的子对象 O_{1i} 与 OIM 对象 O_2 的子对象 O_{2j} 同构,否则,称 O_{1i} 与 O_{2j} 异构.

定义 11~13 并不排斥 O_1 与 O_2 是同一 OIM 对象的情形. 由定义 13,很容易推得以下定理.

定理 2. OIM 对象 O_1 的子对象 O_{1i} 与 OIM 对象 O_2 的子对象 O_{2j} 同构,O_2 的子对象 O_{2j} 与 O_3 的子对象 O_{3k} 同构,则 O_{1i} 与 O_{3k} 同构.

定义 14. 如果 OIM 对象 $O(r,V,E)$ 的所有亲子对象均同构,则称 O 是同构的 OIM 对象,否则,称 O 是异构的 OIM 对象.

定义 15. 设有两个同构的 OIM 对象 $O_1(r_1,V_1,E_1)$ 和 $O_2(r_2,V_2,E_2)$,如果 O_1 的任一亲子对象与 O_2 的亲子对象同构,则称 OIM 对象 O_1 与 O_2 相容.

例 1:设 WWW 上有一个结点 http://www.seu.edu.cn,若需检索的 HTML 文件结构简单表示成如图 1 所示的有向图,有向图的边表示超文本链,结点表示超文本文件. 用 OIM 对象模型可表示成图 2 所示的对象结构,它是一个异构的 OIM 对象.

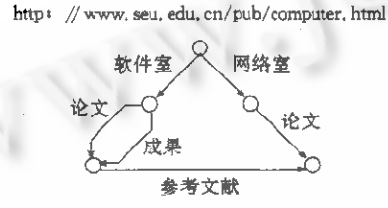

图1 HTML文件链结构

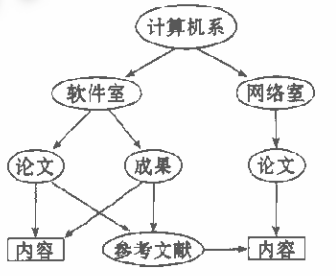

图2 超文本链的OIM对象结构

3 OIM 对象代数

OIM 对象模型从便于异构数据源的集成出发,提供一系列 OIM 对象操作的定义,它们分别是对象并、对象差、对象选择、对象投影、对象粘贴和对象切削操作,这些操作的总和称为 OIM 对象代数. OIM 对象代数满足封闭特性,即 OIM 对象运算的结果仍是 OIM 对象.

3.1 对象并

OIM 对象 O_1 和 O_2 的并记作 $O_1 \oplus O_2$. $O_1 \oplus O_2$ 的亲子对象集是 O_1 和 O_2 的亲子对象集的并.

设 $O_1=(r_1,V_1,E_1)$, $O_2=(r_2,V_2,E_2)$,则 $O_1 \oplus O_2=(r,V,E)$.

O_1 和 O_2 对象并的结果是一个新的 OIM 对象,它的根 r 由对象描述子〈OID,n,t,c〉表示. 其中 OID 由系统生成,对象名 n 可由用户指定或使用系统缺省值,对象类型 t 为引用类型. 由于每个引用类型对象的值实际上反映了该对象与被引用对象之间的引用关系,为便于维护,这种引用关系可以仅由 OIM 对象的引用关系集合 E 描述,引用类型对象的值由系统规定某个缺省值,并不具有实际意义. 为简化问题,本节 OIM 操作的定义均建立在这种前提下. 由于各操作结果 OIM 对象的根为引用类型,其定义可以类推,以后不再赘述,仅定义结果 OIM 对象的对象集 V 和引用关系集 E.

$$V=\{v_i \mid v_i \in (V_1-\{r_1\}) \vee v_i \in (V_2-\{r_2\}) \vee v_i=r\},$$

$$E=\{e_i \mid (e_i \in (E_1-\{\langle r_1,v_m\rangle\})) \vee e_i \in (E_2-\{\langle r_2,v_n\rangle\})) \vee e_i=\langle r,v_m\rangle \vee e_i=\langle r,v_n\rangle \wedge \langle r_1,v_m\rangle \in E_1 \wedge \langle r_2,v_n\rangle \in E_2\}$$

3.2 对象差

OIM 对象 O_1 和 O_2 的差记作 $O_1 \ominus O_2$. $O_1 \ominus O_2$ 的亲子对象集是 O_1 和 O_2 的亲子对象集的差.

设 $O_1=(r_1,V_1,E_1)$, $O_2=(r_2,V_2,E_2)$,则 $O_1 \ominus O_2=(r,V,E)$. 其中

$$V=\{v_i \mid (v_i \in v(MCG(O_1,v_m)) \vee v_i=r) \wedge \langle r_1,v_m\rangle \in E_1 \wedge \neg\exists\ v_n(\langle r_2,v_n\rangle \in E_2 \wedge MCG(O_1,v_m)=MCG(O_2,v_n))\},$$

$$E=\{e_i \mid (e_i \in e(MCG(O_1,v_m)) \vee e_i=\langle r,v_m\rangle) \wedge \langle r_1,v_m\rangle \in E_1 \wedge \neg\exists\ v_n(\langle r_2,v_n\rangle \in E_2 \wedge MCG(O_1,v_m)=MCG(O_2,v_n))\}.$$

对象并和对象差运算不同于关系代数的相应运算,参与运算的 OIM 对象可以不相容,对象的图表示中也可以出现环路. 这种灵活性给异构数据源的集成带来极大的方便.

3.3 对象选择

对象选择是按照一定的条件 f,在给定 OIM 对象 $O_1(r_1,V_1,E_1)$中选取根 r_1 的若干亲子对象.用公式表示为

$$\sigma[f](O_1)=(r,V,E)$$

这里,f 为布尔函数,表示选择条件.

$$V=\{v_i\mid(v_i\in v(MCG(O_1,v_m))\vee v_i=r)\wedge(r_1,v_m)\in E_1\wedge f(MCG(O_1,v_m))\}$$

$$E=\{e_i\mid(e_i\in e(MCG(O_1,v_m))\vee e_i=\langle r,v_m\rangle)\wedge(r_1,v_m)\in E_1\wedge f(MCG(O_1,v_m))\}$$

对象选择不同于关系代数的选择运算,它可以从一个异构的 OIM 对象中选出所有满足条件的亲子对象.

3.4 对象投影

对象投影运算是在给定 OIM 对象的所有亲子对象中,沿指定路径选取从路径终点出发的子对象.

给定 OIM 对象 $O_1(r_1,V_1,E_1)$以及一组原始路径表达式 $p_1,\ldots,p_k$.原始路径表达式由根对象名开始的一组对象名通过“.”连接而成.OIM 对象 O_1 在 $p_1,\ldots,p_k$ 上的投影可写做

$$\prod[p_1,\ldots,p_k](O_1)=(r,V,E)$$

$$V=\{v_i\mid(v_i\in(nodes(p_j)-\{r_1\}))\vee v_i\in v(MCG(O_1,end(p_j))\vee v_i=r)\wedge j\geqslant 1\wedge j\leqslant k\}$$

$$E=\{e_i\mid(e_i\in e(MCG(O_1,end(p_j)))\vee e_i\in(p_j-\{\langle r,v_m\rangle\})\vee e_i=\langle r,v_m\rangle)\wedge\langle r_1,v_m\rangle\in p_j\wedge j\geqslant 1\wedge j\leqslant k\}$$

与关系代数的投影操作不同,无论一个 OIM 对象是否同构,对象投影运算均可进行.

3.5 对象粘贴

对象粘贴是关系代数中外连接操作的扩展,它既可以处理同构的 OIM 对象,又可以处理异构的 OIM 对象,主要用于异构数据源之间对象的集成.

设有 OIM 对象 $O_1(r_1,V_1,E_1)$和 $O_2(r_2,V_2,E_2)$,p_1,p_2 是原始路径表达式,$end(p_1)$是 O_1 的非叶结点,表示 O_1 中的粘贴点,$end(p_2)$表示 O_2 的选取点,f 是粘贴条件.在 O_1 和 O_2 的亲子对象满足条件 f 的情况下,将 O_2 中 $end(p_2)$结点的所有亲子对象粘贴到 O_1 中,作为 $end(p_1)$结点的亲子对象的操作,用公式表示为

$$O_1\otimes[p_1,p_2,f]O_2=(r,V,E)$$

$V=\{v_i\mid(v_i\in(V_1-\{r_1\}))\vee v_i\in v(O_2)\vee v_i=r)\wedge f(MCG(O_1,end(p_1)),MCG(O_2,end(p_2)))\wedge O_2\in son(end(p_2))\wedge\forall\ p_m(\neg\exists\ p_n(p_m$ 是 O_2 中的路径$\wedge p_n$ 是 O_1 中的路径$\wedge begin(p_m)=end(p_2)\wedge begin(p_n)=end(p_1)\wedge end(p_m)=v_i\wedge p_m$ 与 p_n 是同类路径$\}$

$E=\{e_i\mid(e_i\in(E_1-\{\langle r_1,v_m\rangle\}))\vee e_i=\langle r,v_m\rangle\vee e_i=\langle end(p_1),v_n\rangle\vee e_i\in e(MCG(O_2,v_n))\vee e_i=\langle v_k,v_l\rangle\vee e_i\in e(MCG(O_2,v_l)))\wedge f(MCG(O_1,end(p_1)),MCG(O_2,end(p_2)))\wedge(r_1,v_m)\in E_1\wedge v_n\in son(end(p_2))\wedge v_n\notin son(end(p_1))\wedge v_k\in v(MCG(O_1,end(p_1))\wedge v'_l\in v(MCG(O_2,end(p_2))\wedge\exists\ p_s(p_s$ 是 O_1 中的路径$\wedge begin(p_s)=end(p_1)\wedge end(p_s)=v_k)\wedge\exists\ p_t(p_t$ 是 O_2 中的路径$\wedge begin(p_t)=end(p_2)\wedge end(p_t)=v'_l)\wedge p_s$ 与 p_t 是同类路径$\wedge v_l\in son(v'_l)\wedge v_l\notin son(v_k)\}$

粘贴操作专为异构数据源的集成而设计,可以将一个 OIM 对象的某个子对象粘贴到另一 OIM 对象的指定点,使灵活的数据集成成为可能.另外,粘贴操作规定被粘贴对象(O_1)为基本对象,在基本对象的某点粘贴其他对象(O_2)的子对象.在粘贴过程中,某些同类结点粘贴在一起,为用户提供语义一致的数据.

3.6 对象切削

对象切削运算是在给定 OIM 对象的所有亲子对象中,沿指定路径去除从路径终点出发的子对象.

给定 OIM 对象 $O_1(r_1,V_1,E_1)$以及一组原始路径表达式 $p_1,\ldots,p_k$.O_1 沿 $p_1,\ldots,p_k$ 的切削操作,用公式表示为

$$\overline{\prod}[p_1,\ldots,p_k](O_1)=(r,V,E)$$

$V=\{v_i\mid(v_i\in(nodes(p)-\{r_1\})\vee v_i\in v(MCG(O_1,p))\vee v_i=r)\wedge\exists\ p(p$ 是原始路径$\wedge p\notin(p_j^+\cup front(p_j)))\wedge j\geqslant 1\wedge j\leqslant k\}$

$E=\{e_i\mid(e_i\in e(MCG(O_1,p))\vee e_i\in(p-\{\langle r_1,v_m\rangle\}))\vee e_i=\langle r,v_m\rangle)\wedge\exists\ p(p$ 是原始路径$\wedge p\notin(p_j^+\cup front(p_j)))\wedge\langle r_1,v_m\rangle\in p\wedge j\geqslant 1\wedge j\leqslant k\}$

对于没有预知模式的数据源,像 WWW 上 HTML 文件结构,一般人很难了解超文本链的详细结构.用户利用对象切削运算,通过在 OIM 对象中去除已知部分,得到未知部分,给异构数据源的集成带来极大的方便.

4 结束语

数据集成系统研究至今,大多局限于集成有模式结构的数据源.为了能集成没有显式模式或模式无法预知的数据

源的数据,本文提出一种便于异构数据源集成的公共数据模型——OIM 对象模型.OIM 对象模型基于带根连通有向图,因而能自然地描述复杂对象与其成员对象间的引用关系以及 WWW 上 HTML 文件间的链接关系.OIM 对象代数提供对象并、差、选择、投影、粘贴及切削 6 种操作.它的单目运算既能操作同构的 OIM 对象,也能操作异构的 OIM 对象.它的二目运算无需遵循关系代数的"并兼容"原则,无论两个 OIM 对象是否相容,均可执行.这些特性使灵活的数据集成成为可能.

OIM 对象模型及其代数已用于 Versatile 系统[9]的实现.Versatile 是一个基于 CORBA 的可扩展的分布式数据集成系统.在 IONA 公司的 Orbix 产品上,Versatile 目前正对微软公司的 SQL Server、面向对象数据库系统 Versant、文件系统和超文本数据(即 WWW 中的数据)进行包装和集成.由于 OIM 对象模型具有很强的描述能力,使 Versatile 系统的可扩展性得到保证.OIM 对象代数是 Versatile 查询语言 OIQL 的数学基础,主要用于集成系统的查询分解和优化.

参考文献

1 Bright M W *et al*. A taxonomy and current issues in multidatabase systems. IEEE Computer, 1992,25(3):50～59

2 Sheth Amit P. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 1990,22(8):182～236

3 Attaluri G K *et al*. The CORDS multidatabase project. IBM Systems Journal, 1995,34(1):39～62

4 Dodac A *et al*. METU interoperable database system. Sigmod Record, 1995,24(3):56～61

5 Saltor F *et al*. On canonical models for federated DBs. Sigmod Record, 1991,20(4):44～48

6 王能斌.数据库系统.北京:电子工业出版社,1995.24～27
(Wang Neng-bin. Database System. Beijing: Electronic Industry Publishing House, 1995. 24～27)

7 Grant John *et al*. Query language for relational multidatabases. The VLDB Journal, 1993,2(2):153～171

8 Tremblay J P, Manohar R. Discrete Mathematic Structures with Applications to Computer Science. New York: McGraw-Hill Book Company, 1975. 185～237

9 Wang N, Chen Y, Yu B Q *et al*. Versatile: a scaleable CORBA-based system for integrating distributed data. In: 清华大学计算机系编. Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems. 北京:万国学术出版社, 1997.1589～1593

A Data Model and Algebra for Object Integration Based on a Rooted Connected Directed Graph

WANG Ning XU Hong-bing WANG Neng-bin

(*Department of Computer Science Southeast University Nanjing* 210096)

Abstract A data model named OIM is proposed in this paper as the common data model for integration of heterogeneous data sources. Based on rooted connected directed graph both cyclic and acyclic, OIM can describe the relationship of a complex object and its component objects, as well as the links between HTML files in World-Wide-Web naturally. As the metadata is associated with each object, OIM is especially suitable for describing objects without explicit predictable data schemata. As the formal foundation for query decomposition and optimization, a new algebra called OIM algebra, which includes six operations, i.e., object union, difference, select, project, paste and cut, is proposed. In comparison with relational algebra, OIM algebra is more flexible and powerful.

Key words Heterogeneous data sources, semi-structured data, data integration, data model, object algebra.