

从关系数据库到面向对象语义数据库的 语义捕捉和模式转换*

王国仁 于戈 石晶 单吉第 郑怀远

(东北大学计算机系 沈阳 110006)

摘要 本文提出了一个从关系数据库到面向对象语义数据库的两段模式转换方法,该方法是
以语义确认为基础的.包含在局部关系模式中的语义可分为2类:一类是关系内语义,它是包含在
一个局部关系模式内的语义;另一类是关系间语义,它是包含在局部关系模式之间的语义.第1
阶段主要是将关系内语义捕捉到全局集成模式中来;而第2阶段则将关系间语义通过模式重构
操作添加到全局集成模式中来.本文定义了一套用于建立转换机制的映射规则和转换操作,设
计并实现了用于产生G关系的算法和一系列转换公式.

关键词 语义捕捉,模式集成,模式映射,查询翻译.

OSAM*是为复杂的工程应用而设计的一个面向对象的语义数据模型.^[1]就我们所知,
至少已经开发出了2个基于OSAM*模型的异构分布式数据库系统.一个是由美国Florida
大学开发的IMDAS系统^[2],一个是由清华大学和东北大学联合开发的CIMBASE系统.^[3]
在这2个系统中,OSAM*用来作为全局或公共数据模型,并且分别提出了2个基于
OSAM*模型的全局语言OQL^[4]和OSDL.^[5]OQL和OSDL以自上而下的方式操作存储在
局部数据库中的数据.但是由于不支持数据库集成机制而无法操作驻留在已经存在的局部
数据库中的数据.

为了实现数据库集成有很多问题需要研究,如系统结构、集成机制、查询处理与优化、事
务管理、并发控制、完整性和安全性机制等^[6],其中模式转换和模式集成是建立多数据库系
统的基础.

在一个模式中既包含了数据的结构信息又包含了数据的语义信息,因此一个模式转换
必须包括结构转换和语义转换.由于一个数据模型的结构是确定的,因此仅需通过直接映射
就可容易地完成结构转换.在另一方面,由于难以清楚地描述包含在模式中的语义而使得语
义转换非常困难.包含在局部关系模式中的语义可分为2类:①关系内语义,它是包含在一

* 本文研究得到国家 863 高科技项目基金资助.作者王国仁,1966年生,讲师,主要研究领域为分布式数据库系统及
多数据库集成系统.于戈,1962年生,副教授,主要研究领域为分布式数据库系统.石晶,女,1969年生,助教,主要研究领
域为分布式数据库系统.单吉第,1936年生,副教授,主要研究领域为数据库系统及体系结构系统.郑怀远,1931年生,教
授,博士生导师,主要研究领域为数据库系统.

本文通讯联系人:王国仁,沈阳 110006,东北大学计算机系

本文 1995-08-10 收到修改稿

一个局部关系模式内的语义,如属性、关键字等;② 关系间语义,它是包含在局部关系模式之间的语义,如外键依赖语义、概括语义等。为了进行正确的模式转换,必须能确认这 2 类语义并捕捉到全局模式中来。

本文提出了一个两段模式转换方法,它是基于语义确认的。在第 1 阶段,关系内语义被捕捉并转换到全局模式中来;在第 2 阶段,关系间语义通过一系列模式重构操作添加到全局模式中来。

本文第 1 节描述了被转换的模式和基础语义。第 2 节中提出了两段模式转换方法,定义了映射规则和转换操作。第 3 节讨论了转换语句,并通过一个转换实例说明了这些语句的使用。第 4 节提出了一个用于产生 G 关系的算法和一系列转换公式。第 5 节总结全文。

1 问题定义

这一节我们先对关系数据模型和 OSAM* 数据模型进行简要回顾,然后讨论要被转换的基础语义。

1.1 关系模式

关系数据模型用关系来表示实体和实体之间的联系。^[7]形式化地,一个关系模式可以表示为一个四元组 $\langle R, A, P, C \rangle$ 。其中 R 是一个关系集合,每个关系有唯一的名字; A 是一个属性集合,每个属性有唯一的名字、数据类型、取值范围和它所属的关系; P 是每个关系的主关键字的集合; C 是 R 中关系必须遵循的一个约束集合。

1.2 OSAM* 模式

OSAM* 数据模型用关联和方法来表示对象。^[1]一个对象的结构特征通过与其它对象之间的各种关联来定义,而一个对象的行为特征则通过用户定义的方法和规则来定义。形式化地,一个 OSAM* 模式可以定义为一个六元组 $\langle E, I, D, A, C, M \rangle$ 。其中 E 是一个实体对象类(简称为实体类)的集合,每个实体类有唯一的名字; I 是每个实体类的标识属性集合; D 是一个域对象类(简称为域类)的集合,域类构成每个实体类属性的可能的取值范围,即定义域; A 是一个类与其它类之间关联的集合,一个关联有唯一的名字,关联类型(聚合关联、概括关联、交互关联、组合关联)和被关联的组成对象类; C 是一个约束集合; M 是一个方法和规则集合,用来指明完整性和其它语义约束。

1.3 局部关系模式中的基础语义

从实体和联系的角度来看,包含在一个局部关系模式中的基础语义可分成以下 3 类:

- 概括语义:具有共同属性特征和语义约束特征的实体可被概括成为一个更一般的实体。假设 R 是表示概括实体的关系, R_1, R_2, \dots, R_n 为表示子实体的关系,则该概括层次包含了以下语义约束:

(1) 对于所有的 $1 \leq i \leq n$, R 和 R_i 之间具有基数约束 1:0 或是 1:1;

(2) R_1, R_2, \dots, R_n 之间必须具有如下集合约束之一:集合相等、集合相交、集合互斥;

(3) R 和 R_1, R_2, \dots, R_n 之间必须具有以下 2 种约束之一:完全约束和部分约束。完全

约束应满足条件: $R = \bigcup_{i=1}^n R_i$;部分约束应满足条件: $R \subseteq \bigcup_{i=1}^n R_i$ 。

- 交互语义:在 OSAM* 模型中可通过一个交互关联层次来描述和捕捉关系模型中实

体间的联系及其语义. 在一个交互关联层次中包含了以下语义约束:

(1) 在任意 2 个交互实体间具有基数约束 1:1, 1:n 或 n:m;

(2) 每个交互实体具有参照引用约束, 即参加交互的每个实体必须在相应的组成实体类中存在;

· 聚合语义: 通过聚合一个属性集合和相应的组成实体来定义另一个实体. 假设 R 是表示该聚合实体的关系, A 是 R 中的一个属性, R' 是表示组成实体中的一个关系, 则在聚合实体中具有以下语义约束:

(1) 如果 A 是 R 的一个主关键字, 则 A 具有一个主关键字约束;

(2) 如果 A 是另一个关系的一个主关键字, 则 A 具有外键依赖约束;

(3) R 和 R' 之间具有一个 1:1 或 1:n(集合约束)的基数依赖约束;

(4) 每个聚合实体具有参照引用约束, 即参加聚合的所有实体在相应的组成实体类中必须存在.

2 模式转换方法

为了有效而正确地执行模式转换, 我们设计了一个基于语义确认的两段模式转换方法.

阶段 1: 关系内语义捕捉

在这个阶段, 存在于局部关系模式内的关系内语义被捕捉并被转换到全局模式中来. 在阶段 1 中创建的全局模式被称为全局基模式(相应地, 在全局基模式中的对象类被称为基类), 全局基模式是根据局部模式而创建的. 在全局基模式中并不能包括局部关系模式中的所有基础语义, 即: $S(Local-Schema) \supset S(Base-Schema)$, 其中, S 表示相关语义的集合.

阶段 2: 语义添加与模式重构

在这个阶段中, 存在于局部关系模式之间的关系间语义被确认, 并通过相应的模式重构操作将其添加到全局模式中去. 在阶段 2 中创建的全局模式被称为导出全局模式(相应地, 在导出全局模式中的对象类被称为导出类), 导出全局模式是根据全局基模式或其它导出全局模式而建立的. 最后将全局基模式和导出全局模式联结到一个全局概念模式中. 全局概念模式包含了局部模式中的所有基础语义, 即: $S(Local-Schema) \equiv (S(Base-Schema) \cup S(Derived-Schema)) \equiv S(Global-Conceptual-Schema)$

为了实现 2 段模式转换方法, 本文采取了如下技术:

(1) 对于阶段 1, 本文定义了一系列模式转换规则, 以捕捉存在于局部关系模式内的关系内语义, 并自动地将其转换到全局模式中来;

(2) 对于阶段 2, 本文提出了一系列模式重构操作, 以将存在于局部关系模式之间的关系间语义添加到全局模式中来. 如果部分关系间语义仍然丢失的话, 可以定义相应的方法和规则, 并将其包含在丢失语义的相应对象类中. 由于方法和规则难以维护, 人们

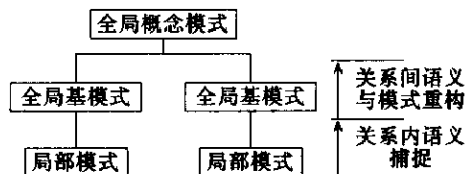


图1 两段模式转换方法

往往喜欢通过结构而不是通过过程或方法来捕捉语义. 两段模式转换方法可用图 1 来表示.

2.1 关系内语义转换规则

将存在于局部关系模式内的关系内语义转换到全局模式中来的语义转换规则如下:

- (1) 一个关系 R 被映射为一个实体类 E , R 和 E 的名字相同.
- (2) 关系 R 的一个属性 A 被映射为相应实体类 E 的一个聚合关联属性, 且它们具有相同的属性名.
- (3) 关系 R 的一个主关键字属性 $Key(R)$ 被映射为相应实体类 E 的一个标识属性.
- (4) 一个关系中某个属性的数据类型和范围被映射为一个基域类, 且基域类与该数据类型有相同的名字. 而且, 相同的数据类型被映射到相同的基域类上.
- (5) 一个参照约束被映射为从参照实体类到被参照实体类的一个聚合关联.

2.2 关系间语义捕捉与模式重构操作

为了将 1.3 小节中描述的关系间语义添加到一个全局数据模式中, 针对 3 种语义类型定义了一系列模式重构操作. 在给出转换定义之前, 先给出几个符号说明.

- (1) E : 表示一个实体类;
- (2) $Id(E)$: 表示实体类 E 的标识属性;
- (3) $G_subc(E)$: 表示实体类 E 的概括关联子类的集合;
- (4) $I_role(E)$: 表示实体类 E 的交互关联角色类的集合;
- (5) $A_attr(E)$: 表示实体类 E 的聚合关联属性的集合;
- (6) $A := B$: 表示集合 B 中的所有元素赋值给集合 A ;

下面, 我们详细给出 5 个模式重构操作的定义.

定义 1. GENERALIZE

给定 $n(n \geq 2)$ 实体类 E_1, E_2, \dots, E_n , 如果满足下列条件:

- (1) 在任意 2 个实体类之间不存在概括关联;
- (2) 它们有相同的关键字, 即: $Id(E_1) = Id(E_2) = \dots = Id(E_n)$;
- (3) 它们之间有一个相同的聚合关联属性子集 $\{Q_1, Q_2, \dots, Q_m\}$; 则在它们之上的一个 GENERALIZE 操作将建立一个新的概括层次, 在这个层次中一个新的实体类 E 作为超类, 且 E'_1, E'_2, \dots, E'_n 是它的子类, 并使得:

$$G_subc(E) := \{E'_1, E'_2, \dots, E'_n\}; A_attr(E) := \{Q_1, Q_2, \dots, Q_m\};$$

对于所有的 $1 \leq i \leq n, A_attr(E'_i) := A_attr(E_i) - \{Q_1, Q_2, \dots, Q_m\}$.

定义 2: MERGE

给定 $n(n \geq 2)$ 实体类 E_1, E_2, \dots, E_n , 如果他们具有相同的结构特征, 即: 对于所有的 $1 \leq i, j \leq n, i \neq j$, 且满足下列条件:

- (1) $Id(E_i) = Id(E_j)$;
- (2) $G_subc(E_i) = G_subc(E_j)$;
- (3) $I_role(E_i) = I_role(E_j)$;
- (4) $A_attr(E_i) = A_attr(E_j)$,

则在它们之上的一个 MERGE 操作将产生一个新的实体类 E , 且使得:

$$Id(E) := Id(E_1); \quad G_subc(E) := G_subc(E_1);$$

$$I_role(E) := I_role(E_1); \quad A_attr(E) := A_attr(E_1).$$

定义 3. GLINK

给定 2 个实体类: E_1 和 E_2 , 如果满足下列条件:

- (1) 在它们之间没有任何直接或间接的概括关联;
- (2) $G_subc(E_1) = \Phi$;
- (3) $Id(E_1) = Id(E_2)$;
- (4) $A_attr(E_1) \subseteq A_attr(E_2)$,

则在它们之上的一个 GLINK 操作将产生一个新的概括层次, 在这个层次中 E'_1 是超类, 而 E'_2 是子类. 且使得:

$$A_attr(E'_1) := A_attr(E_1); I_role(E'_1) := I_role(E_1); G_subc(E'_1) := \{E_2\};$$

$$A_attr(E'_2) := A_attr(E_2) - A_attr(E_1); I_role(E'_2) := I_role(E_2); G_subc(E'_2) := G_subc(E_2).$$

定义 4. INTERACT

给定 3 个实体类: E_1, E_2, E_3 , 如果满足下列条件:

- (1) 它们任意 2 个之间没有任何关联;
- (2) $Id(E_1) = Id(E_2)Id(E_3)$;
- (3) $Id(E_2) \cap Id(E_3) = \Phi$,

则在它们之上的一个 INTERACT 操作将创建一个交互关联, 在这个关联中 E_2 和 E_3 是 E_1 的 2 个角色实体类, 且使得: $I_role(E_1) := \{E_2, E_3\}$.

定义 5. AGGREGATE

给定一个实体类 E_1 , 且 $A_attr(E_1) = \{A_1, A_2, \dots, A_m, A_{m+1}, \dots, A_p\}$; 则在这个实体类之上的一个 AGGREGATE 操作可以创建一个新的实体类 E , 从 E_1 到 E 有一个聚合关联, 且使得:

$$A_attr(E) := \{A_{m+1}, \dots, A_p\};$$

$$G_subc(E'_1) := G_subc(E_1); I_role(E'_1) := I_role(E_1);$$

$$A_attr(E'_1) := \{A_1, A_2, \dots, A_m\};$$

3 转换语句

相应于转换的 2 个阶段, 分别提供了 2 组转换语句:

第 1 组: 为在第 1 阶段进行的模式翻译提供的一组说明语句. 主要语句的简要格式如下:

(1) DECLARE 语句

DECLARE <基域类名>: <系统定义类型> | <用户定义类型> | <数据构造器> of <元素数据类型>

这个语句用来说明与一个局部模式中数据类型相对应的基域类.

(2) DEFINE 语句

DEFINE <基域类名> A (<A 关联属性描述>) from <局部关系名> [composite-key <属性名表>]

这条语句用来定义与一个局部关系相对应的基类. 在定义过程中局部模式中的属性根

据需要可被换名或被过滤掉。

(3) FOREIGN—KEY 语句

FOREIGN—KEY of 〈参照实体类名〉 from 〈被参照实体类名〉(〈外键映射描述表〉);

这条语句是通过聚合关联来映射存在于局部关系模式中的部分外键依赖。

第 2 组:根据在第 2.2 节中定义的转换操作来定义用于语义添加的转换语句,主要语句的简要格式如下:

(1) GENERALIZE

GENERALIZE 〈实体类名〉 into 〈实体类名〉 constraints (〈集合约束描述〉);

这条语句用来捕捉存在于—组实体类之间的概括语义。

(2) MERGE

MERGE 〈实体类名表〉 into 〈实体类名〉

这条语句用来捕捉相同实体类之间的共同语义。

(3) GLINK

GLINK 〈子类名字〉 to 〈超类名字〉;

这条语句用来捕捉存在于实体类之间的概括语义。

(4) INTERACT

INTERACT (〈角色实体类描述表〉) into 〈实体类名〉 (〈角色描述表〉) constraints (〈映射约束描述〉);

这条语句用来表示存在于不同实体类之间的联系。

(5) AGGREGATE

AGGREGATE (〈A 关联属性名表〉) from 〈实体类名〉 into 〈实体类名〉

这条语句用来表示聚合关联语义。

另外,我们也定义了其它几个模式重构语句,如换名一个实体类或一个属性、删除一个类或一个属性。由于它们比较简单,这里不再赘述。

下面给出一个例子来说明如何将一个关系模式转换为一个 OSAM* 模式,关系模式给定如下,其中 * 表示一个关系的主关键字。

FACULTY (fac_no * ;INT, fac_name;CHAR(20), degree;INT(0—10), speciality;CHAR(20), fac_dept;INT);

DEPT (deptno * ;INT, name;CHAR(20), chairman;INT, college;CHAR(30));

TA (ta_no * ;INT, ta_name;CHAR(20), major;INT, gpa;CHAR(2), mode;CHAR(1), tac;INT, sex;CHAR);

RA (ra_no * ;INT, ra_name;CHAR(20), major;INT, gpa;CHAR(2), mode;CHAR(1), rac;INT, age;INT(26—60));

UNDERGRAD (ta_no * ;INT, ta_name;CHAR(20), major;INT, gpa;CHAR(2), minor;INT);

ADVISING1 (fac_no * ;INT, ta_no * ;INT, startdate;DATE);

ADVISING2 (fac_no * ;INT, ra_no * ;INT, startdate;DATE);

关系模式中有以下外关键字依赖:

foreign-key of dept from faculty (fac-dept,deptno);

foreign-key of dept from TA (major,deptno);

foreign-key of dept from RA (major,deptno);

foreign-key of faculty from dept (chairman,ss);

在阶段1,关系模式被转换成如图2所示的基本OSAM*模式,在这个基本OSAM*模式中有些属性通过 DEFINE 语句已被换名. 在阶段2,下面的转换语句被作用在基本OSAM*模式之上,其结果全局模式如图3所示.

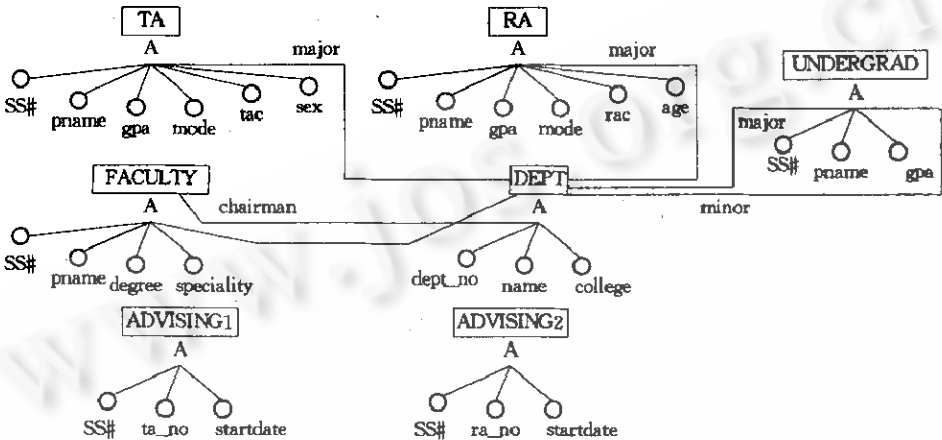


图2 经过阶段1后产生的基本OSAM*模式语义图

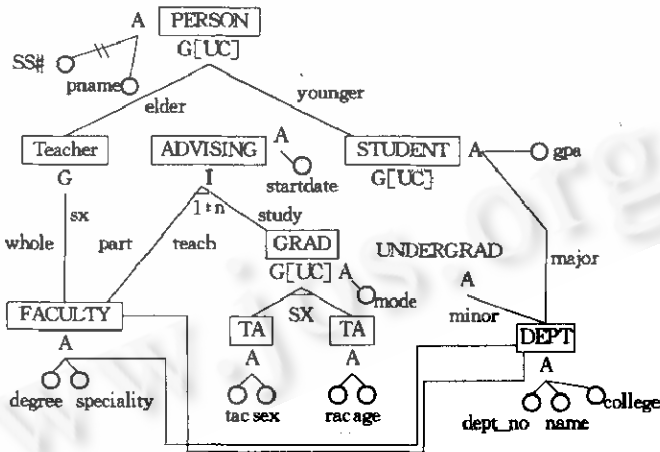


图3 经过阶段2后产生的OSAM*模式语义图

generalize(TA,RA) into GRAD constraints(all sex);

generalize(GRAD,UNDERGRAD) into STUDENT constraints(all sex);

generalize(whole of FACULTY, part of TA) into TEACHER;

generalize(elder of TEACHER, younger of STUDENT) into PERSON;

merge(ADVISING1,ADVISING2) into Advising;

interact(teach of FACULTY,study of GRAD) into advising (fac-no FACULTY. ss, g-no GRAD. ss) Constraints (teach;study=1:n);

4 G 关系的产生

在一个 OSAM* 数据库系统中,对象是通过 G 关系来表示的.^[8]一个 G 关系是一个关系的扩展,它支持 OID 和嵌套结构.为了支持 OSAM* 查询处理,我们需要为全局模式中的每个基类和导出类产生相应的 G 关系.对于 G 关系的产生,我们采用如下规则:

- (1) 每个基类的 G 关系从相应的局部关系中产生;
- (2) 每个导出类的 G 关系表示为基类和其它导出类的映射.

因此在一个基类或导出类之上的所有操作实际上是在基类的 G 关系上执行的.由于在每个基类与每个局部关系之间是一一对一的映射关系,局部模式中所有约束在全局模式中也都被遵循,所以全局操作可以在局部关系上加以正确执行.

下面我们提出了对于基类和导出类 G 关系的产生算法和映射规则.在此之前先给出一些符号说明.

- (1) E, E_1, E_2, \dots, E_n 分别表示关系 R, R_1, R_2, \dots, R_n 的实体类;
- (2) GR_R 表示为关系 R 的 G 关系;
- (3) $GR_E, GR_{E_1}, GR_{E_2}, \dots, GR_{E_n}$ 分别表示实体类 E, E_1, E_2, \dots, E_n 的 G 关系;
- (4) $Attr(R)$ 和 $Key(R)$ 分别表示关系 R 的属性集和主关键字属性;
- (5) $Sub_attr(R)$ 和 $Sub_attr(GRE)$ 分别表示关系 R 和 G 关系 GRE 的属性子集;

4.1 基类的 G 关系

一个基类的 G 关系可从相应的局部关系中直接产生.下面给出基类 G 关系的产生算法.

输入:局部关系 R_1, R_2, \dots, R_n , 基类 E_1, E_2, \dots, E_n

输出:基类 E_1, E_2, \dots, E_n 的 G 关系 $GR_E, GR_{E_1}, GR_{E_2}, \dots, GR_{E_n}$

算法.

(1) 对于每个局部关系 R_1, R_2, \dots, R_n :通过映射每个元组的关键字属性为唯一的 OID 来创建一个 OID 分配表(OAT). $OAT_i = \{ \langle OID, Key(R_i) \rangle \}$ 对于所有的 $1 \leq i \leq n$.

(2) 对于每个局部关系 $R_i (1 \leq i \leq n)$:通过一个自然连接操作来产生一个初始的 G 关系 GR_{R_i} .

$$GR_{R_i} = R_i \bowtie OAT(R_i) \text{ 对于所有的 } 1 \leq i \leq n.$$

(3) 对于每个基类 E :从相应的局部关系 R 中创建一个 G 关系.

$$GR_E = \prod_{sub_attr(R)} (\sigma_P(GR_R))$$

其中 P 是由查询表达式给出的谓词.

4.2 导出类的 G 关系

由于当所有的基类被产生以后一个导出类是从几个基本类和(或)另外几个已存在的导出类上产生的,所以我们可以从相关的那些实体类的 G 关系中产生导出类的 G 关系.为了创建导出类的 G 关系,我们提出了如下转换规则:

规则 1: GENERALIZE 规则

设 E 是 E_1, E_2, \dots, E_n 通过一个 GENERALIZE 操作产生的一个实体类,则

$$GR_E = \bigcup_{i=1}^n \left(\prod_{i=1}^n (A_attr(E_i)) GR_{E_i} \right)$$

规则 2: MERGE 规则

设 E 是 E_1, E_2, \dots, E_n 通过一个 MERGE 操作产生的一个实体类, 则

$$GR_E = \bigcup_{i=1}^n GR_{E_i}$$

规则 3: GLINK 规则

设 E' 是通过一个 GLINK 操作从实体类 E_1 和 E_2 中产生的概括层次中的一个超类, 则

$$GR_{E'} = GR_{E_1} \cap \left(\prod (A_attr(E_i)) GR_{E_i} \right)$$

规则 4: AGGREGATE 规则

设 E 是通过一个 AGGREGATE 操作从另一个实体类 E_1 的属性子集中产生的一个实体类, 则

$$GR_E = \prod_{sub_attr(GR_{E_1})} (GR_{E_1})$$

由于其它的操作如 INTERACT 和 FOREIGN-KEY 仅仅影响全局模式的结构而并不构成或分解对象, 因此它们的转换可以从它们的定义中直接进行。

5 结 论

为了转换一个关系数据库到一个面向对象的语义数据库中, 本文提出了一个基于语义确认的 2 段模式转换方法, 第 1 阶段主要是捕捉存在于局部关系模式内的关系内语义并转换到全局模式中来; 第 2 阶段主要通过模式重构操作将存在于局部关系模式之间的关系内语义转换到全局模式中来。本文提出了 5 种模式重构操作和 8 条模式转换语句, 并讨论了基类和导出类的 G 关系产生规则。基于该方法的模式转换系统已在 SUN 工作站环境下开发出来。

参考文献

- 1 Su S Y W, Krishnamurthy U, Lam H. An object-oriented semantic association model(OSAM*). *AI in Industrial Engineering and Manufacturing: Theoretical Issues and Applications*, Kumara S *et al.* eds. American Institute of Industrial Engineering, 1989.
- 2 Krishnamurthy V, Su S Y W, Lam H. IMDAS—an integrated manufacturing data administration system. *Data and Knowledge Eng.*, No. , North Holland, Amsterdam, 1988.
- 3 张霞, 郑怀远. 基于面向对象和语义关联的异构型分布式数据库系统. *微型计算机*, 1989, 3: 56~62.
- 4 Alashqut A M, Su S Y W, LAM H. OQL: a query language for manipulating object-oriented database. *Proc. of 15th VLDB*, Amsterdam, 1989. 422~433.
- 5 Zhang X, Zheng H. OSDL: the global data language of CIMBASE. *Proc. of Int'l Conf. on CIMS*, Beijing, 1991. 88~87.
- 6 Sheth A P, Larson J A. Federated database system for managing distributed heterogeneous, and autonomous databases. *ACM Computing Surveys*, 1990, 22(3): 183~236.
- 7 Date C J. *An introduction to database system*. Addison-Wesley, 1991.
- 8 王国仁, 郑怀远. CIMBASE 系统中同化器的设计与实现. *计算机工程*, 1992, (3): 45~51.

SEMANTIC CAPTURE AND SCHEMA TRANSFORMATION FROM RELATIONAL DATABASES INTO OBJECT—ORIENTED SEMANTIC DATABASES

Wang Guoren Yu Ge Shi Jing Shan Jidi Zheng Huaiyuan

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

Abstract This paper presents a 2—phase schema transformation method for schema transformation from relational databases into object—oriented semantic databases. The method is based on the semantics identification. The semantics included in local relation schemas can be divided into two parts: one are intra—relation semantics existing within local relations and the others are inter—relation semantics existing among local relations. All intra—relation semantics are captured into the global integration schema in phase 1, and all inter—relation semantics are enriched into the global integration schema by schema reconstruction operations in phase 2. The mapping rules and transformation operations are defined for setting up the transformation mechanism, and a set of algorithms and transformation formulations are provided for generating G—relations.

Key words Semantic capture, schema integration, schema mapping, query translation.