

一种面向对象的多媒体概念模型*

张霞 刘积仁 李华天

(东北大学软件中心 沈阳 110006)

摘要 本文形式化地提出了一种适用于多媒体对象的概念模型 OMCOM. 文中首先分析了多媒体数据模拟的特殊性,探讨了现有的多媒体数据模型的不足,从而提出了把多媒体数据模拟划分为概念与表达两级模拟的思想. 概念级模拟多媒体对象的内容与结构,表达级模拟其表达,包括多媒体对象之间的时态关系和空间关系. OMCOM 就是一种用于实现概念级模拟的模型. 然后根据多媒体概念模拟的需求,形式化地定义了 OMCOM 的各个构成要素. 最后根据 OMCOM 实例的格性质,建立了多媒体对象在概念级的运算体系——对象代数,形式化地定义了其中的各种运算以及对象代数表达式.

关键词 多媒体,面向对象数据库,数据模型,概念模拟,对象代数.

众所周知,在 90 年代的计算机领域中独领风骚的当属多媒体计算技术 (Multimedia Computing),它的应用是 90 年代计算机的又一次革命. 那么,什么是多媒体? 媒体 (Medium) 是指信息的载体,可以是数字、字符、文本、图形、图象、声音和视象,甚至可以是气味、味觉、触觉和温度. 多媒体 (Multimedia) 是指将多种信息媒体 (如文本、图形、图象、声音、视象等) 进行计算机编组处理的技术. 我们把多媒体系统中所处理的数据叫做多媒体数据,包括数字、字符串、文本、图形、图象、声音、视象等. 对这些信息进行统一管理和运用的数据库就是多媒体数据库. 多媒体数据库系统的应用前景十分广阔,可用于办公信息系统、地理信息系统、CAD/CAM 系统、期刊出版系统、医疗信息系统、机器人系统、生态仿真系统以及各种军事应用系统.

多媒体数据库技术从作为传统数据库的一种高级应用开始,到作为适用于多媒体数据的数据库管理系统,已经历了近 10 年的研究与发展. 人们根据自己的理解和应用环境,设计各自的系统. 目前,实现多媒体数据库系统的途径可分为 3 类: (1) 扩充关系数据库方法,如 Informix - Online^[1], ORACLE 7. 0, INGRES 6. 0 等; (2) 面向对象的方法,如 MULTOS^[2, 3], OMEGA^[4], Zenith^[5] 等; (3) 超文本 (或超媒体) 的方法,如 KMS^[6], Intermedia^[7] 等. 关于多媒体 DBMS 应具备的一般特征,至今尚未取得共识,在多媒体数据模拟方面表现尤为突出.

* 本文研究得到国家自然科学基金重点项目的资助. 作者张霞,女,1965 年生,讲师,主要研究领域为多媒体数据库,分布式数据库. 刘积仁,1955 年生,教授,博士生导师,主要研究领域为分布式多媒体信息处理方法学及支撑平台,协议工程. 李华天,1922 年生,教授,博士生导师,主要研究领域为计算机网络,协议工程,分布式系统中的多媒体应用.

本文通讯联系人:张霞,沈阳 110006,东北大学软件中心

本文 1995-06-05 收到修改稿

本文首先分析了多媒体数据模拟的特殊性,提出把多媒体数据模拟分为概念模拟和表达模拟的思想,然后形式化地定义多媒体概念模型 OMCOM(object-oriented multimedia conceptual model),最后建立对象代数体系,它是概念数据库查询的数学基础。

1 多媒体数据模拟的特点

多媒体数据具有复合性、分散性、时序性的特点。复合性指数据的形式多种多样,可以是文本、图形、图象、声音、全动画视象,而且可以通过各种信息单元集成而得的复合对象。从结构上可分为2大类:一类是格式化数据,包括数字、字符;另一类是非格式化数据,包括文本、图形、图象、声音和视象,非格式化数据具有数据量极大、处理复杂等特点。分散性指有关联的数据可以分散地存储在不同的机器、不同的设备上,可用不同的(甚至是异构的)数据库系统来存储与管理,这些数据库可以是存储在光盘上的大型归档数据库,也可以是存储在高速磁盘上的动态数据库。一些多媒体数据(如声音、视象)本身就是时序性的,而多媒体信息实体之间的关系也可能是时序性的,在表达时,还需考虑它们之间的空间关系,确定合理的空间布局,因此在编组时要保证成员信息单元之间时间上的同步/空间上的搭接,对实时性的数据要求更高。

多媒体数据的这种复合性、分散性、时序性对数据模拟提出如下需求^[8,9]:支持丰富的数据类型及相应的处理;说明不完备信息;扩充个别对象的定义到其类型定义之外;编组来自不同数据库的数据并加以一致性处理;描述结构化信息;模拟对象的内部概念与外部表达;支持上下文无关和上下文有关的引用;支持数据共享;支持版本的生成与控制;支持系统预定义的操作和用户定义的操作;支持对象的同步与集成。然而,现有的多媒体数据模型均无法满足这些需求。我们把现有的多媒体数据模型分为3类:关系数据模型、面向对象的数据模型和超文本数据模型。关系数据模型没有捕捉数据实体之间的语义关系、时态关系和空间关系,有关的处理都留给应用程序,通常基于关系数据模型的多媒体 DBMS 只实现多媒体数据的存取。尽管超文本数据模型能够在一定程度上模拟多媒体对象之间的语义关系和时空关系,但由于从本质上说,超文本是由语义网络加上浏览机制构成的,而面向对象的数据模型在结构上也是一种受限的语义网络,因此,超文本系统本身用面向对象的方法实现比较合适。另外,封装性(面向对象的核心概念)适用于多媒体数据库中各种媒体数据的不同存取与操作的实现。因此,我们认为用对象范例来建立多媒体数据模型是3种方法中最适宜的一种。但是核心的面向对象的数据模型并不能完全满足多媒体数据库的需求。研究表明,多媒体数据模型还应该具有如下几个方面的功能:

- (1)模拟非格式化数据(如文本、图形、图象、声音、视象)的内容;
- (2)允许有类型未定义的对象存在,类型只作为多媒体对象存在于数据库中的充分条件,而不是必要条件;
- (3)捕捉多媒体对象之间更丰富的语义,如 IS—PART—OF 等;
- (4)捕捉多媒体对象之间的时间、空间关系;
- (5)模拟多媒体数据的表达,即抽象描述多媒体对象的播放(Play—Out),一方面要规定表达的形式;另一方面还必须描述多媒体对象的播放时序关系和播放空间布局关系,即模拟多媒体对象的时空编组。

为了便于实现,我们提出从概念和表达 2 个层次上模拟多媒体数据的思想,概念级模拟多媒体对象的内容与结构,表达级模拟其表达,包括多媒体对象之间的时态关系和空间关系.为此,我们分别设计了多媒体概念模型:OMCOM 和多媒体表达模型:演员模型.前者基于对象范例,在上述前 3 个方面对核心的面向对象的数据模型加以增强;后者建立在前者之上,基于库所——时间区间 Petri 网,描述多媒体对象之间的时态关系,同时引入与变迁相关联的场景表达式,描述多媒体对象之间的空间关系,从而模拟多媒体数据的时空编组(详见文献[13,14]).

2 概念模型构成要素的形式定义

定义 2.1. 对象,结构上定义为

(1)原子对象是一种对象,原子对象指的是在概念上不可再分割的数据,如数字、字符、文本、图形、图象、声音、视象型数据.

(2)如果 O, O' 是对象, O' 通过属性 a 引用 O , 且 O 的存在依赖于 O' 的存在, 则我们称 O 为 O' 的依赖对象, 表示为 δO , 依赖对象也是一种对象.

(3)如果 O_1, O_2, \dots, O_n 是对象, 那么 $O = [a_1:O_1, a_2:O_2, \dots, a_n:O_n]$ 也是一种对象, 称为元组对象, 其中 a_1, a_2, \dots, a_n 为可区分的属性名, O_i 又可以表示为 $O.a_i$.

(4)如果 O_1, O_2, \dots, O_n 是对象, 那么 $O = \{O_1, O_2, \dots, O_n\}$ 也是一种对象, 称为集合对象.

(5) \top (不相容) 和 \perp (“无定义”) 是两个特殊对象.

①令 $A = \{a_1, a_2, \dots, a_n\}$ 为元组对象 O 的属性集, 则

$$O = [a_1:O_1, a_2:O_2, \dots, a_n:O_n] \leftrightarrow \forall x (x \in A \rightarrow O.x = \perp)$$

$$O = [] \leftrightarrow \forall x (x \in A \rightarrow O.x = \perp)$$

② $[\dots, a_i; \top, \dots] = \top, \{\dots, \top, \dots\} = \top$

③ $\delta \top = \top, \delta \perp = \perp$

(6)对象的所有形式均由原子对象通过(1)~(5)所确定的规则经有限次复合而得,不存在在其他形式.

定义 2.2. 设 O 为任意对象, 则对象 O 的值 $value(O)$, 递归地定义为

(1)如果 O 为原子对象, 那么 $value(O) = O$.

(2)空值 φ 是一种值. 这里, 空值意味着“现在不知其值”.

(3)如果 O 是 O' 的依赖对象, 那么 $value(O)$ 存在, 当 O' 存在; $value(O)$ 不存在, 当 O' 不存在.

(4)如果 $O = [a_1:O_1, a_2:O_2, \dots, a_n:O_n]$, 那么

$$value(O) = [a_1:value(O_1), a_2:value(O_2), \dots, a_n:value(O_n)].$$

(5)如果 $O = \{O_1, O_2, \dots, O_n\}$, 那么

$$value(O) = \{value(O_1), value(O_2), \dots, value(O_n)\}.$$

在面向对象数据模型中, 每个对象具有一个唯一的、不可改变的标识, 这种标识提供一种独立于对象的状态或表现来指称和引用对象的方法, 在系统中, 对象可以独立于它的值而存在. 在综合分析了现有的各种对象标识方法之后, 我们认为用代理值 (Surrogate) 标识对

象,对于大型分布式数据库系统是行之有效的.因为,用代理值标识对象,不但能使对象标识完全独立于数据(即对象值)和地址(即存储位置),而且易于保证它的唯一性、连续性和稳定性.每个对象的代理值在多机、多用户、多程序的整个分布式环境中是唯一的;代理值由系统自动产生,允许有一定特权的用户读出该代理值,但系统从不、也不允许用户对代理值进行修改,从而使之保持连续性;代理值与其所代表的对象一一对应,即任一对象的代理值不得为空值,也不得多于 1 个,同时不存在无对象与之对应的代理值.在 OMCOM 中,对象可分为原子对象和非原子对象 2 类,原子对象又可分为格式化原子对象和非格式化原子对象 2 类.格式化原子对象的对象标识以该对象的值来表示;非格式化原子对象包括文本、图形、图象、声音和视象,这种对象的对象标识以代理值来表示.对于独立非原子对象而言,其对象标识也以代理值来表示;而依赖非原子对象的对象标识以该对象的值来表示.

在 OMCOM 中,用型来概括具有相同特征的对象集合的共同特征.型的定义实质上就是这些共同特征的抽象描述.

定义 2.3. 型,结构上定义为

- (1)原子型是一种型,原子型指在概念上不可再分割的型,包括数字型、字符型、文本型、图形型、图象型、声音型、视象型.
- (2)“无定义”型(表示为 \perp)也是一种型.
- (3)如果 t, t' 是型, t' 通过属性 a 引用 t , 且 t 的存在依赖于 t' 的存在, 则我们称 t 为 t' 的依赖型 δt , 依赖型也是一种型.
- (4)如果 t_1, t_2, \dots, t_n 是型, 那么 $t = [a_1:t_1, a_2:t_2, \dots, a_n:t_n]$ 也是型, 称为元组型, 其中 a_1, a_2, \dots, a_n 为可区分的属性名.
- (5)如果 t' 是型, 那么 $t = \{t'\}$ 也是型, 称为集合型.
- (6)型的所有形式均按(1)~(5)所确定的规则经有限次复合而得, 不存在其他形式.

从上述定义可知,型的属性的定义域可以是任意型,即可以是原子型、“无定义”型、依赖型、集合型或元组型.其中“无定义”型的引入是为了满足“多媒体数据库中应允许有类型未定义的对象存在”的需求,允许动态定义对象的类型,即在生成对象时,定义对象的类型;依赖型的引入是为了捕捉多媒体对象之间的 IS-PART-OF 语义;另外,型的一个属性的定义域也可以是该型本身,即允许递归定义的型存在.系统所支持的原子型可分为基本集和扩充集两部分,基本集沿用 C++ 的全部类型集,其中用“void *”表示指向类型未定义对象的指针;扩充集是为了支持多媒体数据的存取而增加的类型集合,包括 text, graphics, image, audio 和 video.

类是由共享同一属性和方法集合(即具有相同特征)的对象组合在一起构成的.在数据库中对象必须作为类的实例而存在,且只属于一个类,对象与类的关系反映 instance-of 联系.类与型是两个不同的概念,型定义特征,而类定义具有某种特征(型)的对象集合,即可以把类视为型的域.

定义 2.4. 类,假设 t 为任意型,则 t 的类 $D(t)$ 定义为

- (1)如果 t 为原子型,则 $D(t) \subseteq \langle \text{预定义的域} \rangle$, 其中预定义的域是由全部具有预定义型的对象所构成的集合.
- (2)如果 t 为“无定义”型,则 $D(t) \subseteq \bigcup_{i=1}^n \langle \text{预定义的域} \rangle$, 其中 n 为系统所支持预定义

型的个数.

(3)如果 t 是 t' 的依赖型 δt , 则 $D(\delta t) = \delta D(t)$, 即依赖型 δt 的类构成依赖类, 它的全部实例依赖于 $D(t')$ 的实例的存在而存在.

(4)如果 t 是元组型, 即 $t = [a_1:t_1, a_2:t_2, \dots, a_n:t_n]$, 则

$$D(t) = a_1:D(t_1) \times a_2:D(t_2) \times \dots \times a_n:D(t_n).$$

(5)如果 t 是集合型, 即 $t = \{t'\}$, 则 $D(t) = \{O | O \in D(t')\}$.

(6)类的所有形式均按(1)~(5)所确定的规则, 经有限次复合而得, 不存在其它形式.

OMCOM 允许用户从现有型中导出新的子型, 所有的型组成一个有根的有向无环图, 我们称之为型层系. 一个型从其型层系中直接或间接祖先那里继承所有的属性和方法. 从语义上讲, 子型是其超型的特化(Specialization), 它从超型继承属性和方法; 反之, 超型是其子型的概括(Generalization). 型层系捕捉的是超型与其子型之间的 IS-A 联系.

定义 2.5. ISA 映射, 定义为

(1) 设 $t = [a_1:t_1, \dots, a_n:t_n]$, $t' = [a_{i_1}:t_{i_1}, \dots, a_{i_m}:t_{i_m}]$ 为任意元组型, 且 $\{i_1, \dots, i_m\} \subseteq \{1, \dots, n\}$, $1 \leq i_j \leq n$, 则 $I_{t,t'}$ 为从 t 到 t' 的 ISA 映射: $I_{t,t'}: D(t) \rightarrow D(t')$

$$[a_1:x_1, \dots, a_n:x_n] \rightarrow [a_{i_1}:x_{i_1}, \dots, a_{i_m}:x_{i_m}], x_i \in D(t_i).$$

(2) 设 $t = \{s\}$, $t' = \{s'\}$ 为任意 2 个集合型, 且 $I_{s,s'}$ 为从 s 到 s' 的 ISA 映射, 则 $I_{t,t'}$ 为从 t 到 t' 的 ISA 映射: $I_{t,t'}: D(t) \rightarrow D(t')$

$$\{x_1, x_2, \dots, x_n\} \rightarrow \{I_{s,s'}(x_1), I_{s,s'}(x_2), \dots, I_{s,s'}(x_n)\}, x_i \in D(s).$$

定义 2.6. 型层系

型层系 H 是一种有根的有向无环图, 其中结点来自型集合 $TypeSet$, 边 $\langle s, t \rangle$ ($s, t \in TypeSet$) 表示 ISA 映射: $I_{s,t}: D(s) \rightarrow D(t)$, 则型 s 叫做型 t 的子型, 当且仅当存在型 $t_1, \dots, t_n, n \geq 2$, 使 $s = t_1, t = t_n$, 并且 $\langle t_i, t_{i+1} \rangle \in H, i = 1, \dots, n-1$. 如果 $n=2$, 则 s 叫做 t 的直接子型. 如果 s 是 t 的子型, 则 t 也叫 s 的超型.

定义 2.7. 方法

方法定义为一个三元组 $M = \langle (t, m), s, b \rangle$, 其中

(1) $(t, m) \in TypeId \times MethId$, 其中 $TypeId$ 为型标识符的有限集合, $MethId$ 为方法标识符的有限集合; t 为型标识符, m 为方法标识符, t 为 m 所作用的型, 一个方法标识符 m 可以应用于多个型, 系统支持多态性.

(2) $s: (t, m) \rightarrow (t_1, \dots, t_n, t') \in TypeId^* \times TypeId$, 是方法的说明部(Signature), 其中 t_1, \dots, t_n 为方法的参变量型, t' 为方法的结果型.

(3) $b: (t, m) \rightarrow beh \in BehSet$, 是该方法的实现部, 定义方法的语义(即行为). 其中 beh 是实现该方法的程序, $BehSet$ 是实现方法的程序集合.

定义 2.8. 私有方法

设 t 为任一型, 我们把直接在型 t 上实现的方法叫做私有方法. 如果用 $M_s(t)$ 来表示型 t 的私有方法集, 用 $M_s^+(t)$ 来表示作用于 t 的全部方法的集合, 则下式成立:

$$M_s(t) \subseteq M_s^+(t)$$

定义 2.9. 继承方法

设有任一方法 $(t, m) \in M_s^+(t)$, 但 $(t, m) \notin M_s(t)$, 则 $\exists s \langle t, s \rangle \in H$ (即一定存在 t 的一个

超型 s), 使 $(s, m) \in Ms(s)$ (即 m 在 s 上直接实现), 则 (t, m) 叫做 t 的来自于 s 的继承方法. 如果 $\exists s_1, \dots, s_n \langle t, s_1 \rangle, \dots, \langle t, s_n \rangle \in H (n \geq 2)$, 使 $(s_1, m) \in Ms(s_1), \dots, (s_n, m) \in Ms(s_n)$, 则 (t, m) 至多只能继承其中的一个方法 $(s_i, m), 1 \leq i \leq n$.

定义 2.10. 概念模式

概念模式 C 是一个六元组 $C = \langle TypeId, TypeDef, H, MethId, MethSign, MethImp \rangle$, 其中, (1) $TypeId$ 为型标识符的有限集合; (2) $TypeDef: TypeId \rightarrow$ 型的结构定义; (3) H 为型层系; (4) $MethId$ 为方法标识符的有限集合; (5) $MethSign: TypeId \times MethId \rightarrow TypeId^* \times TypeId$, 定义方法的说明部; (6) $MethImp: TypeId \times MethId \rightarrow BehSet$, 定义方法的实现部.

定义 2.11. 概念数据库实例

设 C 为任一概念模式, 则 $Instance_{c,T} = \{V_{i,T} | t \in TypeId\}$ 是概念模式 C 在时刻 T 的一个合法的概念数据库实例, 当如下条件成立:

(1) $V_{i,T}$ 是从 $ObjId$ (对象标识的无限可数集合) 到 $D(t)$ 的一个入射 (Injection), 即 $oid_1, oid_2 \in ObjId, oid_1 \neq oid_2 \rightarrow V_{i,T}(oid_1) \neq V_{i,T}(oid_2)$ 亦即 $V_{i,T}(oid_1) = V_{i,T}(oid_2) \rightarrow oid_1 = oid_2$.

我们把在时刻 T 型 t 的类中实例的对象标识符的集合表示为 $ObjId_T(t)$.

(2) 如果 $s, t \in TypeId$, 且 $\langle s, t \rangle \in H$, 则 $ObjId_T(s) \subseteq ObjId_T(t)$, 且 $V_{i,T} = I_{s,t} \cdot V_{i,T}$, 其中 “ \cdot ” 表示函数复合.

(3) 如果 $s, t \in TypeId, a: s$ (或 $a: \delta s$) 出现在型 t 的结构定义 $TypeDef(t)$ 中, 则下式必须成立: $\forall x \in V_{i,T}(ObjId)$ 在结点 $a: s$ (或 $a: \delta s$) 有值 $a: oid \rightarrow oid \in ObjId_T(s)$

即只允许引用现有对象, 数据库中不允许有对不存在对象的引用.

(4) 设 $t \in TypeId, oid \in ObjId_T(t), s \in TypeId, o \in ObjId_T(s), a: \delta s$ 出现在型 t 的结构定义 $TypeDef(t)$ 中, 在 $V_{i,T}(oid)$ 中 $a: \delta s$ 的值是 $a: o$, 那么 $V_{i,T}(o)$ 依赖于 $V_{i,T}(oid)$ 的存在而存在.

3 对象代数

定义 3.1. 对象相等关系, 递归地定义为

设 A, B 为任意 2 个对象,

(1) 如果 A, B 为原子对象, 且 A, B 相同, 则两者相等.

(2) 如果 A, B 为元组对象, 其属性集分别为 $Attr(A), Attr(B)$, 则 $A=B$ 的充要条件是 A 与 B 各对应属性值均相等, 即

$$A=B \leftrightarrow \{ \forall x [(x \in Attr(A) \wedge A.x \neq \perp) \rightarrow (x \in Attr(B) \wedge A.x = B.x)] \wedge \forall x [(x \in Attr(B) \wedge B.x \neq \perp) \rightarrow (x \in Attr(A) \wedge B.x = A.x)] \}$$

(3) 如果 A, B 为集合对象, 则 $A=B$ 的充要条件是 A, B 中元素成对相等, 即

$$A=B \leftrightarrow \{ \forall a [a \in A \rightarrow \exists b (b \in B \wedge a=b)] \wedge \forall b [b \in B \rightarrow \exists a (a \in A \wedge b=a)] \}$$

定理 3.1. 对象相等关系 = 满足自反性、对称性和传递性, 即

(1) $O=O$;

(2) $O_1=O_2 \rightarrow O_2=O_1$;

$$(3) O_1 = O_2 \wedge O_2 = O_3 \rightarrow O_1 = O_3.$$

此定理可以从对象的定义以及对象相等关系的定义直接证明.

定义 3.2. 子对象关系

对象集合 θ 上的子对象关系 \leq , 递归定义为

$$(1) \text{对于原子对象 } O_1, O_2 \in \theta, O_1 \leq O_2 \leftrightarrow O_1 = O_2;$$

$$(2) \text{对于依赖对象 } \delta O_1, \delta O_2 \in \theta, \delta O_1 \leq \delta O_2 \leftrightarrow O_1 \leq O_2;$$

$$(3) \text{对于属性集分别为 } A_1, A_2 \text{ 的元组对象 } O_1, O_2 \in \theta,$$

$$O_1 \leq O_2 \leftrightarrow \forall x [x \in A_1 \rightarrow O_1.x = \perp \vee (x \in A_2 \wedge O_1.x \leq O_2.x)];$$

$$(4) \text{对于集合对象 } O_1, O_2 \in \theta, O_1 \leq O_2 \leftrightarrow \forall x [x \in O_1 \rightarrow \exists y (y \in O_2 \wedge x \leq y)];$$

(5) $\forall O (O \in \theta \rightarrow O \leq \perp)$, 即对象集合 θ 上的任意对象均为 \perp (不相容) 的子对象; $\forall O (O \in \theta \rightarrow \perp \leq O)$, 即 \perp (“无定义”) 为对象集合 θ 上的任意对象的子对象.

定义 3.3. 子对象, 超对象

设任意对象 $O_1, O_2 \in \theta$, θ 为所有对象组成的集合, 如果 $O_1 \leq O_2$ 成立, 则称 O_1 为 O_2 的子对象, O_2 为 O_1 的超对象.

定义 3.4. 成员对象关系 $<$

对象集合 θ 上的成员对象关系 $<$, 递归定义为

$$(1) \text{对于原子对象 } O \in \theta, O < O;$$

$$(2) \text{对于对象 } \perp, \perp < \perp;$$

$$(3) \text{对于属性集为 } A \text{ 的元组对象 } O \in \theta, \forall a [a \in A \rightarrow O.a < O];$$

$$(4) \text{对于集合对象 } O \in \theta, \forall x [x \in O \rightarrow x < O];$$

$$(5) \text{设任意 } O_1, O_2, O_3 \in \theta, \text{如果 } O_1 < O_2, O_2 < O_3, \text{则 } O_1 < O_3.$$

定义 3.5. 成员对象

设任意对象 $O_1, O_2 \in \theta$, θ 为所有对象组成的集合, 如果 $O_1 < O_2$ 成立, 则称 O_1 为 O_2 的成员对象. 从定义可以看出, 成员对象关系 $<$ 满足传递性, 与子对象关系的概念不同, 成员对象关系描述的是一个对象与其构成部分之间的关系. 成员对象可以用成员对象标识符来表示: 设任意对象 $O_1, O_2 \in \theta$, 有 $O_1 < O_2$, 则 O_1 的成员对象标识符为 $O_2.O_1$.

定理 3.2. 子对象关系 \leq 是对象集合 θ 上的一个偏序关系. (证明过程详见文献[14])

定义 3.6. 对象并 \cup

设 O_1, O_2 为任意 2 个对象,

$$(1) \text{对于任意对象 } O, O \cup \perp = \perp, O \cup \perp = O, O \cup O = O;$$

$$(2) \text{如果 } O_1, O_2 \text{ 为原子对象, 且 } O_1 \neq O_2, \text{则 } O_1 \cup O_2 = \perp;$$

$$(3) \text{如果 } O_1, O_2 \text{ 在最高层次上具有不同种类的结构, 则 } O_1 \cup O_2 = \perp;$$

$$(4) \text{如果 } O_1, O_2 \text{ 为元组对象, 它们的属性集分别为 } A_1, A_2, \text{设 } A_1 \cup A_2 = \{a_1, a_2, \dots, a_n\};$$

对于 $i=1, 2, \dots, n$, 相应于属性 a_i , O_1, O_2 的成员对象分别为 O_{1i}, O_{2i} (若 $a_i \in A_1$, 则 $O_{1i} = \perp$; 若 $a_i \in A_2$, 则 $O_{2i} = \perp$).

如果 $\forall i [i \in \{1, \dots, n\} \rightarrow O_{1i} \cup O_{2i} \neq \perp]$, 则 $O_1 \cup O_2 = [a_1: O_{11} \cup O_{21}, \dots, a_n: O_{1n} \cup O_{2n}]$; 否则 $O_1 \cup O_2 = \perp$;

$$(5) \text{如果 } O_1, O_2 \text{ 为集合对象, 则 } O_1 \cup O_2 \text{ 为 2 个集合的并, 即 } O_1 \cup O_2 = \{o \mid o \in O_1 \vee o \in O_2\}.$$

$\in O_2$);

(6)对象并 \cup 运算按(1)~(5)所确定的规则经有限次递归进行,不存在其它形式.

定义 3.7. 对象交 \cap

设 O_1, O_2 为任意 2 个对象,

(1)对于任意对象 $O, O \cap \perp = \perp, O \cap \top = O, O \cap O = O$;

(2)如果 O_1, O_2 为原子对象,且 $O_1 \neq O_2$, 则 $O_1 \cap O_2 = \perp$;

(3)如果 O_1, O_2 在最高层次上具有不同种类的结构, 则 $O_1 \cap O_2 = \perp$;

(4)如果 O_1, O_2 为元组对象, 它们的属性集分别为 A_1, A_2 , 设 $A_1 \cup A_2 = \{a_1, a_2, \dots, a_n\}$;

对于 $i=1, 2, \dots, n$, 相应于属性 a_i, O_1, O_2 的成员对象分别为 O_{1i}, O_{2i} (若 $a_i \in A_1$, 则 $O_{1i} = \perp$; 若 $a_i \in A_2$, 则 $O_{2i} = \perp$).

如果 $\forall i [i \in \{1, \dots, n\} \rightarrow O_{1i} \cap O_{2i} = \perp]$, 则 $O_1 \cap O_2 = []$; 否则 $O_1 \cap O_2 = [a_1: O_{11} \cap O_{21}, \dots, a_n: O_{1n} \cap O_{2n}]$;

(5)如果 O_1, O_2 为集合对象, 则 $O_1 \cap O_2$ 为两个集合的交, 即 $O_1 \cap O_2 = \{o \mid o \in O_1 \wedge o \in O_2\}$;

(6)对象交 \cap 运算按(1), (5)所确定的规则经有限次递归进行,不存在其他形式.

定理 3.3. 对象集合 θ 关于子对象关系 \leq 构成一个格. (证明过程详见文献[14])

我们把格 $[\theta, \leq]$ 称为对象格. 由定理 3.3 的证明^[14]可知, 对于任意的 $O_1, O_2 \in \theta, O_1 \cup O_2$ 等于 O_1 和 O_2 的最小上界, $O_1 \cap O_2$ 等于 O_1 和 O_2 的最大下界, 因此, 对象格 $[\theta, \leq]$ 诱导出一个代数系统 $[\theta, \cup, \cap]$. 根据格及其所诱导的代数系统的性质, 我们得出如下结论:

定理 3.4. 对于任意 $a, b, c \in \theta$, 有

$L1: a \cup b = b \cup a, a \cap b = b \cap a$ (交换律)

$L2: a \cup (b \cap c) = (a \cup b) \cap c, a \cap (b \cup c) = (a \cap b) \cup c$ (结合律)

$L3: a \cup a = a, a \cap a = a$ (幂等律)

$L4: a \cup (a \cap b) = a, a \cap (a \cup b) = a$ (吸收律)

即对象集合 θ 上的 2 个运算: 对象并 \cup 和对象交 \cap 满足交换律、结合律、幂等律和吸收律.

从本质上讲, 从数据库中检索数据是提取对象的子对象的过程, 为了便于描述提取子对象的过程, 我们在对象格 $[\theta, \leq]$ 及其所诱导的代数系统 $[\theta, \cup, \cap]$ 的基础上, 引入如下运算, 这些运算与对象并 \cup 和对象交 \cap 运算一起在对象集合 θ 上构成一个代数系统, 我们称之为对象代数.

由于对象代数运算不局限于对象的某一层, 可能涉及到一个复杂对象的各个层次, 如提取子对象的子对象. 因此, 对象变量的标识体系必须能够明确区分相应于复杂对象的各个层次的变量, 我们采用已叙述过的成员对象标识符来表示具有任意复杂结构的对象变量.

定义 3.8. 对象差 $-$

设任意 2 个对象 $O_1, O_2 \in \theta$

(1)对于任意对象 $O \in \theta, O - \perp = O, \perp - O = \perp, O - \top = \perp, \top - O = \top, O - O = \perp$;

(2)如果 O_1, O_2 为原子对象, 且 $O_1 \neq O_2$, 则 $O_1 - O_2 = O_1$;

(3)如果 O_1, O_2 在最高层次上具有不同种类的结构, 则 $O_1 - O_2 = O_1$;

(4)如果 O_1, O_2 为集合对象, 则 $O_1 - O_2$ 为 2 个集合的差, 即

$$O_1 - O_2 = \{o \mid o \in O_1 \wedge o \notin O_2\};$$

(5)如果 O_1, O_2 为元组对象, 设 O_1, O_2 的属性集为 A_1, A_2 , 其中 $A_1 = \{a_1, a_2, \dots, a_n\}$, 则 $O_1 = [a_1:O_{11}, a_2:O_{12}, \dots, a_n:O_{1n}]$, 相应于属性 $a_i (i=1, \dots, n)$, O_2 的成员对象为 O_{2i} (如果 $a_i \in A_2$, 则 $O_{2i} = \perp$), 则 $O_1 - O_2 = [a_1:O_{11} - O_{21}, a_2:O_{12} - O_{22}, \dots, a_n:O_{1n} - O_{2n}]$;

(6)对象差一运算按(1)~(5)所确定的规则, 经有限次递归进行, 不存在其他形式。

从 OMCOM 的形式定义来看, 在结构上, 传统的关系模型是它的特例。关系代数中的 2 个基本运算: 投影和选择, 是根据关系模型中关系的二维表格的结构特征而定义的。投影被用来从一个关系中选取某些特定的属性, 而选择被用来从一个关系中选取具备某些条件的元组。2 种运算均具有封闭性, 即运算的结果也是关系, 但投影运算的结果与原关系在内涵上有所改变, 选择运算的结果与原关系在外延上有所变化。那么在对象代数中是否也应该定义这样 2 种运算? 如果从 OMCOM 的角度出发, 把一个关系对象看作元组的集合, 即 $R = \{O_1, \dots, O_n\}$, 则投影运算选取各元组对象的子对象, 这些子对象具有相同的属性, 运算结果为这些子对象所构成的集合, 即 $R' = \{O'_1, \dots, O'_n\}$, 其中 $O'_i \leq O_i (i=1, \dots, n)$; 而选择运算选取集合 $\{O_1, \dots, O_n\}$ 的子集, 其结果为集合对象 R 的一个子对象 $R' \leq R$ 。就 OMCOM 而言, 投影与选择运算的作用是相同的, 即选取子对象。另一方面, 从对象的定义可以看出, 对象是递归定义的, 元组与集合 2 种构造子具有正交性, 即元组对象的属性对象可以是集合对象, 而集合对象的元素可以为元组对象。因此, 在对象代数中不宜单独定义投影运算和选择运算。基于这种想法, 我们定义了一种统一的选取子对象的运算, 命名为映择, 符号表示为 Γ_ρ , 其中 Γ 为映择运算符; ρ 为映择表达式, 确定映择的条件。

定义 3.9. 对象映择 Γ_ρ

设任意对象 $O \in \theta$, 则对象 O 关于表达式 ρ 的映择运算定义为 $O' = \Gamma_\rho(O), O' \in \theta$ 且下式成立: $O' \leq O \wedge O'$ 满足 ρ , 其中 ρ 为映择表达式, 确定映择运算的条件, 其递归定义如下:

(1) \emptyset (空), 是映择表达式。

(2)谓词表达式, 是映择表达式。谓词表达式的运算对象为常量或属性对象变量(用成员对象标识符范例表示); 运算符可以包括算术比较运算符($<, =, >, \leq, \geq, \neq, \in$)和逻辑运算符($\wedge, \vee, \rightarrow$)。

(3)如果 $\rho_1, \rho_2, \dots, \rho_n$ 为映择表达式, 则 $\rho = [a_1:\rho_1, \dots, a_n:\rho_n]$ 为映择表达式。如果 $n=1$, 则 $[\]$ 可以省略; 如果 $\rho_i = \emptyset$, 则 $a_i:\rho_i$ 可表示为 a_i 。

(4)如果 ρ' 为映择表达式, 则 $\rho = \{\rho'\}$ 为映择表达式, 最外层 $\{\}$ 可以省略。

在 OMCOM 中, 元组对象具有复杂的层次结构关系, 元组对象的属性对象可以为元组、集合等, 因此, 关系代数中一个十分重要的运算—连接(Join)的定义对于 OMCOM 中的元组对象并不适用。为了提供类似的功能, 我们重新定义了连接(Join)运算。

定义 3.10. 对象连接 X

设 O_1, O_2 为组成元素为元组对象的集合对象,

$$O_1 = \{[a_1:O_{11}, \dots, a_n:O_{1n}, c_1:O'_{11}, \dots, c_k:O'_{1k}]\}$$

$$O_2 = \{[b_1:O_{21}, \dots, b_m:O_{2m}, c_1:O'_{21}, \dots, c_k:O'_{2k}]\}$$

其中 c_1, \dots, c_k 为 2 个对象的公共元组对象属性, $n \geq 0, m \geq 0, k \geq 0$, 且 $n+k > 0, m+k > 0$,

则 O_1, O_2 的对象连接 X 定义为

$$O_1 \times O_2 = \{O \mid O = [a_1:O_{11}, \dots, a_n:O_{1n}, b_1:O_{21}, \dots, b_m:O_{2m}, \\ c_1:O'_{11} \bar{\cap} O'_{21}, \dots, c_k:O'_{1k} \bar{\cap} O'_{2k}] \\ \wedge O'_{11} \bar{\cap} O'_{21} \neq \perp \wedge \dots \wedge O'_{1k} \bar{\cap} O'_{2k} \neq \perp \}$$

其中 $\bar{\cap}$ 为对象交运算的扩充, 定义为: 对于任意 $i=1, 2, \dots, k$, 有

(1) 如果 O'_{1i} 为集合对象, 且 $O'_{2i} \in O'_{1i}$, 则 $O'_{1i} \bar{\cap} O'_{2i} = O'_{2i}$; 如果 O'_{2i} 为集合对象, 且 $O'_{1i} \in O'_{2i}$, 则 $O'_{1i} \bar{\cap} O'_{2i} = O'_{1i}$;

(2) 对于其他任何情况, $O'_{1i} \bar{\cap} O'_{2i} = O'_{1i} \cap O'_{2i}$, \cap 为前文所定义的对象交运算.

关系代数中的自然连接运算 ∞ 是对象连接 X 运算的特例, 因为, 关系是元组对象的属性对象均为原子对象的集合对象, 对于原子对象 O_1, O_2 而言, 若 $O_1 \cap O_2 \neq \perp$, 则 $O_1 = O_2$, 而对应公共属性相等正是自然连接的条件. 因此, 在这种情况下, 对象连接运算变成了自然连接运算.

定义 3.11. 对象 σ -连接 $X_{\sigma xy}$

设 O_1, O_2 为组成元素为元组对象的集合对象,

$O_1 = \{[a_1:O_{11}, \dots, a_n:O_{1n}]\}, O_2 = \{[b_1:O_{21}, \dots, b_m:O_{2m}]\}$ ($n > 0, m > 0$) 则 O_1 与 O_2 的对象 σ -连接 $X_{\sigma xy}$ 定义为

$$O_1 \times_{\sigma xy} O_2 = \{O \mid O = [a_1:O_{11}, \dots, a_n:O_{1n}, b_1:O_{21}, \dots, b_m:O_{2m}] \wedge x \sigma y\}$$

其中 x, y 分别为 O_1, O_2 的属性, 以成员对象标识符表示; σ 为算术比较运算符, 可以是 $<, =, >, \leq, \geq, \neq, \in$.

对象和对象代数运算的组合, 就构成了对象代数表达式, 其递归定义如下:

定义 3.12. 对象代数表达式

(1) 对象本身是一个对象代数表达式;

(2) 如果 A 是对象代数表达式, 那么 $\Gamma_p(A)$ 也是对象代数表达式;

(3) 如果 A, B 是对象代数表达式, 那么 $A \times B, A \times_{\sigma xy} B, A \cup B, A \cap B$ 和 $A - B$ 也是对象代数表达式;

(4) 如果 A 是对象代数表达式, 那么 (A) 也是对象代数表达式;

(5) 在对象代数表达式中, $()$ 运算优先级最高, 其他运算符优先级相同, 运算符 Γ_p 服从右结合, 运算符 $\times, \times_{\sigma xy}, \cup, \cap$ 和 $-$ 服从左结合;

(6) 对象代数表达式的所有形式均按 (1)~(5) 所确定的规则经有限次复合求得, 不存在其他形式. 对象代数表达式的结果也是对象.

4 结束语

本文提出了一种面向对象的多媒体概念模型 OMCOM, 形式化地定义了它的基本构成要素及其运算体系——对象代数. 为了满足多媒体数据库概念模拟的需求, OMCOM 对核心的面向对象数据模型进行了扩充: 允许有类型未定义的对象存在; 提出了依赖对象的概念, 以便捕捉多媒体对象之间的 IS-PART-OF 语义; 允许用户利用 OMCOM 所支持的丰富的语义抽象机制, 对非格式化数据的内容进行概念模拟. 文中还提出了 ISA 映射和型层

系的概念,来形式化地描述 OMCOM 的概括(generalization)抽象.根据 OMCOM 中实例对象的结构特点,形式化地定义了对象相等、子对象和成员对象 3 种关系,基于它们的定义与性质,建立了多媒体对象在概念级的查询运算体系——对象代数,赋予对象并、交、差、连接和 σ -连接以新的含义,并且提出了一种新的对象代数运算——对象映择,最后提出了由对象和对象代数运算组合而成的对象代数表达式.

建立在这种新的数据模拟观念上的多媒体数据库程序设计语言 M * * 及其应用开发环境 M * * Base 的实验原型已部分实现,该系统建立在东北大学软件中心开发的分布式多媒体数据存取系统 OpenBASE 之上.

参考文献

- 1 Shetler T. Birth of the BLOB. *Byte*, 1990, **15**(2):221~226.
- 2 Meghini C *et al.* Conceptual modelling of multimedia documents. *Computer*, 1991, **24**(10):23~30.
- 3 Bertino E *et al.* Query processing in a multimedia document system. *ACM TOOLS*, 1988, **6**(1):1~14.
- 4 Yoshifumi Masunaga. Design issues of OMEGA; an object-oriented multimedia database management system. *Journal of Information Processing*, 1991, **14**(1):60~73.
- 5 Kemp Z P *et al.* Zenith system for object management in distributed multimedia design environments. *Information and Software Technology*, 1992, **34**(7):427~436.
- 6 Akscyn R M *et al.* KMS; a distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 1988, **31**(7):820~835.
- 7 Haan B J *et al.* IRIS hypermedia services. *Communications of the ACM*, 1992, **35**(1):36~51.
- 8 Berra P B *et al.* Architecture for distributed multimedia database system. *Computer Communications*, 1990, **13**(4):217~231.
- 9 Klas W *et al.* Using an object-oriented approach to model multimedia data. *Computer Communications*, 1990, **13**(4):204~215.
- 10 Davcev D *et al.* Distributed multimedia information retrieval system. *Computer Communications*, 1992, **15**(3):177~184.
- 11 Thomas D C, Little *et al.* Synchronization and storage models for multimedia objects. *IEEE J. on Selected Areas in Communications*, 1990, **8**(3):413~427.
- 12 Straube D D *et al.* Queries and query processing in object-oriented database systems. *ACM TOOLS*, 1990, **8**(4):387~430.
- 13 Zhang Xia, Liu Jiren, Li Huatian. MOPOM; an object-oriented multimedia data presentation model. *Advances in Modelling and Analysis*, 1994, **31**(2):37~56.
- 14 张霞. 基于对象范例和 Petri 网的多媒体数据库模型、语言及其实现[博士论文]. 沈阳:东北大学, 1994.

AN OBJECT-ORIENTED MULTIMEDIA CONCEPTUAL MODEL

Zhang Xia Liu Jiren Li Huatian

(Computer Software Center Northeastern University Shenyang 110006)

Abstract This paper presents a formal conceptual model for multimedia objects, name-

ly, OMCOM (object-oriented multimedia conceptual model). First, the particularities of the multimedia data modelling are analyzed, the inadequacies of existing multimedia data models are discussed. As a result, the idea is proposed that multimedia data modelling should be divided into two layers; one is the conceptual layer which models the contents and structures of multimedia objects, and the other is the presentation layer which models their presentations including their temporal and spatial relationships. OMCOM is the model used to implement the conceptual modelling. Secondly, the formal definitions of the elements in OMCOM are given according to the requirements of multimedia conceptual modelling. Finally, object algebra is defined, which is based on the lattice property of OMCOM instances. Object algebra is the system of operations over multimedia objects on the conceptual layer. The formal definitions of object algebra operators and expressions are given.

Key words Multimedia, object-oriented database, data model, conceptual modelling, object algebra.