

数据模型的智能化扩充研究 ——面向对象的知识数据库模型*

姚卿达 董祖明

(中山大学软件所, 广州 510275)

摘要 数据模型的智能化扩充旨在从数据模型着手, 提高数据库系统的性能与功能, 使其可存贮和处理复杂对象和知识, 增强结构和行为的建模能力、语义构造能力, 能演绎推理。本文承接此思路, 基于面向对象方法, 吸收语义数据模型和人工智能中的演绎推理机制, 提出一个面向对象的知识数据库模型。讨论其基本概念与性质、查询语言、与传统模型的比较和实验系统 O-CEANS 的体系结构及实现方案。

关键词 数据库, 面向对象, 知识数据库模型。

计算机新的应用领域, 如 AI, CIMS, OIS, CASE 等不断涌现, 对数据库技术提出新的要求: 定义和操纵复杂结构的能力; 行为(操作)抽象的能力; 语义表达机制; 演绎推理机制; 完整性约束表达能力等。这些也正是以关系数据库模型为代表传统模型的不足之处。

中山大学软件所的国家自然科学基金课题“数据模型的智能化扩充研究”^[12-14]为解决此问题作了一系列研究, 提出了知识数据库模型的概念, 探讨了其系统结构和工作原理, 研制了带因果联系语义的关系数据模型实验系统 RCEAS, 同时也探讨了相关的查询优化技术、基础理论及关系模式可修改性等问题。

本文提出的面向对象的知识数据库模型 OOKDM 正是在上述背景下, 以面向对象方法为基本框架, 吸收语义数据模型丰富的语义构造机制, 引进演绎数据模型的规则表示、存贮以及演绎推理机制, 综合而成。概括而言就是 OO+AI+DB。目标是从数据模型入手, 增强数据库存贮和处理复杂对象和知识的能力, 增强其结构和行为的建模能力和语义表达能力, 并可作演绎推理。数据模型智能化扩充思路是从应用中把知识、方法和智能提取出来, 融合到数据模型中。OOKDM 正是承接这一思路发展而成, 它可作为研究和实现智能数据库系统的基础。

* 本文 1991 年 8 月 23 日收到, 1991 年 12 月 24 日定稿

作者姚卿达, 57 岁, 教授, 主要研究领域为数据库与知识库。董祖明, 30 岁, 工程师, 主要研究领域为数据库, 面向对象方法。

本文通讯联系人: 姚卿达, 广州 510275, 中山大学软件所

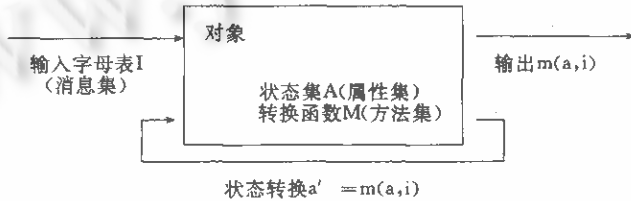
1 面向对象的知识数据库模型 OOKDM

OOKDM 模型的基本概念包括对象、类、属性、方法、约束、消息、继承性、语义构造子和规则、广义因果规则等.

1.1 对象

在 OOKDM 模型中,应用领域的事物和概念都抽象为对象.

定义. 对象是一个五元组 $Object = \langle Oid, A, M, C, I \rangle$, 其中 Oid 是对象的唯一标识符或名称, 称为对象标识; A 是对象的具有一定数据结构的私在存贮区域, 称为属性集; M 是操作集合, 称为方法集, 只有通过 M 中操作才能访问属性集 A 中的属性, 操作含接口与实现两部分; C 是定义在 A 和 M 上的约束条件, 说明对象在结构和行为上受到的限制, 称为约束集; I 是对象与外界的接口描述, 包括对象受理的操作名称及其说明, 称为消息集.



其中 $m \in M, a \in A, i \in I, a' \in A$

对象可看作一个有限自动机 $(K, \Sigma, \delta, q_0, F)$, 其中有限状态集 K 为属性集 A 包含的状态集合; 输入字母表 Σ 是消息集 I ; 状态转换函数 δ 对应方法集 M 中操作; 初始状态 q_0 对应新创建对象的属性集, 而终止状态集 F 相应于满足一定条件的对象状态.

有限自动机的模型可考虑扩充到 OO 的其它概念.

OOKDM 模型支持元组、集合、表等语义构造子.

定义. 对象递归定义如下:

(1) 原子对象 (Atom Object), 指概念上不再细分的整数型 int , 实数型 $real$, 字符型 $string$, 和逻辑型 $bool$ 等基本类型的值; 它们的值即标识自身, 其属性集、方法集、约束集和消息集是系统预定义的 (Built-in);

(2) 元组对象 (Tuple Object) $O = \langle a_1:O_1, \dots, a_n:O_n \rangle$, 其中 a_1, \dots, a_n 是可区分的属性名, O_1, \dots, O_n 是分量对象. 若 O_1, \dots, O_n 都是原子对象, 则称 O 为简单对象;

(3) 集合对象 (Set Object) $O = \{O_1, \dots, O_n\}$, 其中 O_1, \dots, O_n 是同类的不同对象, 称为元素对象;

(4) 表对象 (List Object) $O = [O_1, \dots, O_n]$, 其中 O_1, \dots, O_n 是同类的对象 (可相同), 称为成员对象;

(5) 对象空间中的每个对象都可由以上 (1) - (4) 经有限次复合而成.

为了定义对象的等值与同一, 这里给出两个特殊函数 id 和 val , 对于某给定对象 O , $id(O)$ 表示其 Oid , 而 $val(O)$ 表示其值.

定义. 对象 O 的值 $\text{val}(O)$ 递归定义如下:

(1) 若 O 是原子对象, 则 O 的值是 O 本身, 记为 $\text{val}(O)=O$;

(2) 若 O 是元组对象, 则 O 的值是

$$\text{val}(O)=\text{val}(\langle a_1:O_1, \dots, a_n:O_n \rangle)=\langle a_1:\text{val}(O_1), \dots, a_n:\text{val}(O_n) \rangle$$

(3) 若 O 是集合对象, 则 O 的值是 $\text{val}(O)=\text{val}(\{O_1, \dots, O_n\})=\{\text{val}(O_1), \dots, \text{val}(O_n)\}$

(4) 若 O 是表对象, 则 O 的值是 $\text{val}(O)=\text{val}([O_1, \dots, O_n])=[\text{val}(O_1), \dots, \text{val}(O_n)]$

定义. 给定对象 O_1 与 O_2 ,

(1) 若 $\text{id}(O_1)=\text{id}(O_2)$, 则 O_1 和 O_2 是同一个对象, 称为同一的;

(2) 若 $\text{val}(O_1)=\text{val}(O_2)$, 则 O_1 与 O_2 的值相等, 称为等值的.

定理. 对象的同一和等值具有自反性、对称性和传递性, 是一个等价关系.

1.2 对象标识

OOKDM 模型支持对象标识(Object Identifier)的概念. 对象标识是系统在创建对象时赋予它的一个系统范围内的唯一标识.

1.3 属性

对象的属性集是对象自有的具有一定数据结构的存贮区域, 表示对象自身的状态.

对象的属性有一可相互区分的标识名称, 每个属性对应一个对象类(待下文定义). 属性的取值可定义为其对应的对象的对象标识符, 即当它定义在原子对象类上时, 取实际的值; 当它定义在非原子对象类上时, 取相应对象的对象标识符.

属性也可从导出规则求值^[7], 这是 OOKDM 模型中包含的知识之一.

1.4 方法

在面向对象的框架中, 对象封装了数据和操作, 只有对象自身的操作才能访问对象的状态数据. 外界与对象的联系是通过发消息调用对象的操作来实施的. 每个操作包含接口和实现两部分. 此操作组成了对象的方法集^[1].

定义. 对象 $O=\langle \text{Oid}, A, M, C, I \rangle$ 的方法集 M 中每个操作 m 的格式为: 返回值 方法名(参数表) 方法体

在 OOKDM 模型中, 操作的接口包括操作的名称、参数及返回值的说明, 操作的具体实现(方法体)可由多种语言范例支持, 如一阶谓词演算式的 PROLOG 过程, 或是过程式的 C 语言函数, 或者某种经扩充的面向对象的 SQL 语言的查询块.

OOKDM 中的操作分为一般方法和导出属性方法两种.

1.5 约束

OOKDM 模型中, 对象包含一组约束条件, 指明对象的属性和方法应受到的限制, 这是一种语义完整性约束. 可用逻辑表达式来描述它.

定义. 对象 $O=\langle \text{Oid}, A, M, C, I \rangle$ 中约束 C 是一个逻辑表达式, 使得 A, M 满足: (1) $C(A)=\text{TRUE}$; (2) 若 $C(A)=\text{TRUE}$, 则 $C(M(A))=\text{TRUE}$.

约束规定了对象在结构和行为两方面的限制条件. 这是 OOKDM 的知识之一.

1.6 消息

在一个对象 $O=\langle \text{Oid}, A, M, C, I \rangle$ 中, 属性集 A 、方法集 M 和约束 C 对于对象外部而言

是透明的. 对象之间的交互作用是通过消息传递进行的.

定义. 给定对象 $O = \langle \text{Oid}, A, M, C, I \rangle$, 消息集 I 是对象 O 唯一的对外接口, I 中每一消息 i 包括以下部分: (1) 发送对象标识; (2) 接收对象标识; (3) 消息名称; (4) 消息参数表; (5) 返回值.

1.7 类

类 (CLASS) 是对一组相似的对象抽象. 它将该种对象的共同特征提取出来, 由该类的对象 (实例) 共享. 类的概念对应抽象数据类型的概念^[10].

定义. 类是一组相似对象的抽象, 它是一个六元组

$$\text{CLASS} ::= \langle N, A, M, C, I, R \rangle$$

其中, N 是类的标识或名称; A 是类的属性描述; M 是类中操作 (方法) 集合的接口说明与具体实现; C 是类中对象实例必须满足的约束条件谓词; I 是类中对象统一的公共对外接口 (消息集) 说明; R 是类的继承性描述, 记为 $\{ \text{CLASS1} \} \text{CLASS2}$, “ $\}$ ”表示概括 (IS-A) 与聚集 (IS_PART_OF, HAS_A) 等语义联系构造子^[10].

满足: (1) $\text{CLASS}. A = \bigcap \text{obj}. A$

(2) $\forall \text{CLASS}. M = \bigcap \text{obj}. M$

(3) $\text{obj} \in \text{CLASS}, \text{CLASS}. C(\text{obj}) = \text{TRUE}$

(4) $\text{CLASS}. I = \bigcap \text{obj}. I;$

以上 $\text{obj} \in \text{CLASS}$.

A, M, C 与对象定义相同, 同样可对对象的递归定义去定义类, 以支持元组、集合和表等类型构造子.

同一类中的对象实例具有各自的私有存贮状态, 同时共享类中的方法、约束、消息及继承性说明.

类的描述包括内部实现 (A, M 和 C 等) 和对外特性 (I 等) 两方面. 类通过描述消息的格式及其解释与处理来定义此类的对象的外部特性. 通过描述内部属性状态、方法实现和约束来定义对象的内部实现.

类抽象了一组对象的静态和动态两方面的特性. 类也隐含了“对象工厂”和“对象仓库”的概念^[1].

1.8 继承性

定义. 设有类 $\text{CLASS1} = \langle N1, A1, M1, C1, I1, R1 \rangle$ 和 $\text{CLASS2} = \langle N2, A2, M2, C2, I2, R2 \rangle$, 其中 $R1 = \{ \text{CLASS1} \} \text{CLASS2}$, 则称 CLASS1 继承 CLASS2 , 称 CLASS1 为 CLASS2 的子类, CLASS2 为 CLASS1 的超类, 其含义为: $\text{CLASS1} = \langle N1, A1 \cup A2, M1 \cup M2, C1 \cap C2, I1 \cup I2, R2 \rangle$, 即子类继承其超类的属性描述、方法描述、约束描述、接口 (消息) 描述和超本身具有的继承性. 子类可重定义其超类的方法, 这称为复载 (OVERRIDE). 若 $|R1| = 1$, 则称 CLASS1 具有单继承性; 若 $|R1| > 1$, 则称 CLASS1 具有多继承性. (注: $|R1|$ 表示集合 $R1$ 中元素的个数)

定理. 继承关系 $\}$ 具有自反性、非对称性和传递性, 它是一个偏序关系.

定理. 系统中的类的集合 C 与继承关系 $\}$ 构成一个类的偏序格 $\langle C, \rangle$.

1.9 对象模型的广义因果联系规则

在OOKDM模型中,规则有几个含义:(1)对象类中的属性导出规则;(2)对象类中的语义约束规则;(3)建立在对象一级的因果联系。

其中前两种在上文已有论述,此处着重讨论(3)。文献[12]提出了带因果联系语义的关系模型RCEAS。从这一思路继续扩充,可建立带广义因果联系网络的对象系统OCEANS(Objects with Cause—Effect Association Nets System)。

OOKDM模型中的因果联系规则CE涉及原因对象、结果对象、触发条件以及相应动作等成份,其物理表示是一个四元组:

〈原因对象,结果对象,触发条件,动作〉

在逻辑上可用一个因果联系对象来表示一条CE规则:

CE_obj = 〈Oid, A, M, C, I〉

其中Oid为CE规则的名称或标识,A包括原因对象和结果对象的匹配描述(可以用导出属性规则使一条CE规则作用到一组原因对象和一组结果对象),C表示CE规则的触发条件,而M表示相应的动作或解释。

这种广义的因果联系规则包含了通常意义下的产生式规则,这是OOKDM模型中知识的又一种含义。

1.10 对象数据库模式与实例

定义. OOKDM中,对象数据库模式定义为:DB ::= {CLASS},实例定义为:db ::= {{object}}.

2 OOKDM模型的数据/知识定义与操纵

我们为OOKDM模型设计了一个数据/知识对象定义与操纵语言的框架,称为ODML(Object Definition and Manipulation Language)。

2.1 类的定义

2.1.1 对象类

〈类〉 ::= 〈系统预定义类〉 | 〈用户定义类〉

〈系统定义类〉 ::= int | real | string | bool

〈用户定义类〉 ::= 〈元组类〉 | 〈集合类〉 | 〈表类〉 | 〈复合类〉

〈元组类〉 ::= 〈类名〉 〈分量属性名〉 {, 〈类名〉 〈分量属性名〉}

〈集合类〉 ::= { 〈类名〉 }

〈表类〉 ::= [〈类名〉]

〈用户定义类〉 ::= CLASS 〈类名〉 〈继承性定义〉; 〈属性定义〉; 〈方法定义〉; 〈约束定义〉; 〈消息定义〉 ENDCLASS;

2.1.2 继承性

〈继承性定义〉 ::= INHERITANCE: [IS-A {〈超类名〉};] [HAS-A {〈超类名〉};]

OOKDM模型支持多继承的机制。

2.1.3 属性定义

〈属性定义〉 ::= ATTRIBUTES: { 〈域类名〉 〈属性名〉 [= 〈导出属性规则〉] }

〈导出属性规则〉 ::= 〈方法名〉(〈参数表〉)〈方法体〉;

可选的导出属性规则子句用于在生成一个新对象时初始化此属性,或者是一个属性求值的过程(方法),说明如何计算(导出)此属性的值。

2.1.4 方法定义

OOKDM 模型支持方法(操作)具体实现的多语言混合机制。

〈方法定义〉 ::= METHODS: {〈返回值类型〉〈方法名〉(〈参数表〉)〈方法体〉};

〈方法体〉 ::= 〈编译伪指令〉〈具体实现〉

〈编译伪指令〉 ::= #C++ | #PROLOG | #EIFFEL | #SQL | ...

〈具体实现〉 ::= 〈C++, PROLOG 或 SQL 的语句〉

2.1.5 约束定义

〈约束定义〉 ::= CONSTRAINTS: 〈逻辑表达式〉;

〈逻辑表达式〉 ::= 〈布尔表达式〉[〈OP〉〈逻辑表达式〉] | 〈逻辑表达式〉 | NOT 〈逻辑表达式〉

〈OP〉 ::= AND | OR

〈布尔表达式〉 ::= 〈类布尔属性〉 | 〈类属性〉〈比较符〉〈类属性〉 | 〈类方法调用〉〈比较符〉〈类方法调用〉

〈比较符〉 ::= > | >= | = | <= | <

分别表示大于、大于等于、等于、小于等于、小于。

2.1.6 消息定义

〈消息定义〉 ::= MESSAGES: 〈返回值类型〉〈消息名〉(〈参数表〉);

〈参数表〉 ::= 〈参数〉{; 〈参数〉} | 〈消息名〉(〈参数〉){; 〈消息名〉([〈参数〉])}

〈消息名〉 ::= 〈方法名〉

2.2 对象的操纵

对象操纵包括创建、更新、删除和查询等。它实际上是向此对象类或对象实例发送相应的操纵消息。对象接收到此合法的消息后调用自己的方法进行相应的动作。

〈对象操纵〉 ::= 〈对象创建〉 | 〈对象删除〉 | 〈对象更新〉 | 〈对象查询〉

2.2.1 对象的创建与删除

〈对象变量定义〉 ::= {〈域类名〉〈对象变量名〉};

〈对象创建〉 ::= [〈对象变量名〉 =]〈域类名〉. NEW(〈初始化参数〉);

〈对象删除〉 ::= 〈对象变量名〉. DELETE(); | 〈域类名〉. DELETE(〈对象标识〉);

2.2.2 对象的更新

〈对象更新〉 ::= 〈对象变量名〉 UPDATE(〈参数表〉); | 〈域类名〉 UPDATE(〈参数表〉);

2.2.3 对象的查询

〈对象属性查询〉 ::= 〈返回值变量〉 = 〈对象名〉属性求值消息();

〈对象查询〉 ::= 〈结果对象名〉 =

SELECT [* | 〈消息序列〉]

FROM 〈域类名〉 {, 〈域类名〉}

WHERE〈约束条件逻辑表达式〉

〈消息序列〉 ::= { [〈域类名〉.] 〈消息名〉 ([〈参数〉]) }

在 OOKDM 模型的世界里,这种操纵语言保持了闭合的性质(即一个操作的输出可作为另一操作的输入,操作可复合).

2.3 一个例子

这一小节给出一个以大学为背景的例子,说明以上定义与操纵的用法.在此例中,有 person, teacher, student 等实体.

```

CLASS person
  ATTRIBUTES:
    [string first,string second] name;
    int age;
    person father;
  METHODS:
    {person} ancestor() #PROLOG
      THIS:ancestor(X) :- THIS:father(Y), Y:father(X).
      THIS:ancestor(X) :- THIS:father(Y), Y:ancestor(X).
  MESSAGES:
    [string first,string second] name();
    int age();
    person father();
    {person} ancestor();
ENDCLASS;
CLASS teacher
  INHERITANCES,IS-A {person};
  ATTRIBUTES:
    string title;
    int salary=comp_sal(string title) #C++
      {switch (title) {
        case "prof": return 300;
        case "lect": return 200;
        case "ta": return 100;
        default: return 500;
      }
    };
  MESSAGES:.....
ENDCLASS;
CLASS student
  INHERITANCE,IS-A {person};
  ATTRIBUTES:
    string degree;
    real gpa;int credit=comp_credit() #PROLOG
      THIS:comp_credit(150) :- THIS:degree(X), X="ba".
      THIS:comp_credit(30) :- THIS:degree(X), X="ma".
      THIS:comp_credit(20) :- THIS:degree(X), X="phd".

```

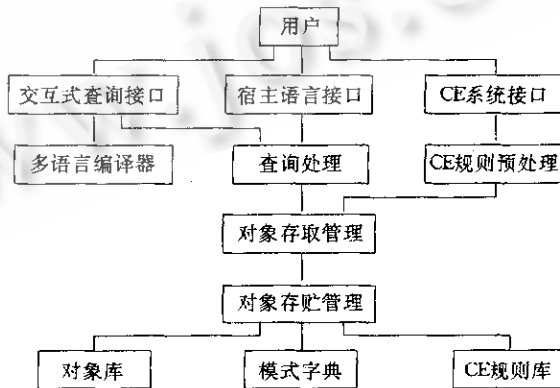
```

MESSAGES; .....
ENDCLASS;
CLASS teacher-assistent
    INHERITANCES:IS—A {student,teacher};
    ATTRIBUTES:[int year, string season]semester;
    CONSTRAINTS:gpa>80;
    MESSAGES;.....
ENDCLASS;
student s1
teacher t1;
t1=teacher.new(name(["yao","qd"]),age(53),title("prof"));
s1=student.new(name(["dong","zm"]),age(25),degree("ma"),thesis("OOKDM"),t1);
teacher.delete(t1);
student.delete(s1);
for s1 in student
    printf("NAME IS %s, THESIS IS %s, TUTOR IS %s\n",
        s1.name().first,s1.name().second,s1.thesis());
{teacher-assistent} good-TA;
good-TA=SELECT *
    FROM teacher-assistent
    WHERE teacher-assistent.member(x) AND x.gpa>90;

```

3 实验系统 OCEANS 的体系结构及实现方案

在充分利用“带因果联系的关系模型”实验系统 RCEAS^[12]的成果的基础上,以 OOKDM 为基本模型,提出一个“带因果联系网的对象模型”实验系统 OCEANS (Objects with Cause—Effect Association Nets Systems),讨论其体系结构和实现方案. OCEANS 的体系结构如图所示.



OCEANS 的用户通过交互式查询接口, 宿主语言接口和因果规则接口与系统交换信息, 可定义对象类、定义 CE 规则和对知识对象数据库进行查询. 主要功能:(1)知识对象模

式定义;(2)对象类因果联系 CE 规则定义,预编译和维护;(3)交互式 and 宿主式查询,多语言混合接口;(4)查询处理及优化;(5)对象存取管理和对象存贮管理;(6)对象因果联系查询(WHAT and HOW);(7)CE 规则的触发条件处理。

我们拟在微机上以关系数据库管理系统 ORACLE 和面向对象的程序设计语言 C++ 为基础,增加对象管理层和 CE 规则处理层,实现 OCEANS。

4 OOKDM 模型与传统数据库模型的比较

OOKDM 模型吸收了数据库模型发展史上众多成功的因素,主要是以面向对象模型为框架,结合了语义数据模型的语义表达能力和 AI 及演绎数据模型中的知识表示和演绎推理机制,以及“数据模型智能化扩充研究”已有的成果,概括起来就是:(1)可显式表示具有复杂结构的知识对象;(2)在对象中既抽象了结构特征,也抽象了行为特征;(3)支持导出数据规则的概念;(4)具有概括和聚集等丰富的语义联系表达机制;(5)可显式表达对象在结构和行为两方面受到的约束规则;(6)可表达基于知识对象的因果联系规则;(7)可定义规则并利用规则进行演绎推理;等等。

可见,OOKDM 模型在建模能力、知识表示和利用等方面优胜于传统的模型。OOKDM 模型可作为新一代智能数据库系统的基础,支持新的数据库应用。

结束语:面向对象的方法、语义数据模型和演绎(逻辑)数据模型都是当今数据库学界研究的热点。数据库技术要适应新挑战,需吸取其它领域先进的思想方法。从数据库的基础——数据模型着手改革,是提高数据库技术的最彻底的策略之一。OOKDM 正是基于此认识的一种尝试。作者希望本文既作为“数据模型智能化扩充研究”项目最新工作的小结,也作为新一轮研究的开端。欢迎指正。

参考文献

- 1 Atkinson M, Bancilhon F, DeWitt D *et al.* The object-oriented database system manifesto. DOOD'89, 1989.
- 2 Abiteboul S. Towards a deductive object-oriented database language. DOOD'89, 1989.
- 3 Biller H, Neuhold E J. Semantics of data bases; the semantics of data models. Readings in Artificial Intelligence and Databases, 1977.
- 4 Brodie M L. 未来的智能信息系统:AI 与 DB 技术的结合. 计算机科学,1989(3).
- 5 高级 DBMS 功能委员会. 第三代数据库系统宣言. 计算机科学,1991(1).
- 6 Gallaire H. Logic and databases: a deductive approach. Readings in Artificial Intelligence and Databases, 1989.
- 7 Hull R, King R. Semantic database modelling: survey, application, and research issues. ACM Computing Surveys, 1987;19(3).
- 8 拉古纳海滩讨论会. 数据库管理系统研究的未来方向. 计算机科学, 1989.
- 9 Potter W D, Trueblood R P. Traditional, semantic, and hyper-semantic approaches to data modelling. Computer, 1988;21(6).
- 10 Tsichritzis, Dionysios. Data model. Prentice-Hall, 1982.
- 11 Ullman J D. Principles of database and knowledgebase system. 1990.
- 12 Yao Qingda, Rong Y. Relation data model with cause-effect association. Proc. of 12th VLDB, 1986.
- 13 姚卿达, 纪岳. 数据模型智能化扩充研究(一). 中山大学学报, 1989;28(3).

14 姚卿达, 纪岳. 数据模型智能化扩充研究(二). 计算机科学, 1989(3).

RESEARCH OF INTELLIGENT EXTENSION OF DATA MODELS —AN OBJECT—ORIENTED KNOWLEDGE—DATABASE MODEL

Yao Qingda and Dong Zuming

(*Computer Software Institute, Zhongshan University, Guangzhou 510275*)

Abstract Data model is the foundation of the Intelligent Database Systems. The Research of Intelligent Extension of Data Models is aiming at improving the performance and functions of database systems from the root of data models, so that they can handle knowledge objects, capture more semantics of the application, and have the ability of deduction. In this paper, an Object—Oriented Knowledge—Database Model is presented. The concepts and characteristics, definition and manipulation language frame and architecture of the experimental system OCEANS are discussed in detail.

Key words Database, object—oriented, knowledge—database model.