

决策支持系统 模型管理 面向对象 (12)

71-74

面向对象的模型管理问题

TP11

聂培尧 (山东财政学院 济南250014)

A

摘要

在使用面向对象方法的模型管理中,模型应解释为能根据对“消息”的接收及回答来执行各种模型管理功能的对象的集合。本文讨论了使用结构模型框架对构成模型的对象类进行划分的问题,并对执行各种模型操作所需要的协议进行了描述。本文还就使用一个关系DBMS来实现面向对象的模型管理系统的可行性问题进行了探讨。

一、引言

模型管理是构成决策支持系统(DSS)基础的一个重要研究领域之一。根据Sprague和Carlson^[1]的观点,模型管理的第一步是模型的表示,然后,在此基础上应为模型提供各种操作,如模型的生成、重构、更新以及模型结果的获得等。用于模型管理功能的软件应把数据与DSS的会话子系统紧密地集成在一起。

近几年来,关于模型的表示虽然已提出了很多方法,但这些方法或者是把模型表示成为数据,或者是把模型按知识库中的数据形式表示^[2-4]。这些方法在集成DSS的数据管理子系统方面具有良好的前景。最近,文献[5]中又提出了一种把模型表示成为基于结构建模框架、并具有逻辑结构的关系数据库的思想。本文则主要就如何对这种关系数据库进行操作以执行各种模型管理功能及使用面向对象的方法对模型进行管理等问题进行探讨。

二、结构建模

面向对象程序设计的思想起源于SIMULA。该语言是六十年代中期开发的,其特征是类的概念。它的实例由操作、协同程序和

子类的集合组成。其中操作带局部状态,协同程序通过“resume”操作模拟并行执行,子类继承父类的操作与状态。第一个交互式基于显示的面向对象的设计语言是SMALLTALK^[7]。图1示出了SMALLTALK中对象的某些类和子类。

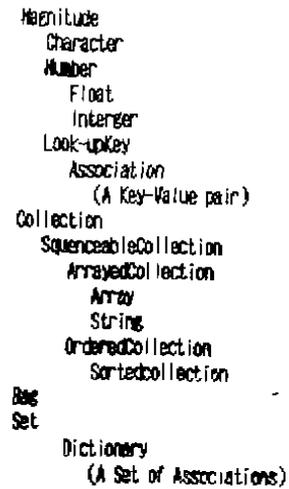


图1 SMALLTALK中的对象

结构建模^[8]是一种描述模型的形式框架,它包含了一种用以描述模型模式(模型类)的语言和用以描述模型实例详细内容的数据库。

下面我们给出了结构模型的一种面向对

收到日期92-1-22。聂培尧 副教授,主要从事数据库系统及MIS、DSS领域的研究。

象的观点，并就把一结构建模表示成为一关系数据库的方法同文献[5]中提出的方法进行了比较。

1. 作为对象的结构建模

根据SMALLTALK的约定，我们在图2中给出了一结构建模中的对象类，其中的对象为元素 (Elements)。元素有三种子类：①实体 (Entities)，包括基本的 (Primitive) 和复合的 (Compound)；②属性 (Attributes)，包括固定的 (Fixed) 和可变的 (Variable)；③函数 (Functions)，包括数值的和逻辑的，后者也称测试元素 (Test Elements)。测试是以函数的子类给出的，表示了它继承了函数的一般特性。实体具有两种子类，即基本实体 (PrimitiveEntity) 及复合实体 (CompoundEntity)，用以表示除了它们所共有的性质外，基本实体及复合实体所具有的特殊性质。另外，除了基本实体外，每一元素都有一个调用时序 (Calling Sequence)，也即，具有一个它们所依赖的元素表。基本依赖 (ElementalDependency) 是联系 (Association) 的子类 (联系是一关键字-值的偶对)。基本依赖有三个子类，即：CompoundOf, AttributeOf和FunctionOf。

Element	Association
Entity	ElementalDependency
PrimitiveEntity	CompoundOf
CompoundEntity	AttributeOf
Attribute	FunctionOf
FixedAttribute	GenericDependence
VariableAttribute	GenericCompoundOf
Function	GenericAttributeOf
Test	GenericFunctionOf
Genus (A set of Elements)	ModularDependence
EntityGenus	ModularCompoundOf
PrimitiveEntityGenus	Dictionary
CompoundEntityGenus	ElementalCallingSequence
AttributeGenus	GenericCallingSequence
FixedAttributeGenus	RootedTree
VariableAttributeGenus	ElementalStructure
FunctionGenus	GenericStructure
TestGenus	ModularStructure
Module (An Array of Genera)	Composite-Object
Compound Module	Structured-Model

图2 结构模型中的对象

在结构建模中，一元素的有序依赖集被称为一基本调用时序 (ElementalCallingSequence)。另外，根据面向对象的观点，一基本调用时序是模型中的一个字典，或者为模型中所有基本依赖联系 (ElementalDependency Associations) 的集合。在结构建模的低层结构中，相同类型的元素，如果它们满足“类属相似”特性，则把它们聚集到一个属 (Genus) 中。也就是说，如果属A (记GENUS A) 中的一个元素是依赖于 (调用) 属B (记GENUS B) 中的某些元素，则 GENUS A 中的每一元素都调用 GENUS B 中的某些元素。这样，存在有若干个属的子类，每个子类均为相应类型的元素集合。相应的，存在联系的子类的类属依赖 (GenericDependency)。该子类记录了 GENUS A 是类属依赖于 GENUS B 的。同样，关于这些联系也存在一个称为类属调用时序 (GenericCallingSequence) 的字典。

在聚集的最高层，若干个相关的属数组称为结构建模的模块。模块是一种数组而不是集合，因为它的元素至少是偏序的。也即，在模块中仅当 $i < j$ 时，第 i 属才可调用第 j 属。一复合模块也为一数组，它的元素则为其它的模块。这就产生了称为模块依赖 (ModularDependence) 的模块间联系类以及一个称为模块内容 (ModularContents) 的联系字典。基本调用时序、类属调用时序及模块内容为三个字典，其中每一个都可表示为一边上带有标记用以指示分支顺序的有根树 (RootedTree)。以上这三种字典分别对应于基本结构 (ElementalStructure)、类属结构 (GenericStructure) 和模块结构 (ModularStructure) 的三种子类。

最后，还有一个作为复合对象子类的对象，称为结构模型 (Structured Model)。一个复合对象包含了一个预先描述的其他类中的对象列表。在这种情况下，一结构模型是由若干个元素、属、模块、字典及相应的有根树所表示的集合。

2. 作为关系的结构模型

在图3中,我们给出了含有结构建模的数据库的若干关系。在基本关系(Elemental Relations)中,我们又分别使用了关系E和A (AREAL、AINT、ACHAR)表示实体类和固定属性类的全部实例。在数据库中,我们还列出了全部可变属性及函数的关系,这是因为有时要对它们进行赋值。同样,在模块关系MODULE中列出了模块类的全部实例。在属关系GENUS中列出了属类的全部实例。在面向对象的术语中,MODULE、GENUS、E、AREAL、AINT及ACHAR的其他属性均被称为实例变量。另外,CONTENT、CALLS及CE关系分别表示了模型内容、类属调用时序和基本调用时序。DATATYPE关系包含了属性属(AttributeGenera)和函数属(FunctionGenera)的实例变量(即数据类型),而RULES仅包含了函数属的实例变量(即函数规则)。

```

Modular Relations
MODULE (ModuleName, Intern)
CONTENT (ModuleName, Contains)

Generic Relations
GENUS (GenusName, Type, Intern)
CALLS (GenusName, Section, CalledGenusName)
RULES (GenusName, Rule)
DATATYPE (GenusName, Dtype)

Elemental Relations
E (GenusName, Index, CalledGenusName, CalledIndex)
CE (GenusName, Index, CalledGenusName, CalledIndex)
AREAL (GenusName, Index, Value)
AINT (GenusName, Index, Value)
ACHAR (GenusName, Index, Value)
    
```

图3 结构建模的关系

面向对象的观点与结构建模关系相互间的联系为:(1)结构建模对象的类模块、属及元素(以及其他的子类)在MODULE、GENUS和基本关系中各自是作为元组说明的,(2)结构建模对象的联系及字典类(以及它们的子类)是在CONTENTS、CALLS和CE关系中被说明的;(3)一结构模型是一复合对象,其实例相应于图3所示的关系数据库的实例。

三、作为消息的模型操作

一个对象,除了类及实例变量外,还应有用来响应消息的方法(Methods)集合。若把一结构建模解释为一个对象的集合,则下一步应把模型操作解释为消息。

通常,消息是按协议方式安排的。协议指明该对象所能接收的消息。在对象内部,每个消息对应一个方法,而方法实施对数据的运算。例如,每个类必须有一个用于创建、修改、或删除其它实例等的操作协议。除此之外,每个集合也必须有一个用于增加及删除元素的协议。

图4示出了SMALLTALK语言中的几个类字典协议。除了用于创建、修改、删除的协议外,还有很多用于存取或测试的协议。这些协议在数据库中称为查询。结构对象应对所有超类(集合、数组或字典)所响应的协议也能响应。另外,还存在一些对高层模型的操作,如由集成或聚集对模型进行重构的操作等,但在SMALLTALK中这些协议不能直接模拟,因此必须把这些功能扩充到结构模型的方法中去。这样,方法中应含有发送给集成模型子系统的多种消息,如测试、存取、创建、修改以及删除等。另一类高层模型的操作是根据对模型的“求值”、“求解”或“优化”(分别相应于“what if”、“what

```

Class Protocols

Creating/Updating/Deleting    ClassVariable
                                ClassDescription
                                MethodDictionary
                                Subclass, Superclass
                                Instance
                                InstanceVariable

Accessing
(NameOf?, ValuesOf,
NumberOf?)

Testing
(ist?...?, Includes?...?, Accesses?...?)

Set Protocols
Adding/Deleting an Element
Testing (NumberOfElement)

Dictionary Protocols
Adding/Deleting an Association (Key-Valued Pair)
Accessing (HasKey?, [aKey?])
Testing (Includes: [aValue?])
    
```

图4 SMALLTALK中的协议

works”及“what's best”方法来对可变属性及函数进行赋值。我们称这类模型操作是“赋值的”。例如，对一模型“优化”意味着对所有可变属性进行赋值以使每一个测试的值为“True”并使某些函数为最大或最小。这一般是靠调用一数值过程，如线性规划中的Simplex算法等来完成的。

```

Create/Update/Delete an AttributeGenus
Name: [anIdentifier]
Interpretation: [aString]
Dtype: [aDomain]

Create/Update/Delete GenericAttributeOf
Association
Attribute Genus: [anAttributeGenus]
Is Generic Attribute of: [anEntityGenus]

Create/Update/Delete an AttributeElement
Elementof: [anAttributeGenus]
Attributeof: [anEntityElementIndex]
Value: [aValue]

Accessing AttributeGenus
WhatIs: Interpretation? Dtype?
NumberOf: Elements?

Accessing AttributeElement
WhatIs: Value

```

图5 “Attribute Genus”类的协议

四、实现方面的问题

上面所讨论的面向对象方法的模型管理问题在其具体实现上，一个可能的方法是使用面向对象语言来建造一模型管理系统。但是需要指出的是，为了把数据与会话子系统集成在一起，整个DSS应按相同的方法进行编码。另一种方法是对关系数据模型进行扩充^[9]。在这类方法中，有一种则是采用了关系数据库中事务的面向对象的观点^[10]。我们说对于一关系模型管理系统的实现也可考虑沿用这些方法中的某些思想。

使用关系DBMS的方法则是把“对象”看作是关系、属性、域、码及元组，并能提供查询语言中所定义的全部操作（方法）、数据定义语言DDL（如对关系的创建、删除及属性定义等）及数据操纵语言DML（如对元组的插入、修改、删除等）。另外，大多数DB-

MS提供了把若干对象基本类作为数字及字符进行存取的功能。对于新对象的定义可根据图2所示的模式定义之。

五、结束语

由上面的讨论可以得到这样一个结论，即面向对象的程序设计方法用于模型管理是合适的。并且，根据结构模型框架面向对象的观点及作为关系数据库表示的观点，把模型管理系统作为一关系DBMS来实现是可行的。

参考文献

- [1] Sprague, R.H. Jr.等, Building Effective Decision Support Systems, Prentice-Hall, Inc., 1982
- [2] Dolk, D.R等, Knowledge Representation for Model Management Systems, IEEE TODS, SE-10:6, 1984
- [3] Elam, J.J.等, Knowledge Engineering Concepts for Decision Support System Design and Implementation, Proc. of Fourteenth Annual Hawaii Int'l Conf. on System Science, North Hollywood, CA, 1980
- [4] Fedorowicz, J.等, Representing Modeling Knowledge in an Intelligent Decision Support System, 1986
- [5] Lenard, M. L., Representing Models as Data, Journal of Management Information Systems, 2:4, 1986
- [6] 彭智勇, 面向对象的规范描述及转换, 计算机科学, No.1, 1990
- [7] Goldberg, A.等, Smalltalk-80, Addison-Wesley, MA
- [8] Geoffrion, A.M., Structured Modeling, Western Management Science Institute, UCLA, CA, 1985
- [9] Brodie, M.L., On the Development of Data Models, In: On Conceptual Modeling, ed. Brodie, M.L.等, Springer-Verlag, 1984
- [10] Brodie, M.L.等, On the Design and Specification of Database Transactions, 同上