第2卷第2期
1999年6月

管 理 科 学 学 报
JOURNAL OF MANAGEMENT SCIENCES IN CHINA

Vol. 2 No. 2
Jun. 1999

研究简报

面向对象的模型生成与链接

林 杰[1]　　郭耀煌
（复旦大学管理学院）　（西南交通大学经济管理学院）

【摘要】本文用 Smalltalk 语言完成了模型的生成与链接. 首先建立了各种基本原子模型类，如线性规划模型类、运输模型类等，再用原子模型类生成实例对象，从而产生各种具体的原子模型. 在通过模型链接组建复合模型时，提供了一个复合模型类 Commodel 来完成子模型运行、链接等功能. 最后，用实例进一步说明了其使用法，并证明此方法可行和有独特的优点.

关键词：模型，模型链接，决策支持系统，面向对象设计
分类号：C934

0 引 言

DSS 这一概念于 70 年代首先由 Gorry 和 Morton 提出，它是以解决半结构化和非结构化问题为目的，便于模型和数据管理的交互式计算机系统，用以提供有效的决策支持. 在 DSS 中，模型处于十分重要的地位.

经过近三十年的研究，模型库管理系统随 DSS 的发展有了长足的进展，出现了各种模型表示法和模型库管理系统. 但是，现有模型库管理系统的模型生成与链接仍不完善，如模型的多样性，模型与方法、模型与数据的特性匹配，模型与数据的独立性等问题都有待进一步解决.

面向对象分析与设计技术在当今比较盛行，经过近些年来的研究应用，不管在理论上还是在技术上都已经比较成熟，它在系统分析、系统设计等许多方面都已显示其成功之处. 实践证明，把面向对象的概念和技术应用于模型管理系统设计非常适合，它能有效解决模型管理系统存在的许多问题[1,2].

1 模型抽象

模型是以某种形式对一个系统的本质属性的描述，以刻画系统的功能、行为及其变化规律. 人们为了有效管理、生成模型和科学描述客观现实世界，引入了复合模型、子模型、原子模型三个范畴. 复合模型由多个子模型组成，而子模型又可分解成几个更小的模型(子子模型). 子模型可以是原子模型，也可以是复合模型. 因此可以把模型分成两类：原子模型和复合模型.

决策者利用模型输出信息以制定决策和控制过程进展，他们对结果及其运用效果感兴趣，但大都对模型工作原理了解不深或相对不感兴趣. 从使用角度来看，模型也可视为一个反映输入与输出关系的黑盒[1]，如图 1(a)所示. 也就是说可把模型比喻成一块集成电路，用户只要对集成电路的引脚和功能了解清楚就能应用，而其内部结构的设计和生产则由集成电路的设计者和生产厂家来完成. 为此、模型可抽

① 林 杰：博士. 通讯地址：上海复旦大学管理学院管理科学系. 邮编：200433.
本文 1998 年 1 月 19 日收到.

象成：外部封装(输入、输出变量)与由知识工程师完成的内部结构，如图 1(b)所示；同时，为模型的正确使用与管理，还需要配备模型说明字典(模型字典)．这样就可把模型视为独立对象，按面向对象的分析与设计思想对其管理与操作．本文用 Model 类来建立模型的外部封装与模型字典框架，用其子类——各种模型模板类建立模型的内部结构，用模型模板生成的实例形成各种具体模型(本文选用 Smalltalk[4]语言进行分析)．

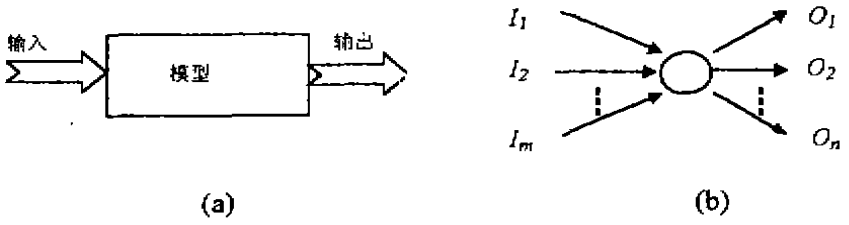

图 1 模型外部视图

2 模型类

犹如所有的集成电路都有引脚、内部结构和使用说明一样，所有模型都有外部封装、内部结构及模型字典．尽管模型的内部结构千差万别，但其外部封装、模型字典都有相似的结构．因此引出 Model 类，其属性和操作方法确定模型的输入、输出变量，输入数据的来源，输出数据的去处，及建立模型字典框架等．给出 Model 类如下：

```
Object subclass #Model
    instanceVariableNames:
      'name inputstream outstream inputvariable outvariable indatatype outdatatype
            modeldictionary '
      classVariableName:''
      pooldictionaries:''
```

其中 Object 为 Smalltalk 中所有类的超类，Model 类作为其子类．Model 的有名实例变量有：name，存放模型名；inputstream(输入流)，指明输入数据来源(键盘、数据库文件或其它设备)；outstream(输出流)，指明输出数据去处(显示器、磁盘文件或打印机等)；inputvariable(输入变量集)，存放模型的输入变量；outvariable(输出变量集)，存放模型的输出变量；modeldictionary(模型字典)，是一个字典变量，其键和值如表 1 所示；indatatype，指明输入数据格式；outdatatype，说明输出数据格式．

表 1 模型字典

键 名	值	键 名	值
functf	功能描述	modeldate	模型建立日期
riable	模型可靠程度	builder	建模者名称
inputvarde	输入变量描述	modeltype	模型类型
outvarde	输出变量描述	…	…

对应的一些过程和方法(methods)有：

name ： aString

把 aString 放入 name 中，aString 是一个模型名称的字符串．

inputstream : aStream

把一个流 aStream(输入路径或设备)存放在接收者的输入流 inputstream 中.

outstream : aStream

把一个流 aStream(输出路经或设备)存放在接收者的输出流 outstream 中.

inputvariable : aSymbol to : aVariable

把接受者(模型)的输入端 aSymbol 定义成变量 aVariable,并存放在输入变量集 inputvariable 中. aSymbol 为模型输入端编号(I_1,I_2…),aVariable 为模型输入变量.

outputvariable : aSymbol to : aVariable

把接受者(模型)的输出端 aSymbol 定义成变量 aVariable,并存放在输出变量集 outvariable 中. aSymbol 为模型输出端编号(O_1,O_2…),aVariable 为模型输出变量.

at : akey put : anObject

回答 anObject. 如果接收者(模型)的 modeldictionary 字典包含键为 akey 的键/值对,则用 anObject 替换该键/值对的值. 否则增加 akey/anObject 对.

remove

把接受者从模型管理系统的模型目录库中删除,并删除代表模型的全局变量. 接受者为一模型模板实例.

removeclass : anObject

把模型模板 anObject 删除,如果此模板有实例模型,则报告出错. 此消息发给 anObject 的超类.

save

把接受者存放到模型管理系统的模型目录库中去. 接受者为一个代表模型(模型模板实例)的全局变量.

name

返回接受者的模型名,即把模型有名实例变量 name 的值返回.

deleteat: aKey

从 modeldictionary 中删除包含键 aKey 的键/值对,无此键报告出错.

…

3　原子模型

3.1　原子模型模板

不同模型的内部结构、操作方法、端点定义各不相同,这种差异决定了模型有不同的功能,本文采用模型模板来描述这种差异,即一种模型模板定义了该模型类独有的数据结构、内部变量、运行方法等.

模型模板分为原子模型模板、复合模型模板. 本节只讨论原子模型模板的建立.

设某一分析领域需要建立三类模型(这里不针对某一真实领域,只用来说明建立方法):一般线性规划模型、运输问题模型、神经网络模型. 为三类模型建立三个模型模板:Linearpro(线性规划模板)、Transportation(运输问题模板)、ANN(神经网络预测模型模板). 建立类结构图 2 所示.

运输问题可用一般线性规划的数学模型描述,也可用其求解方法求解[5],如单纯形法,因此把运输问题模型模板作为线性规划模型模板的子类,以继承其描述形式及求解算法. 同时,也在自己范围内建立更适合自己的描述形式及求解算法,如表上作业法等.

神经网络模型(ANN)无法继承一般线性规划及其子类的描述及求解方法. 但它也有外部封装及描述字典,因此,ANN 为 Model 的子类. 同时,在其内部建立自己的结构及相应算法完成 ANN 模板描述.

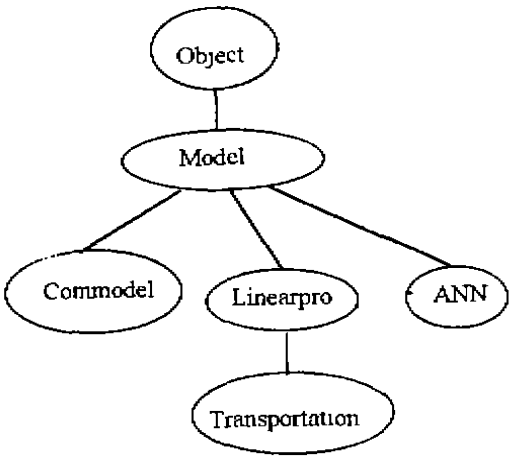

图 2 类结构

下面将建立一般线性规划问题模型模板. 线性规划的数学模型表示如下:

$$\begin{cases} \max(\min) z = \sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij} x_j \leqslant (=, \geqslant) b_i \quad i = 1, 2, \cdots, m \\ x_j \geqslant 0 \quad j = 1, 2, \cdots, n \end{cases} \tag{1}$$

对类 Linearpro 建立以下有名实例变量:

varm , varn , maxormin, cstream, astream, bstream , xstream , constraint , operations

其中 varm、varn 分别存放式(1)中 m、n 的值. maxormin 指明模型为极大或极小化问题. constraint 存放各约束方程的约束条件,可为"≥"、"="、"≤"之一. cstream、astream、bstream 分别指向存放参数 c_j、a_{ij}、b_i 的流,各流可以指向键盘、数据库文件或其他设备. operations(一个字典),其存放模型的可用算法(单纯形法、对偶单纯形法等)的名称及存放这些算法的路径和文件名.

对其建立类方法(class methodes):

new

新建立一个 Linearpro 实例并初始化.

addoperations:anObject and:aStream

向接受者的算法字典 operations 中加入算法 anObject 和指向算法程序所在位置的路径 aStream.

deleoperations : anObject

把算法 anObject 从接受者的算法字典中删除.

…

实例方法(intance methods):

operations : akey

用键 akey 的值所指算法完成对接受者运算. 接受者为一个一般线性规划问题. akey 为算法字典中的一个键,其值是算法的路径和文件名.

putm : anInteger1 andn : anInteger2

把 anInteger1、anInteger2 分别置于有名实例变量 varm、varn 中.

cstream : aStream

把 aStream 存放在 cstream 中,aStream 是读取参数 c_j 的路径及文件名.

astream : aStream

把 aStream 存放在 astream 中,aStream 是读取参数 a_{ij} 的路径及文件名.

…

3.2 原子模型生成及使用

下面以一个实例介绍原子模型的建立过程.

设某厂生产甲乙两种产品,已知制成一吨产品甲需用资源 A 3 吨,资源 B 4m³;制成 1 吨产品乙需用资源 A 2 吨,资源 B 6m³,资源 C 7 个单位,一吨产品甲和乙的经济价值分别为 7 万元和 5 万元,三种资源的限制量分别为 b_1 吨、b_2 m³、b_3 个单位,试建立一个运算模型:输入为限制量 b_1、b_2、b_3,输出为生产产品甲、乙的产量,要求使总经济价值最高.

生成和运行原子模型过程如下:

①定义一个线性规划模型变量 Prodecision

Prodecision := Linearpro new

②给 Prodecision 命名"生产决策"

Prodecision name: '生产决策'

③确定模型的输入变量

Prodecision inputvariable: I_1 to: b_1

...

④确定模型的输出变量

Prodecision outputvariable: O_1 to: x_1

Prodecision outputvariable: O_2 to: x_2

Prodecision outputvariable: O_3 to: max z

⑤建立模型字典内容

Prodecision at: functf put: thestring "thestring 为模型功能描述字符串."

...

⑥建立模型 m、n 的值、约束方程的约束条件等参数.

Prodecision putm: 2 andn: 3

...

⑦给出各个参数存放的路径或文件.

Prodecision cstream: file1 "文件 file1 存放参数 C=(3.2)"

Prodecision astream: file2

"文件 file2 存放参数 $A=\begin{bmatrix}3 & 2\\ 4 & 6\\ 0 & 7\end{bmatrix}$"

⑧测试模型 Prodecision

⑨把 Prodecision 存放到模型管理系统的模型目录数据库中.

Prodecision save

⑩调用、运行模型

指明模型输入数据来源.

确定模型输出文件及路径.

运行模型.

4　复合模型

4.1　复合模型类

现实问题仅仅靠原子模型就能解决的并不多,往往需要几个模型联合求解,这就要求模型管理系统对模型的层次开发和重用提供支持.即把被验证是正确的求解模型不加修改就可以连续运行,作为积木块构造功能更强大的求解模型——复合模型.复合模型的外部视图与原子模型相似,因此,可以定义复合模型类 Commodel 作为 Model 的子类,定义如下:

```
Model subclass #Commodel
    instanceVariableNames:
      'logicstruct interface'
```

classVariableName:''

pooldictionaries:''

对复合模型的内部描述有两部分:逻辑部分和接口部分[3]. 逻辑部分存放复合模型中所有子模型的名称及其运行顺序,接口部分存放各子模型接口连接表. 存放逻辑部分、接口部分的实例变量分别为 logicstruct、interface. logicstruct 为 Array(数组)实例;interface 为 Dictionary(字典),其键为模型名,值为一个 Set(集合),集合收集了该模型所有连接方式.

Commodel 的类方法(class methods)

new: anInteger

logicstruct := Array new: (anInteger−1) "logicstruct 设置为数组(Array),其中存放复合模型中有子模型的数量和子模型的运行顺序."

interface := Dictionary new: anInteger "把 interface 设置为一个字典."

logicstruct at: 1 put: anInteger "数组的第一个元素为复合模型中子模型的数量."

...

实例方法(instance methods)

logicstructat: anInteger Put: aString

anInteger 为大于 1 但小于或等于复合模型调用子模型数量的一个正整数,aString 为一个模型名. 此方法把子模型 aString 放入 logicstruct 数组中,同时在字典 interface 中建立以模型名 aString 为键的键/值对,值设置成一个 Set 收集. 注意:logicstruct 中的先后顺序对应复合模型中子模型的运行顺序.

logicstruct: anInteger

把数组 logicstruct 中序号位置为 1 的对象换为 anInteger, anInteger 为复合模型中有子模型的数量.

logicstructat: anInteger

返回 logicstruct 中序号为 anInteger 的对象. anInteger 为 1 时返回复合模型中子模型的数量,大于 1 时返回子模型名.

interface: aString1 to: aString2

把 aString2 放入到字典 interface 的键为 aString1 的收集中去. aString1 为子模型名,aString2 为与子模型 aString1 的输入、输出的一个链接.

interface: aString1 find: aString2

在 interface 字典键为 aString1 收集中查找链接 aString2,有返回 True,无返回 False.

...

4.2 复合模型实例

对图 3 复合模型建立如下(设在系统中子模型 A、B、C、D 已经存在):

Acommodel := Commodel new: 4 "Acommodel 是代表图 3 所示复合模型的全局变量"

Acommodel logistruct: 2 put: 'A'

Acommodel logistruct: 3 put: 'C'

Acommodel logistruct: 4 put: 'B'

Acommodel logistruct: 5 put: 'D'

Acommodel interface: 'A' to 'A. I1−I1'

Acommodel interface: 'A' to 'A. I2−I2'

Acommodel interface: 'A' to 'A. O1−B. I1'

Acommodel interface: 'A' to 'A. O2−B. I2'

Acommodel interface: 'A' to 'A.O3—C.I3'
Acommodel interface: 'B' to 'B.I1—A.O1'
...

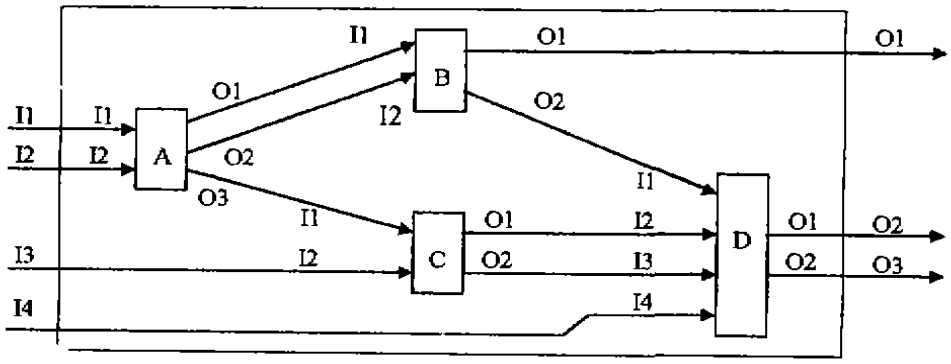

图 3　复合模型实例

此时,图 3 复合模型链接已建立. Acommodel 中的 logicstruct 和 interface 的内容如图 4 所示.

logicstruct:
(表示模型调用顺序为 A、C、B、D)

4
A
C
B
D

interface 内容为:

链	值 (Set)
A	A.I1-I1　A.I2-I2　A.O1-B.I1　A.O2-B.I2　A.O3-C.I1
B	B.I1-A.O1　B.I2-A.O2　B.O1-O1　B.O2-D.I1
C	C.I1-A.O3　C.I2-I3　C.O1-D.I2　C.O2-D.I3
D	D.I1-B.O2　D.I2-C.O1　D.I3-C.O2　D.I4-I4 D.O1-O2　D.O2-O3

图 4　Acommodel 实例变量的内容

复合模型 Acommodel 运行时,按 logicstruct 所指顺序调运子模型 A、C、B、D. 由于各子模型的输入、输出都是流,因此实现数据对接时,按 interface 内容把各子模型相链接变量指向数据库或内存区的同一处或同一内存变量,自然实现模型运行数据对接.

5　结　论

通过以上分析,在已有原子模型类的情况下,建立各种具体的原子模型是相当容易的,且其过程可程序化. 原子模型类的建立虽然较模型模板生成模型复杂,但也比传统方法易于实现. 使用本文提出的方法在子模型链接组成复合模型时,若系统的界面友好,如让用户用表格或图形的方式表示子模型的链接关系,系统可自动进行复合模型中子模型的调用、数据对接和运行,实现复合模型的自动组建. 若系统配有专家咨询系统或智能引导系统,引导用户完成模型的建立与链接,系统很适合管理知识丰富而计算机知识相对薄弱的管理人员使用.

参 考 文 献

1　Lenard. An object-oriented approach to model management. Proc. 20th Hawaii International Conference on System Science, Kana-kailua, HW: Computer Society Press, 1987; 1: 509～515

2　聂培尧. 面向对象的模型管理问题. 计算机科学, 1993; 6: 71～74

3 张学凤，陈明裕，陈永年. 决策支持系统中的模型及模型库管理系统. 计算机研究与发展，1993；3：36～41
4 夏晓东，宋杰，刘柏译. 面向对象的程序设计—Smalltalk 语言及环境. 北京：北京航空航天大学出版社，1990
5 郭耀煌等. 运筹学原理与方法. 成都：西南交通大学出版社，1994. 78～97

An Object-Oriented Approach to Model Generation and Coupling

Lin Jie
School of Management，Fudan University
Guo Yaohuang
School of Economic and Management，Southwest Jiaotong University

Abstract The paper analyses model generation and coupling with Smalltalk language. After creating various atom model classes，such as linear programming class，transportation model class，which are instances of atom model classes，model coupling is used to create composite model. In the end，the method is proved feasible by the examples and has some unique advantages.

Keywords： model，model coupling，decision support system，object-oriented design