

# DSS中的模型表示与模型库

于晓迪 (清华大学经管学院)

## 摘 要

有关模型表示及模型库的研究,是当前DSS、MS/OR等领域中十分活跃的研究课题。本文讨论了模型库应具有的基本特性及其一般结构,给出了一种结构模型表示方法,并讨论了该方法与关系数据库的关系。作者认为,这是一种较为实用的模型表示方法。

## 一、问题的背景

有关模型表示及模型库(MB)的研究,是当前DSS、MS/OR等领域中十分活跃的研究课题。它的提出有着其深刻的背景:

### 1. MB在DSS中的重要性:

·模型是分析问题,提供决策方案的基础,将模型移入信息系统,导致了传统的MIS发展成DSS,仅有足够的数据库,决策人员的决策活动只能依赖于查询、分类及归纳总结各类数据,这便是传统的MIS的局限性。根据Simon的模式,它仅仅支持了决策活动中的第一阶段,

提供帮助,利用属于我们的软件系统,但让他们以自己认为是最有价值的方式去使用”。

在DSS小组最希望影响哪些经理和哪些决策的意义上讲,还没有明确的目标市场。反映扩展决策支持并且是切合实际的用于行动的定义或许能描述为:我们的作用是在关键的计划和决策方面帮助高级的经营管理者,运用专家经验和建立计算机系统能够对他们最主要的经营目标做出重大的和直接的贡献。

这一定义显然排除了那些相对简单的支持对象,例如象电子报表程序和即席模型化这样的生计简易市场。也许这一断言对某个具体的DSS小组太苛刻了,他们可能这样标榜自己:“与高层的市场和综合计划经理一

·决策环境是经常变化的,随着环境的变化,相应的模型要作适当的调整——重构,以适应这种变化。因而,DSS应提供一种统一有效、方便灵活的管理模型的方法。

·DSS应提供利用多个模型的组合来解决复杂问题的手段。

·DSS应支持建模活动。由于问题的复杂性,其模型的建立可能要经过多次反复,后面的结果依赖于前一次建立的模型。

### 2. 传统的模型使用中的问题

·每一个模型至少有三种表示,适于没有受过MS/OR等专门训练的人(通常是经理一类的决策人员)的“自然”表示、适于分析人员使用的数学

同工作,弄清在哪些方面能够为他们建立针对性的分析和模型化能力,以用于竞争和环境的观测”。

若不考虑特殊的方面,一般看法认为,大多数DSS小组没有明确的服务对象,他们不知道积极支持哪些管理者和决策,以及他们对专家和计算机方面的投资能够给公司提供多少利润。同样,许多DSS研究也是在被默认为是传统的甚至是被动的支持水平上进行的,因为那是行业的实际状况。这可能属实,但研究的目标应当为行业提供更多的抉择机会。

(未完待续)

[颜永璞译自Decision Support System  
3(1987), 253—265 萨师煊校]

表示及适于在计算机上求解的表示。这种冗余表示造成低效率，也常常引起不一致性，且完成一个问题的求解需要多方面的技巧。

- 通常很难为非专业人员所理解，是以技术(算法)为中心，而不是以问题为中心。

- 计算机上的软件实现通常是模型与求解算法不独立，模型稍有变化，已有的软件很难适应这种变化。

## 二、答案：新一代模型库系统

数据库系统的巨大成功，特别是DBMS的特性给了人们以很大的启发。具有与数据库系统许多类似特性的新一代模型库系统是解决上述问题的有效途径之一。它应具有如下的特性：

1. 适于与管理人员通讯、分析人员使用及计算机上直接运行的统一表示形式。

2. 模型表示与模型求解算法的相互独立性。支持构造模型，调要优化、模拟、推理等求解器的标准界面。

3. 能有效地适应多种模型范畴，如行为分析、决策树、流网、图、马尔科夫链、排队系统等。

4. 支持模型生命周期的各个阶段。

5. 一般的模型结构表示(抽象模型)与具有详细的参数数据值的特殊模型实例(具体模型)描述的相互独立性。

6. 具有友好的用户界面，支持用户对模型的多视图。

7. 象DBMS管理数据那样管理模型。

8. 基于知识。

这些特性是设计模型库时的基本原则。从模型库建造者的角度来讲，模型库的建造主要解决两个方面的问题：

- 模型的存贮功能，它包括：模型的表示，模型的逻辑存贮，模型的物理存贮。

- 对模型的操作，主要有：抽象模型的实例化，选择一个模型，模型的组合，模型的生成，模型的重构，模型的分解，模型的求解。

对模型的操作依赖于其表示形式。因而模型库建造的首要问题是模型的表示问题，

且其表示形式应支持前述模型库的八个特性。

## 三、结构模型表示法及其特性

### 1. 结构模型表示法

由前述的讨论，模型表示应支持：算法独立，多级抽象，DBMS兼容性及基于知识。结构模型表示法思想源于程序设计语言中发展出的抽象数据类型的思想。一个抽象数据类型由三部分组成：数据对象，数据对象上允许的操作，规则；某些操作在某些条件下不允许等等。

这种表示方法不孤立地看待数据，而是将其视为某种意义上的代数结构。其优点是：

1. 提供看待数据结构的统一方法，
2. 能方便地由较低级的抽象构造更高级的抽象，
3. 一旦定义，能被所有的程序员一致使用。这些优点对于模型环境也同样十分有用的。

结构模型表示法将模型视为由以下三部分组成的抽象结构：

元素结构，

对该模型所允许的操作，及

规则；操作之间的关系。

元素结构描述模型所针对的问题，它由元素及其关系构成，这一部分适于决策人员理解，而后两部分分别适于分析人员使用及理解计算机上的执行过程。三部分逻辑上可视为一个整体，物理上又可分开处理。

下面我们用传统的运输问题来非形式地给出这种模型表示方法。运输问题可叙述如下：有 $m$ 个供应点，供应某种商品，且知各点的供应量，有 $n$ 个需求点，需要该种商品，且知各点的需求量。已知每个供应点到需求点的单位商品运费，在一般供需平衡的条件下，求每个供应点运到需求点的商品数量，使总的运费支出最小。

元素结构中有7类基本元素：

- 原始实体元素(PR)，通常表示与一模型相关的对象或概念；

• 复合实体元素 (CO)，通常表示用其它的对象或概念来定义的对象或概念；

• 常值属性元素 (A) } 表示实体的属性；  
• 变值属性元素 (VA) }

• 取值为数值的函数元素 (F)；

• 取值为真假的函数元素 (T)；

• 相关元素 (RE)，即与其它模型结构相关的元素。

对于运输问题，供应点和需求点都是原始实体元素；两者之间的联结是复合实体元素；（即，每一个联结可以用一个供应点和一个需求点的组合来定义），与每一联接相关的运输费用是一常值属性元素，而其运输的商品数量则是一变值属性元素；总运输费用是属性的一取值为数值的函数元素；而检查是否满足了需求点的要求则是属性的一取值为“真假”的函数。

这些元素间不是孤立而是相互关联的。我们定义一种称之为“引用”的关系。除原始实体元素之外，每一个元素都有一个“引用序列”，或称“依赖元素集”。例如，运输联接依赖于（引用）它所联接的供应点和需求点；运输费用依赖于它所对应的联接；总运输费用依赖于两组属性元素：每一联接的运输费用及在每一联接上的运输数量。

所有这些元素可按所谓的“类相似性”分成一些子集，每个子集中的元素具有相同的元素类型。而所谓“类相似性”是指，如果子集A中有一元素引用子集B中的一个或多个元素，则A中的所有元素必须都至少引用B中的一个元素。对于运输问题可分成这样几个子集：供应点集 (SUP)，供应点供给的商品数量组成一常值属性集 (AMOUNT)；需求点集 (CUST)；需求量集 (DEN)；运输联接线集 (LINK)；每一联接上的运输量组成的流量集 (FLOW)；每一联接线上单位商品运输费用组成的运输成本集 (COST)；总成本集 (TOT)；测试是否超过供应量的函数集 (T: SUP)；测试是否满足需求量的函数集 (T: DEN)。

下面我们来看看结构模型表示法三部分的具体构成：

**元素结构** 由按“类相似性”分成的元素子集，及它们间的引用关系组成。运输问题的元素结构可用图表示如下：

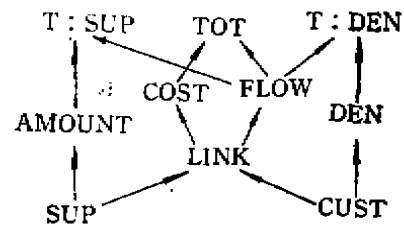


图1

可为元素结构设计一种正文表示形式。如果除掉每个集中元素的细节，上图则表示了一类（抽象的）运输问题。若给出具体的元素细节表，便构成一个具体的运输问题。对该问题进行优化操作，便是一线性规划问题。而象上述那样直观的图形表示，管理决策级的人员是很易理解的。

**模型操作** 用一阶谓词形式表示，每一个操作作用一谓词来表示。对运输问题可允许的操作有：

#### Manipulation

Solve (lp-matrix)

•Solve matrix representation

Trans1 (lp-eqn, lp-matrix)

•Transform lp-eqn to matrix representation

Trans2 (lp-eqn, graphic)

•Transform lp-eqn to graphic representation

Extra (lp-matrix, xj)

•Extract decision variables from matrix

Linear (eqn, decision-variable)

•Is equation linear in decision-variables?

Trans3 (graphic, lp-eqn)

•Transform graphic to lp-eqn repr

End-of-Manipulation

所列的几种模型表示形式的转换操作，支持用户对一个模型的多种视图。

**规则** 也可用一阶谓词形式表示, 下面是一简单的描述例子:

#### Rulers

1. Linear (TOT, decision-variables)
2. Trans1(TOT, Ip-matrix) & Trans1 (T: sup, Ip-matrix) & Trans1 (T: DEN, Ip-matrix).
3. Solve (min (TOT) ) & subject to (T: SUP) & Subject to (T: DEN)

#### End-of-Rulers

### 2. 结构模型表示法的特性

这一节将考察结构模型表示法作为新一代模型库系统的基础的适用性。依据前述的八个特性来讨论。

1) **统一的表示形式** 结构模型表示方法将模型视为由元素结构及相应的操作等所构成的抽象结构。元素结构的图形表示适于与管理决策人员通讯; 而其正文形式表示及操作适于分析人员使用; 利用关于模型的知识及发展元素结构正文表示形式的编译程序, 便可方便地在计算机上直接执行。

2) **模型与其求解算法的独立性** 模型表示与对它所允许的操作在表示上都是分离的, 而不是象传统的那样统一在软件程序中。模型可放在模型库中, 类似求解算法可放在方法库中。

3) **通用性** 现有的研究表明, 它对于MS/OR中的许多问题是适用的。对于其它模型范畴, 如系统模拟、排队系统、流网等的适用性还有待进一步的讨论。

4) **支持模型生命周期的各个阶段** 其大多数支持依赖于模型管理系统的功能, 就模型表示这一方面来说, 体现在模型表示的可修改性, 这可通过开发相应的编辑程序来达到。

5) **抽象模型与具体的模型实例的相互独立性** 抽象模型由元素结构描述, 而元素细节表给出具体的模型实例。

6) **支持用户对模型的各种视图** 可开发各种模型表示形式间的转换算子。如, 元素结构的图形表示, 正文表示, 方程形式, 矩阵形式等等。

7) **象DBMS管理数据那样管理模型** 后续的讨论将表明, 元素结构可用关系来实现, 因而可极大的利用关系数据库管理系统的功能来管理模型。

8) **基于知识** 可将规则部分进一步扩完成描述与模型相关的知识的“微型”知识库。它包括三方

面的知识: 领域的, 模型本身的, 如, 模型是一决策变量的线性方程; 与求解过程相关的, 如求线性规划问题时, 最终要将其转换成矩阵表示形式等等。这些知识既有助于建模活动, 也有助于决策人员理解模型。

因此, 新一代模型库系统可通过实现基于结构模型表示法的模型库来达到。

### 3. 元素结构的关系表示

我们同样用运输问题来说明怎样用关系方法来实现元素结构。

一个元素结构可用7类关系来表示:

**Relation1: Elementstruc Relation.** 有两个域, Name和Interp(otation), 给出元素结构的概貌。Name是关键字。

**Relation2: Subset Relation.** 有三个域, Name, Type和Interp. 分别给出元素中所有子集的名字, 子集中元素的类型和对元素的一些解释。Name是关键字。

**Relation3: References Relation.** 有三个域, Name, Order, 及Refed, 分别给出引用者名; 引用序列编号及被引用者。前两个域是关键字。每个子集有一下标集, 有的子集的下标可由另外的子集决定。例如, 每个需求点都有需求量, 我们把这种关系也在References中表示出来, 只不过其Order域值为0。

**Relation4: Equations Relation.** 有两个域, Name和Equation, 给出函数元素 (T类或F类) 的求值规则。Name域是关键字。

**Relation5: list Relation.** 有三个域, Name, Index, Rename, 分别给出PR类和CO类元素所在子集的名, 它在子集中的编号 (下标), 及其实际名字。Rename域是关键字。

**Relation6: Datatype Relation.** 有两个域 Name和Datatype, 给出属性类子集的名字及其元素的数据类型, Datatype分为两部分, 第一个字符R、I、C, 分别表示实数, 整数, 字符, 而后面的数字则表示存储位数。

**Relation7: ValueR Relation.** 有三个域, Name, Index, Value, 分别给出属性元素的子集名, 该元素在子集中的下标及其属性值。(这类关系一般有三种, 分别对应整数(I)、实数(R), 及字符(C))。

下面给出运输问题的详细的关系描述:

Table 1 Elementstruc Relation

Name	Interp
Transp	A transportation Problem
SUPI	Information about suppliers
CUSI	Information about customers
TransI	Information about transportation

Table 2 Subset Relation

Name	Type	Interp
SUP	PR	A list of suppliers
CUST	PR	A list of customers
LINK	CO	A transportation link between supplier and customer
COST	A	Unit cost of shipment on each link
FLOW	VA	Amount shipped on each link
DEN	A	Amounts required by cust
AMOUNT	A	Amounts supplied by sup
TOT	F	Total cost of shipment on all links
T: SUP	T	Test of shipment not exceeding supply
T: DEN	T	Test of shipment meeting demand

Table 3 References Relation

Name	order	Refed
TOT	1	COST
TOT	2	FLOW
LINK	1	SUP
LINK	2	CUST
T: SUP	1	AMOUNT
T: DEN	1	DEN
COST	1	LINK
FLOW	1	LINK
AMOUNT	0	SUP
DEN	0	CUST

不难看出, 如果开发出元素结构正文表示形式的编译程序, 编译结果为上述关系表示, 再以关系表示为基础, 生成模型的各种视图, 如图形的, 方程的等等, 是并不困难

的。

Table 4 Equation Relation

Name	Equation
TOT	$SUM_i(SUM_j(COST_{ij} * Flow_{ij}))$
T: DEN	$SUM_i(Flow_{ij}) = DEN_i$
T: SUP	$SUM_j(Flow_{ij}) \leq AMOUNT_j$

Table 5 List Relation

Name	Index	Rename
SUP	1	Tsinghua
SUP	2	Eeta
CUST	1	Wang
CUST	2	Liu
CUST	3	Zhang
LINK	1	Tsinghua—Wang
LINK	2	Tsinghua—Liu
LINK	3	Tsinghua—Zhang
LINK	4	Eeta—Wang
LINK	5	Eeta—Liu
LINK	6	Eeta—Zhang

Table 6 Datatype Relation

Name	Datatype
DEN	R16
AMOUNT	R32
FLOW	R16
COST	R8

Table 7 ValueR Relation

Name	Index	Value
DEN	1	200
DEN	2	300
DEN	3	400
AMOUNT	1	500
AMOUNT	2	800
COST	1	5
COST	2	6
COST	3	7
COST	4	8
COST	5	9
COST	6	10

#### 4. 模型库系统中的组合问题

模型库系统中至少有三类组合问题:

- 模型组合(model integration)
- 操作组合(manipulation integration)
- 实用程序组合(utility integration)

后两者更多的是属于软件模块的组合问题, 依赖于特定的问题及所采用的软件设计技术, 是实现模型库系统时应考虑的因素之一。

模型组合: 结构模型表示方法本质上是用“分解”的思想来看待模型, 而不是象方程表示那样将模型视为一不可分割的整体。按对应的元素类来进行组合, 形成新的元素类, 从而形成新的模型。因此, 从原则上来说, 该表示方法是适于实现“自然”的模型组合的。

目前, 从直观上来看, 利用相关元素类实现模型的“顺序组合”, 即一模型的输出结果是另一模型的输入参数(或说一模型的运行依赖于另一模型的运行是不困难的)。

进一步的讨论可从如下的几个层次来考虑:

- 具体模型实例间的组合(具有元素细节表)
- 模型结构间的组合(同一模型范畴中的抽象模型间的组合)
- 不同模型范畴间的模型组合

这些方面也是设计模型语言(modeling language)时必须考虑的重要因素之一。

#### 图、模型库的基本逻辑结构

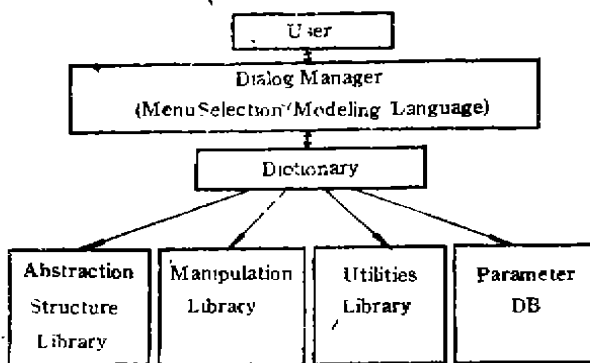


图2

基于结构模型表示法的模型库系统主要包括: 数据字典、抽象结构库、实用程序库、参数数据库。其界面为菜单选择或模型语言。其逻辑结构可图示如图2。

其中:

**Dictionary**: 列出系统中的所有对象及其所在的存储位置。相当于一个索引表。

**Abstraction structure library**: 各种模型的结构表示, 它能为模型构造者提供各种相关的建模知识, 支持其建模活动。

**Manipulation Library**: 存放各种模型求解算法及模型表示形式的转换算子。常见的有, 优化操作、模拟操作、求解方程操作, 查询处理、推理操作、给出模型的图形表示, 给出模型的方程表示等等。

**Manipulation Interface**:

- 接受对模型操作的要求及元素细节表。
- 调用相关的Manipulation。

**Utilities Library**: 存放各种实用程序。如, 元素结构正文表示的编译程序, 正文编辑程序, 表格编辑程序, 图形生成器等等。

**Parameter Library**: 存放模型运行所需的各种参数数据, 它可能来自DSS的数据库, 也可能是中间运行结果。

一个模型运行的简单过程为:

- 1) 确定当前的模型是哪个模型抽象的实例。
- 2) 寻找抽象模型中的Manipulation定义, 特别是求解操作。
- 3) 根据求解操作的定义生成调用序列。
- 4) 执行调用序列求解模型。

#### 五、进一步的课题

以结构模型表示法为基础, 开发新一代模型库系统, 还有许多有意义的工作可做。作者认为, 如下的诸方面都可是这方面进一步的课题:

- 基于结构模型表示法的模型语言及其编译程序的实现。
- 模型库管理系统(MBMS)的设计与实现。
- 模型库(MB)与数据库(DB)之间的自动界面。即使两者之间的联系对于用户来说是透明的。
- 模型库与用户之间的交互式界面的设计与实现。

# 重写技术及其在计算机科学中的应用

陈火旺 王怀民

(长沙国防科技大学计算机系)

## 摘 要

项重写系统是一种描述不确定计算的计算模型。近十年来,重写技术在计算机科学的许多重要领域得到广泛应用,表现出作为知识信息处理系统的良好性质,引起了人们对重写技术的重视。本文介绍重写技术的主要内容,着重强调了完全过程的思想,同时说明了目前重写技术几个活跃的研究课题。本文还介绍了重写技术在自动定理证明、逻辑程序设计和软件开发中的应用。

项重写系统 (term rewriting system, 简记为TRS) 是一种表达不确定计算的形式系统。早在三十年代数理逻辑的研究中, Church等逻辑学家就在TRS方面进行了开创性的研究工作。近十多年来,重写技术在计算机科学领域引起了人们新的兴趣和重视,其原因可以从两个方面分析:一方面,重写技术在形式系统的描述、自动定理证明、程序设计语言、形式化软件开发以及编译代码的优化和生成等方面的应用已经取得了十分有意义的研究成果;另一方面,人们认为重写技术能方便、自然、有效地支持知识信息处理和并行计算机体系结

构。这是今天人们更加重视重写技术的原因所在。这里我们介绍重写系统和重写技术的基本内容,发展趋势以及在自动定理证明,逻辑程序设计语言和形式化软件开发中的应用。

## 一、重写系统及重写技术的基本内容

项重写系统实际上是一组定向等式,即称为重写规则。例如,群理论G中的等式公理被定向之后就得到了重写系统 $R_G^0$ 。

- 以模型库为中心的DSS环境。
- 人工智能技术的应用。

### 参考文献

- [1] R. H. Sprague and E. D. Carlson "Building Effective Decision Support Systems" Englewood Cliffs NJ, Prentice-Hall 1982
- [2] R. H. Bonczek, C. W. Holsapple and A. B. Winston "Future Directions for Developing Decision Support Systems" Decision Science, Vol 11, pp 616—631, Oct, 1980
- [3] D. R. Dolk "A Generalized Model Management System for Mathematical Programming" ACM Transactions on Mathematical Software, Vol12:2, pp 93—125 1986
- [4] D. R. Dolk and B. R. Konsynski "Knowledge Representation for Model Management System" IEEE Trans on Software Engineering SE—10, No 6, 1984
- [5] R. H. Bonczek, C. W. Holsapple and A. B. Winston "A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management" OR, 28, No 2, pp 263—281
- [6] Dutta and A. Basu "An Artificial Intelligence Approach to Model Management in Decision Support System" Computer Vol 17, No 9, 1984