

# 有关智能决策支持系统的知识模型的探讨

徐洁磐

(南京大学计算机系, 南京 210008)

## SOME DISCUSSION OF A KNOWLEDGE MODEL FOR INTELLIGENT DSS

Xu Jiepan

(Department Computer Science, Nanjing University, Nanjing 210008)

**Abstract** This paper proposed a method using knowledge and knowledge process point of view by which every parts of IDSS can be unified, based on the method IDSS can be conveniently interfaced and easy constructed.

Now we are researching a IDSS called NCIDSS based on the method, which has a lot of characteristic different to other IDSS.

**摘要** 本文提出了一种用知识与知识处理观点统一智能决策支持系统的方法, 这种方法能使智能决策系统接口方便, 构造简单.

目前, 我们正在研制一个基于这种方法智能决策支持系统 NCIDSS, 它有很多有别于其它智能决策支持系统的特点.

### § 1. 引言

决策支持系统(Decision Support System)是处理半结构与非结构化问题的系统, 它在处理过程中经常需要用到知识与经验, 因此在其中引入知识库是很必要的, 这种决策支持系统称为智能决策支持系统, 简称 IDSS(Intelligent Decision Support System).

IDSS 在发展过程中已出现有多种构造方法, 最常见的是由数据库、方法库、模型库与知识库等四个库通过接口构成 IDSS, 这是种分离式结构, 它的接口复杂、各库间功能冗余大, 本文中作者提出一种以知识与知识处理来统一 IDSS 的观点与方法, 将 IDSS 中分离的各个库统一成整体, 从而达到消除功能的冗余, 相互取长补短, 接口方便, 结构简单, 功能强等特点, 并使原有四个库简化成数据库、方法库与规则库(退化的知识库)等三个库. 在此思想基础上, 目前我们正在研制一个 IDSS 叫 NCIDSS, 本文主要就是讨论 NCIDSS 的实现思想与原理.

## § 2. 智能决策支持系统的统一表示方式

在本节中主要讨论用一阶逻辑语言统一传统的 IDSS 中各部分的表示方法. 我们知道, 在 AI 中知识的内涵是极其丰富的, 知识一般由事实与规则组成, 知识的表示有多种形式, 在本文中我们用一阶谓词逻辑作为知识表示语言, 用谓词表示事实, 用 Horn 子句的规则表示知识中的规则. 我们用一阶谓词逻辑作为表示 IDSS 各部分的统一语言.

### 2.1 数据库表示

在关系数据库中, 一个  $n$  元关系  $R(x_1, x_2, \dots, x_n)$  可用一阶谓词中的一个  $n$  元谓词:  $P(x_1, x_2, \dots, x_n)$  表示, 而关系数据库是若干关系的集合, 这样, 一个关系数据库可用若干个谓词表示.

### 2.2 方法库表示

方法是一种数学算法, 一般可用公式表示:

$$(y_1, y_2, \dots, y_m) - AG(x_1, x_2, \dots, x_n) \quad (1)$$

其中  $y_j (j=1, 2, \dots, m)$  是输出变量,  $x_i (i=1, 2, \dots, n)$  是输入变量, 这个公式(1)表示由  $n$  个输入变量经计算后可得  $m$  个输出变量的一种数学算法, 它可用下面的  $n+m$  元谓词  $A$  刻画:

$$A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) \quad (2)$$

方法库由若干方法组成, 这样, 一个方法库可用若干个谓词表示.

### 2.3 模型库表示

模型库由模型组成, 而模型则由方法与数据通过 Horn 子句构成(具体构造方法见下节), 因此一个模型库可由若干条 Horn 子句规则组成.

### 2.4 知识库表示

知识库存放知识, 知识由事实与规则组成, 其中事实用谓词表示, 规则用 Horn 子句表示. 从 IDSS 整体讲, 知识库中事实与数据库、方法库中事实是一致的, 因此, 这部分可用数据库、方法库存放, 而真正的知识库仅只要存储规则, 因此, 知识库实际上退化成为规则库.

由上面讨论可知, 用知识的观点将 IDSS 分为事实与规则两个处理部分, 其中数据库、方法库处理事实, 规则库处理规则, 模型库可并入规则库. 这样, IDSS 各部分就变得简单并接口也很容易.

## § 3. 用一阶逻辑语言作构模语言

本节我们用一阶逻辑语言构造模型, 从而使模型成为一个 Horn 子句规则, 因此, 一阶逻辑语言可作为 IDSS 中的构造模型的语言.

模型一般由方法与数据通过构模语言构成. 构模语言的功能是将方法按若干种构造方式构成模型框架, 然后再赋以领域数据从而构成一个模型. 它的具体构造过程如下:

1. 将方法按下面几种方式构成模型框架:

(1) 方法的嵌套: 即一个方法  $A_1$  的输出变量可作为另一个方法  $A_2$  的输入变量.

(2) 方法的递归: 即两个方法  $A_1, A_2$  的输出变量可互相作为对方的输入变量.

(3) 方法的相容: 两个方法  $A_1, A_2$  有相同的输入变量.

(4) 方法的并列: 两个方法  $A_1, A_2$  无相同的输入变量与输出变量.

一个模型框架可由上述四种方法构造而成,图 1 给出了一个模型框架的例,它由方法  $A'_1, A'_2, A'_3$  经嵌套与相容构造而成的。

2. 模型框架的赋值:

由方法所构成的模型框架有其单独的输入与输出变量,对它们赋以具体数据库中值后即构成一个实际的模型。

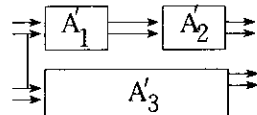


图 1

上面所述的模型构造方法可用一阶谓词逻辑中的 Horn

子句规则表示. 设方法  $A_1, A_2$  可分别用谓词  $P_1(x_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3)$  表示, 则对方法的嵌套可用下面的 Horn 子句规则表示:

$$N(x_1, x_2, x_3, y'_1, y'_2, y'_3): -P_1(x_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3), y_1 = x'_1, y_2 = x'_2, y_3 = x'_3$$

方法的递归可用下面的 Horn 子句规则表示:

$$R(x_1, x_2, x_3, y'_1, y'_2, y'_3): -P_1(x_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3), y_1 = x'_1, y_2 = x'_2, y_3 = x'_3$$

$$R(x_1, x_2, x_3, y_1, y_2, y_3): -R(x_1, x_2, x_3, y''_1, y''_2, y''_3), P_1(y'_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3), y_1 = x'_1, y_2 = x'_2, y_3 = x'_3$$

方法的相容可用下面的 Horn 子句规则表示:

$$S(x_1, x_2, x_3, y_1, y_2, y_3, y'_1, y'_2, y'_3): -P_1(x_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3), x_1 = x'_1, x_2 = x'_2, x_3 = x'_3$$

方法的并列可用下面的 Horn 子句规则表示:

$$T(x_1, x_2, x_3, x'_1, x'_2, x'_3, y_1, y_2, y_3, y'_1, y'_2, y'_3): -P_1(x_1, x_2, x_3, y_1, y_2, y_3), P_2(x'_1, x'_2, x'_3, y'_1, y'_2, y'_3)$$

图 1 所示的模型框架可表示如下:

$$F(x_1, x_2, x''_2, y'_1, y'_2, y''_1, y''_2): -P_1(x_1, x_2, y_1, y_2), P_2(x'_1, x'_2, y'_1, y'_2), P_3(x''_1, x''_2, y''_1, y''_2), y_1 = x'_1, y_2 = x'_2, x_2 = x''_1$$

其中  $P_1(x_1, x_2, y_1, y_2), P_2(x_1, x_2, y_1, y_2)$  及  $P_3(x_1, x_2, y_1, y_2)$  分别表示方法  $A'_1, A'_2, A'_3$  的谓词。

设图 1 所示的模型框架中的变量分别可与数据库中的若干属性值对应:  $a_1, a_2, a_3, b_1, b_2, b_3, b_4$ , 此时可构成模型:  $F(a_1, a_2, a_3, b_1, b_2, b_3, b_4)$

这样整个构模过程可用一阶逻辑语言中 Horn 子句规则表示, 而模型亦可用规则表示。

### § 4. 用一阶逻辑语言表示语义约束

在传统的 IDSS 的各库中均有一些语义约束条件需要表示, 它们都可用一阶逻辑语言表示:

(1) 数据库中语义约束条件的表示

数据库中语义约束条件用一阶逻辑语言表示的问题已在 70 年代数据库逻辑中讨论得很多了, 它包括完整性约束、安全性约束, 模式定义语义、函数依赖等, 我们仅对后面两种作讨论。

设有关系模式:  $R(x_1, x_2, \dots, x_n)$ , 它的对应谓词表示式为  $P(x_1, x_2, \dots, x_n)$ , 其中  $x_i (i=1, 2, \dots, n)$  为属性, 它应满足性质:  $Chr_i(x_i)$  这样  $R$  的模式定义语义可表示:

$$Chr_i(x_i): -P(x_1, x_2, \dots, x_n) \quad (i=1, 2, \dots, n)$$

设有关系模式  $R(x_1, x_2, \dots, x_n)$  及其谓词表示式  $P(x_1, x_2, \dots, x_n)$ , 如有  $x_j$  依赖于  $X, X = (x_{i_1}, x_{i_2}, \dots, x_{i_m})$ , 此时它可用一阶逻辑语言的规则表示:

$$x_j = x'_j: -P(x_1, x_2, \dots, x_n), P(x'_1, x'_2, \dots, x'_n), x_{i_1} = x'_{i_1}, x_{i_2} = x'_{i_2}, \dots, x_{i_m} = x'_{i_m}$$

(2) 方法库中语义约束条件的表示

方法库中方法的语义约束包括完整性约束、安全性约束, 计算依赖, 计算模式约束以及输出变量约束等, 其中前两种与数据库中类似, 后三种我们在文中予以讨论.

计算依赖: 方法谓词  $A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$  中  $x_i$  为输入变量,  $y_j$  为输出变量, 每个  $y_j$  必经由  $X = (x_1, x_2, \dots, x_n)$  计算而得, 即每个输出变量必依赖于输入变量, 这种依赖称为计算依赖, 它可用一阶逻辑语言中的 Horn 子句规则表示:

$$y_j = y'_j: -A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m), A(x'_1, x'_2, \dots, x'_n, y'_1, y'_2, \dots, y'_m) \\ x_1 = x'_1, x_2 = x'_2, \dots, x_n = x'_n (j=1, 2, \dots, m)$$

计算模式语义: 方法谓词  $A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$  中的  $x_i, y_j$  均满足一定的性质  $Chr_k(u)$ , 这样, 此方法的计算模式语义可用一阶逻辑语言的规则表示:

$$Chr_i(x_i): -A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) (i=1, 2, \dots, n) \\ Chr_j(y_j): -A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m) (j=1, 2, \dots, m)$$

输出变量约束: 方法谓词  $A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$  中输出变量各不相同, 这就是输出变量约束条件, 它可用一阶逻辑语言中的规则表示:

$$y_i \neq y_j: -A(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m), i \neq j (i=1, 2, \dots, n, j=1, 2, \dots, m)$$

(3) 模型库中语义约束条件的表示

模型库中模型由方法构成, 它是一个与领域有关的方法组合, 因此, 模型的语义约束与方法类似, 所区别的是它的语义约束多偏重于与领域有关的约束知识, 因此, 模型的语义约束就不作专门讨论了.

这样, 数据库、方法库、模型库的语义约束均可用一阶逻辑语言中的规则表示, 并以约束性规则形式存入规则库并受规则库管理与检验.

这节告诉我们规则库可以支持 IDSS 中的其它三个库的语义约束条件的表示与检验, 使整个 IDSS 的结构更简单, 功能更强.

### § 5. NCIDSS 的结构与处理的特点

基于上面四节所述的思想, 我们所研制的 NCIDSS 采用如下的结构与处理方式:

- (1) 用一阶逻辑语言统一表示 NCIDSS 中的各个库.
- (2) 用一阶逻辑语言作为构模语言用以构作模型. 模型可用 Horn 子句规则表示.
- (3) NCIDSS 由数据库、方法库及规则库三部分组成, 其中规则库是只有规则而无事实的知识库.
- (4) 用一阶逻辑语言中的 Horn 子句规则表示数据库、方法库中的语义约束条件.
- (5) NCIDSS 中用事实表示的知识存储于数据库、方法库中, 其中以数据元组形式出现的事实叫显式事实, 它存储于数据库中, 以算法形式出现的事实叫隐式事实, 它存储于方法库中.
- (6) NCIDSS 中用规则表示的知识存储于规则库中, 规则库能作规则推理与规则检验. 规则库存放表示模型的规则、表示数据库、方法库中语义约束条件的规则以及其它规则. 为表示

方便起见,规则中可以定义若干虚谓词,如模型、语义约束均可用虚谓词表示.规则分为两种,一称演绎性规则,它用于演绎推理,另一称约束性规则,它表示变量间的内在联系.

(7) NCIDSS 的知识查询的过程如下:

首先,对一个知识查询公式,调用规则库并通过规则推理得到一个不出现有虚谓词的知识查询公式.

对这个查询公式中的实谓词,它们或表示关系或表示方法,因此可分解成为对数据库的查询或对方法库的查询,并最终获得查询结果.

上述的查询推理过程实际上是将整个查询分为顺序对三个库操作的过程,这些操作均为传统操作,既简单又有效,且可使知识查询过程不必设置特殊专用操作且可获较高的效率.

下面我们用一个简例说明 NCIDSS 的特点.

例:设有一设备报修 IDSS,它的功能是在给出某类设备及其大、中、小各零件损坏数量后,即能告之该设备因属何种修理范围(修理范围包括大修、中修与小修三种).一般,修理范围由下面的公式决定:

该设备损坏的大零件数为  $r_1$ ,中零件数为  $r_2$ ,小零件数为  $r_3$ .此外,对不同设备、不同零件有不同权值,对大、中、小零件分别有:  $d_1, d_2, d_3$ .此时对:  $d = \sum_{i=1}^3 r_i \cdot d_i$  如有,  $0 < d \leq 50$ , 则设备属小修范围,  $50 < d \leq 100$ , 则设备属中修范围,  $100 < d \leq 200$ , 则设备属大修范围.

在此 IDSS 中设备的不同权值由数据库中关系 R 给出(见表 1),修理范围与计算公式间的关系由数据库中关系 R' 给出(见表 2).

表 1 关系 R

| 设备名 | $d_1$ | $d_2$ | $d_3$ |
|-----|-------|-------|-------|
| A   | 10    | 5     | 1     |
| B   | 5     | 2     | 1     |
| C   | 20    | 10    | 5     |
| ... | ...   | ...   | ...   |

表 2 关系 R'

| x   | d   |
|-----|-----|
| 小修  | 1   |
| 小修  | 2   |
| ... | ... |
| 大修  | 200 |

公式  $\sum_{i=1}^3 r_i \cdot d_i$  为数学模型,求积与求和公式均为方法,即方法库中有方法:

$$\text{Times}(x, y, z): x \cdot y = z \quad (\text{A}-1)$$

$$\text{Plus3}(x, y, z, h): x + y + z = h \quad (\text{A}-2)$$

模型  $\sum_{i=1}^3 r_i \cdot d_i$  可由下面规则构成:

$$M(r_1, r_2, r_3, d_1, d_2, d_3, d): \text{times}(r_1, d_1, z_1), \text{times}(r_2, d_2, z_2), \text{times}(r_3, d_3, z_3), \text{Plus3}(z_1, z_2, z_3, d) \quad (\text{R}-1)$$

决定修理范围可用下面的规则表示:

$P(\text{设备名}, r_1, r_2, r_3, x) : -R(\text{设备名}, d_1, d_2, d_3), M(r_1, r_2, r_3, d_1, d_2, d_3, d), R'(x, d) (R-2)$   
 规则中  $x$  表示修理范围.

这样, 这个 IDSS 其数据库由  $R, R'$  (在表 1, 2 中) 给出, 方法库包含 (A-1)、(A-2), 规则库包含 (R-1)、(R-2).

现给出一个知识查询: 设备 A 的大、中、小零件损坏数分别为 3、5、2, 求 A 属何种修理范围?

此查询要求可写为:  $P(A, 3, 5, 2, x)$ , 其查询过程为:

(1)  $P(A, 3, 5, 2, x) : -R(A, d_1, d_2, d_3), M(3, 5, 2, d_1, d_2, d_3, d), R'(x, d)$

(2)  $P(A, 3, 5, 2, x) : -R(A, d_1, d_2, d_3), \text{times}(3, d_1, z_1), \text{times}(5, d_2, z_2), \text{times}(2, d_3, z_3),$

$\text{Plus3}(z_1, z_2, z_3, d), R'(x, d)$

(3) 此时,  $P$  可由  $R, R', \text{time}$  及  $\text{Plus3}$  决定, 而  $R, R'$  可由数据库查询得到,  $\text{time}, \text{Plus3}$  可由方法库查询得到, 顺次对下列各项查询:

$R(A, d_1, d_2, d_3), \text{times}(3, d_1, z_1), \text{times}(5, d_2, z_2), \text{times}(2, d_3, z_3), \text{Plus}(z_1, z_2, z_3, d), R'(x, d)$

最后得到  $x = \text{中修}$ .

**结论:** 本文提出一种以知识及知识处理的观点统一 IDSS 的思想, 并以一阶逻辑语言为知识表示语言, 在此思想推动下, 我们可以简化 IDSS 的结构, 增加功能与方便接口, 将一个分离的系统构造成一个整体, 用此种思想我们在 IBM-PC/386 机上研制了一个 IDSS 叫 NCIDSS, 它具有上述的优点, 并且占内存少、效率高.

### 参考文献

[1] Bocea, Some Step Towards a DBMS Based on KBMS, IFIP (1986).  
 [2] Dolk, Konsynski, Knowledge Representation for Model Management System, IEEE Tran. On Software Engineering, Vol. 10, No. 6 (1984).  
 [3] 徐洁磐, 张仲谋, 一个智能决策支持系统生成器的设计与实现, 计算机研究与发展 1990. 10.

欢迎 订 阅

《计算机辅助设计与图形学学报》

1993 年开始邮局公开发行

邮发代号: 82-456 国际标准刊号: ISSN1003-9767

国内统一刊号: CN11-2925/TP