

智能决策支持系统的信息结构及其实现

Takahara Y., Iijima J.和Sato R.〔日〕

摘 要

本文提出了一个具体的智能决策支持系统(IDSS)的方案及其实现,描述了为实现该系统所需的信息结构,并揭示了IDSS设计问题的诸多特征。作为该方案的论证文中还阐明了一个用于公司规划问题的IDSS的实现情况。

一、引 言

最近,人们一直致力于研究如何把人工智能技术引入决策支持系统(DSS)中,并指出人工智能(AI)技术可以与传统的DSS生成器按各种方式在不同程度上相结合。可是,实际上人们对于AI技术与DSS问题结合的合适体系结构问题至今还没有明确的答案。

我们的研究是为了以下目的来开发IDSS生成器。

- 1) 开发战术级辅助规划系统;
- 2) 寻求AI技术与DSS技术的协调结合,而不是其简单的集成方法;
- 3) 从数学的一般系统理论意义上阐述所开发的系统,以弄清IDSS的基本特征;
- 4) 探讨开发IDSS的方法学。

IDSS的问题求解过程,由三种极不相同的信息处理过程组成,见图1。因为在问题求解中有人为因素,所以DSS应支持问题求解人对信息给出的含义方面。可惜,用IDSS找出人类问题求解的完备模型,是件很不容易的事,若不然,我们就可以直接设计IDSS。因此,我们的方法是从功能和规格说明两个方面去研究并建造具体而特定的IDSS原型。在这项研究开始时,我们已有一个称为ARIS-

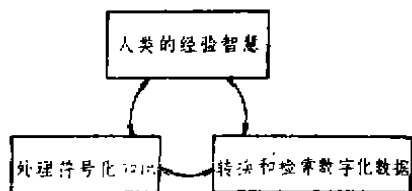


图1 用IDSS求解问题的三种不同信息处理

TOTLE的DSS生成器,它是一个S.A.I.商品化的DSS。我们一直在构造像电子表格、专家系统及其接口这样的部件,并且利用这个IDSS生成器,我们建造了证券投资IDSS、利润计划编制IDSS和公司规划IDSS。

二、公司规划IDSS的实现

1. 公司规划IDSS概述

假设有一个制造业,它生产两种产品,并购买两种原材料,而且有用于支付和贷款的流动资金。试订出这个制造业半年的公司规划。

公司规划由几个决策子问题和综合指标计算组成,即销售计划、生产计划、采购计划、直接和间接管

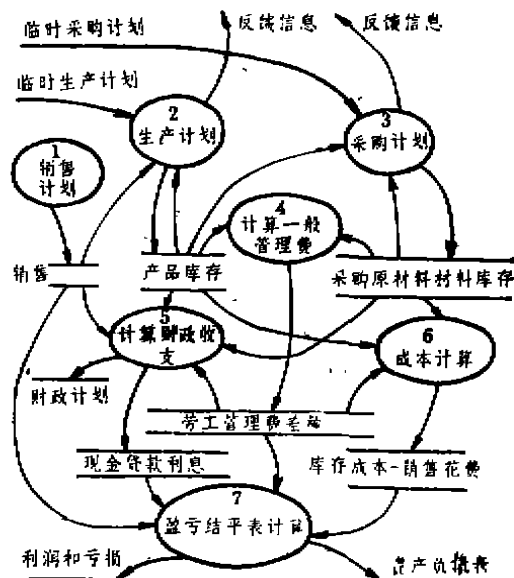


图2 公司规划IDSS的信息及处理概貌

理费用、成本计算、筹资和财政报告。这种规划是一种执行规划。

用户使用我们的IDSS, 将解决月产量和采购计划问题, 以便能得到月筹资计划、利润和亏本报告, 以及半年的资产负债表。这样, 用户就可以专门从资金流量方面来评价这个制造业的经营模拟情况。如果用户不满意这个结果, 可以再做另一种试验。这个系统的处理概貌示于图2。这里采用了结构化分析中数据流图的表示法。图2中上下划线的实体和圆圈中的实体分别表示数据文件(即有关信息)和它们的处理。每一处理过程都将改变输入数据文件的内容, 圆圈里的数字表示图中操作的顺序。

这个IDSS生产计划是其决策子问题之一。该计划问题说明如下: 设半年的总产量 $Q_{Total Product}(i)$ 是第 i 种产品半年的产量, 第 i 种产品 k 月份的产量为 $Q_{Product}(i)(k)$, 第 i 种产品在 k 月份($1 \leq k \leq 6$)的库存量为 $Q_{Inventory}(i)(k)$, 而 k 月份第 i 种产品的最小库存量为 $Q_{LevelProdInv}(i)(k)$ 。试求第 i 种产品在 k 月份的产量 $Q_{Product}(i)(k)$, 其中 $1 \leq i \leq 2$, 且 $1 \leq k \leq 6$, 该产品满足如下两个条件: 对于每一 i 和 k ,

(1) $\sum_{k=1,6} Q_{Product}(i)(k) = Q_{Total Product}(i)$,

(2) $0 \leq Q_{Inventory}(i)(k) \leq Q_{LevelProdInv}(i)(k)$ 。(注意, $Q_{Inventory}(i)(k) = Q_{Inventory}(i)(k-1) + Q_{Product}(i)(k) - Q_{Sales}(i)(k)$, 其中 $Q_{Sales}(i)(k)$ 是第 i 产品 k 月份的销售量。

这个生产计划方案是由R. Mattessich提出来的。尽管他编过一个程序来求解这个问题, 但我们的实现允许用户直接自由地输入 $Q_{Product}(i)(k)$, 从而可得到交互性计划。采购计划用类似的方法解决。如果用户完成这一计划并执行这些计算模型, 就能得到财政报告。这些模型的执行顺序应提前安排妥当。

2. 公司规划IDSS中的信息和处理

在我们的IDSS中, 电子表格是用户和IDSS之间的“万能”接口。也就是说, 用户在电子表格上描述他的问题, 然后在该表格上求解问题。这个系统的结构如图3所示。在第四部分给出该系统的严密描述。

如上所述, 用户的主要问题是做出财政报告。用户应解决两个子问题: 一是生产计划问题, 二是采购计划问题。图4中描述了用户和系统所关心的

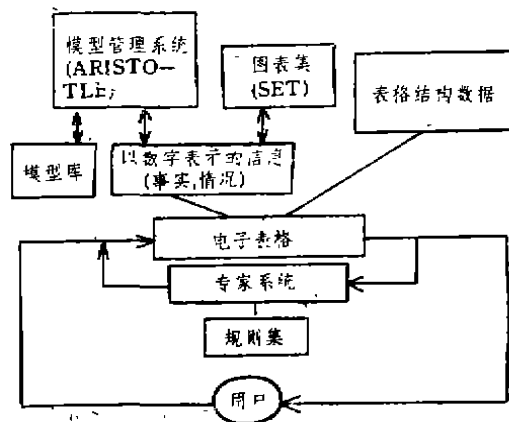


图3 IDSS的配置

生产计划信息和处理。月销售量、月产量和月库存量表示在电子表格上。月产量的确定要求满足一些约束条件。DSS数据库和电子表格中的数值数据被连接起来。月库存量由电子表格的自动重算功能计算。这些处理功能是图4中圆圈2.2所示的功能。

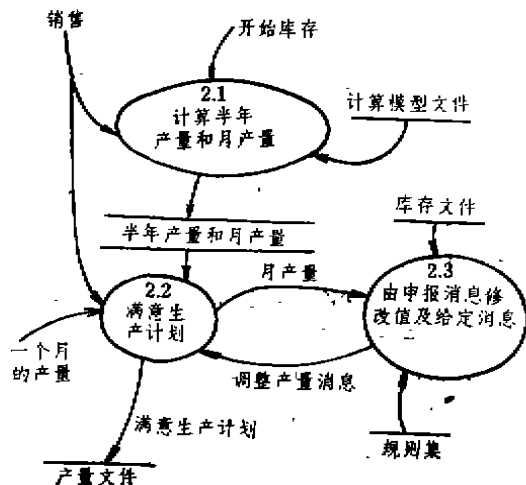


图4 由专家系统辅助的交互生产计划

电子表格上的每个活动由图4中圆圈2.3所示的专家系统部分地控制和监测。用户一经输入 k 月份第 i 产品的产量 ($Q_{Product}(i)(k)$), 符合以下两个约束条件时, 就执行两种处理: ①保持半年产量不变, 现在与以前产量之差加到下一个月份; ②若某月份的库存量不足, 则专家系统在某些槽里写上库存量短缺的信息。如果所有库存满足最小库存条件, 专家系统就写上“库存条件满足”。我们称这些功能是由专家系统监控的。(1)和(2)的处理被描述为规则, 它们由IDSS用户或设计者编写。这些规则也

可以在电子表格上进行描述。

图4中圆圈2.1的功能是从库存开始计算半年的产量并订出销售计划。

三、IDSS问题求解的智能

这里我们研究在IDSS问题求解活动中AI技术是如何发挥作用的。

一般说来,在某些方面AI技术可以与DSS技术结合在一起。这些方面包括人机接口、数据库管理部份、模型建立、通过模型的方案产生与分析。

在实现过程中,AI技术系由专家系统监控而获得的。生产计划所要解决的问题是令人满意的任务,这里令人满意的任务意指该问题能满足约束条件。它比任务优化简单,因为一般说来优化不仅需要满足目标的一套准则,而且也需要断言该目标具有可选空间中任何其它目标所不能匹配的性质。即使求解似乎简单,组织中的许多决策问题也具有令人满意的性质,但把现实问题表述成一个优化任务却是十分困难的。“为保持半年产量不变的用户输入在线调整”功能被看作是缩小或剪裁搜索空间。这里,我们的观点是搜索策略主要在人的头脑里,而部分在规则集中。

因此,如果用户想改变一下保持半年产量不变的方式,那么他可以在电子表格上直接赋上他希望的数值,或者在电子表格上重新编写规则以替换掉旧的规则。在这种意义上讲,用户直接在IDSS配置中进行规划时,可以调整他们的“口味”。

接下来,我们考察IDSS中电子表格的作用。电子表格是人与其它IDSS部件的接口,这些部件包括模型库、模型管理系统、数据库、数据库管理系统、图形库和专家系统。这些部件能够处理许多信息。在电子表格上有两种抽象数据类型。一类是数值抽象(比如月产量),另一类是符号抽象(比如“四月份库存不足”的信息)。用专家系统可以动态地处理和改变这类符号数据。它指出用户应做的下一个决策。如果用户按照这一信息改变四月份的产量值,那么其结果是改变了他的问题,并再一次接收到一个信息。

所发生的事是自调节反馈。这种反馈机理作用如下,假设有这样一个系统,它由处理机、控制器及控制器执行的规则集组成。这就是说,该系统中涉及到三个等级。控制器传送一个信号给处理机,这时处理机可以改变它的操作,并且这一改变影响到规则集。因此,控制器改变它的控制规律。在这

种意义上讲,控制器本身是按其自己的行为进行组织的。看起来,有许多自调节反馈机理的例子。Imada把这一机理叫做人作为社会成员的自反性,Giddens则称为社会成员的“两重性”。超循环中的自动催化行为的思想是另一个例子。一个简单的实例是语言的语法与语音法则间的关系。

换句话说,人的功能、符号化知识的处理和数字处理间的联系虽说简单仍是一种自调节反馈的实现。这种反馈是在电子表格上由两种类型的数据抽象进行的。

四、公司规划IDSS的表示

1. 概要

这种表示法是系统理论中最重要的概念之一,是状态概念的一种应用。下面表示法中的“状态记录”相应于自动机的状态。“状态记录”的结构表示了IDSS中所需信息的特征。IDSS中由用户进行的操作及模型进行的变换均应保持符号化知识与数据之间的一致性,它作为不变性条件在表示中被体现出来。

这种表示法由三部分组成,状态、操作符和不变性条件。

2. 表示

记号:采用维也纳开发方法(VDM)的记号。/*...*/表示一个注释。“VAR, vartype”表示一个变量被定义其名字VAR及其类型vartype。“Anything-list”是任何库类型的一个有限表。两个表(list1和list2)的连接被表示为“list1/list2”。记录结构允许变量的取值是从可能不同类型抽取的定长值序列。这对应于数学上的笛卡儿积(直积)。符号“::”用于记录构造,而不用变量定义情况下的符号“:”。字段选择符,也是变量,用作数学上的投影。先前定义的小写字母的变量代表其瞬间值。设func是一个函数,dom func是这个函数的定义域。“if...then...else...”用作作为一种元语言。

(1) 状态 IDSS的“状态”如图5所示。IDSS中数值数据的数据库的结构描述成脚本集(scenario-set)。一个脚本是变量的时间函数的集合。IDSS中有许多脚本,因为它们用于模拟和比较。因此规定出变量名、时间和脚本,就能得到一个值。在本实现中,ARISTOTLE数据库和电子表格数据间有一个连接器。在这种表示里,这种连接被简单地表示成Cell Value中的脚本数据。

(2) 操作符 操作符的说明由四部分组成:①操作名及其所选的参数;②带有关键字“ext”的--

```

State:: INTERFACESHEET: SheetState /* state of spread sheet with abstracted data */
NUMERICALDATA: Scenario /* numerical data used in IDSS */
KNOWLEDGE: Rule-list /* a set of rules for expert system */
CONTEXTLIST: IsPredicate-list /* interface between sheet and expert system */
OPERATIONCONTROL: Script /* waiting list of the operations */

SheetState:: CONTENT: SheetData /* a spread sheet */
CURSOR: XPosition×YPosition /* cursor position */
COMMAND: SheetTransform /* command a user inputted */

Rule:: IFPART: IsPredicate-list /* if-part of a rule */
THENPART: IsPredicate-list /* then-part of a rule */
/* A rule consists of both if-part and then-part */

Script:: SHEET: Bool /* TRUE if the sheet operation can start */
INFER: Bool /* TRUE if expert system can start */
CALCULATE: Bool /* TRUE if calculation model can start */

SheetData = {#(XPosition×YPosition→CellValue) /* the set of possible sheets */
CellValue = RUTextU (arithmetic polynomial over SSterm) UTime×VarName×Scenario
/* contents of a cell */

SSterm = {s | (s ∈ CellVariable)∨
(V t1, ..., tn ∈ RUSSterm) (p(t1, ..., tn) ∈ SSterm), where p is a polynomial}
Scenario: SenId → Variable-set /* numerical data in ARISTOTLE */
Variable = VarName → TimeFunction-set /* variables in IDSS are time functions */
SenId: NaturalNumber /* there can be many scenarios */
TimeFunction: Time → R
VarName: Text
R = /* the set of real numbers */
A = /* the set of alphabets */
Text = A*
XPosition: NaturalNumber
YPosition: NaturalNumber
SheetTransform = Sheetdata → Sheetdata /* command change the sheet data */
Time: NaturalNumber
Bool = {TRUE, FALSE}

IsPredicate:: ATTRIBUTE:TextUCellVariableU"CURSOR" /* attributes used in a rule */
EQUAL: ("=")
VALUE: TextUR
CellVariable = {#(X×XPosition×YPosition)

```

图5 IDSS的状态

```

MOVECURSOR(M: SheetTransform)
/* describes cursor movement on a spread sheet */
ext INTERFACESHEET: w SheetState
pre is-move-command(m)
post interfacesheet(CURSOR(x',y')) = moved(x,y,m)

is-move-command: SheetTransform → Bool
is-move-command(m) = TRUE, m ∈ {→, ←, ↑, ↓}
moved(x,y,m) = /* x or y is incremented/decremented by m */

```

图6 移动光标

个子句,指出该操作需要访问状态的哪一部分。读/写访问规定为“rd”/“wr”;③前提条件和④后置条件分别使用关键字“pre”和“post”表示,他们是输入参数和状态变量之上的谓词。

IDSS中有四种操作符:①MOVECURSOR,在电子表格上移动光标;②WRITECELL,给电子表格输入数据并启动专家系统;③CONSULT,

进行推理;④CALCULATE,用于管理模型库。这些操作符分别在图6、7、8和9中说明。

(3) 不变性条件 这种表示是表达无论执行什么操作时数据的一致性的。图10中的“Inv-state”表示哪些应该一致。

五、结 论

在设计具体的IDSS时,遇到了若干困难。这

```

WRITECELL( c: SheetTransform)
/* writes a cellvalue in a cell.
   And immediately starts consultation by expert system */
ext  INTERFACESHEET:  wr SheetState
     KNOWLEDGE:       rd Rule-list
     CONTEXTLIST:     wr IsPredicate-list
     OPERATIONCONTROL: wr Script
pre  is-cellvalue(c)
post interfacesheet(content') = interfacesheet(content) + [ cursor → evaluate(c) ] &
     operationcontrol(SHEET) = FALSE &
     operationcontrol(INFER) = TRUE &
     contextlist' = convertsheet(interfacesheet', knowledge)

is-cellvalue: SheetTransform → Bool
is-cellvalue(c) = TRUE, c is CellValue
evaluate: CellValue → CellValue
evaluate(c) = c, if c is RUText,
              evaluate(R(x,y)), if c is R(x,y),
              p(evaluate(t1), ..., evaluate(tn)), if c is p(t1, ..., tn).
convertsheet: SheetState × Rule-list → IsPredicate-list
convertsheet(s, r) = contextlist ←→
  r(IFPART (ATTRIBUTE) = contextlist(ATTRIBUTE) &
    if (∃ s contextlist) (f(ATTRIBUTE) s dom interfacesheet(CONTENT))
    then f(VALUE) = s(CONTENT) (f(ATTRIBUTE))
    else if (∃ s contextlist) (f(ATTRIBUTE) = "CURSOR")
    then f(VALUE) = s(CURSOR).

```

图7 写单元

```

CONSULTO
/* executes forward inference with contextlist as facts and rule-set as knowledge.
   The spread sheet is consistent with the contextlist. */

ext  KNOWLEDGE:       rd Rule-list
     INTERFACESHEET:  wr SheetState
     CONTEXTLIST:     wr IsPredicate-list
     OPERATIONCONTROL: wr Script
pre  operationcontrol(INFER) = TRUE
post contextlist' = stepinfer(contextlist, r, 2) &
     if (∀ s contextlist') (f(ATTRIBUTE) s dom interfacesheet(CONTENT))
     then interfacesheet(CONTENT) (f(ATTRIBUTE)) = f(VALUE) &
     operationcontrol(INFER) = FALSE &
     operationcontrol(SHEET) = TRUE
     operationcontrol(CALCULATED) = TRUE

stepinfer: (IsPredicate-list) × Rule-list × NaturalNumber → IsPredicate-list
stepinfer(c, r, 0) = c
stepinfer(c, r, n) = if ( (r(IFPART) ⊂ stepinfer(c, r, n-1)) &
  (r(THENPART) ⊄ stepinfer(c, r, n-1)) )
  then stepinfer(c, r, n-1) / r(THENPART)
  else if ( (r(IFPART) ⊂ stepinfer(c, r, n-1)) &
    (r(THENPART) ⊂ stepinfer(c, r, n-1)) ) or
    (r(IFPART) ⊄ stepinfer(c, r, n-1))
  then stepinfer(c, r, n-1)

```

图8 推理

些困难包括不同模型的建造、模型的证实、规则的表述、规则的证实、以及问题求解过程的分解。头两个问题与 IDSS 中的 AI 技术没什么关系。在 IDSS 中, 规则被写成 "if-then" 形式, 并且约二十条规则用于规划电子表格。规则的表述和证实都是困难的任务。为了调试规则, 可以在推理执行前后

使用 CONTEXTLIST, 它是电子表格与专家系统间的接口数据。一个问题是如何把求解过程分解成三种数据处理功能, 需要对求解问题进行深入研究才能作出这个问题的回答。正如本文第三部分所讨论的, 我们在电子表格上提出了两种数据抽象。一种是模型计算所处理的数值数据的抽象, 另一种是

```

CALCULATEO
/* transforms numericaldata, i.e., scenario, as described in the calculation models */

ext NUMERICALDATA: wr Scenario
  OPERATIONCONTROL: wr Script
pfe operationcontrol(CALCULATE) = TRUE
post numericaldata' = modelrun(numericaldata) &
  operationcontrol(CALCULATE) = FALSE

```

图9 后模型计算数值数据的值

```

/* This simply requires consistency of data in IDSS and rules of expert system. In other words,
the data in IDSS would not change if expert system were applied. That is, the data is the fixed
set against operation of expert system. */

inv-state = (Vn)
  (stop infer (convertsheet(interfacesheet, r), n) = convertsheet(interfacesheet, r))

```

图10 状态不变性

符号化知识推理的结果。电子表格用作一种工作台，而专家系统不仅可用于符号推理，也能缩小可选择的搜索空间。因此，用户在电子表格上的操作由专家系统监控并且按自调节反馈机理加以简单实现。

今后的问题包括：（具体问题的）财政计划系统的开发（作为IDSS生成器问题）。

• 实现DSS数据库、电子表格、专家系统和图

形库的更一致的数据结构。

• 建造具有高功能的专家系统（现已有了Prolog可用）。

• 开发新的接口特性，一般实现各种层次的自调节反馈机理。

参考文献（略）

〔李海泉译自《ICSSE/88》杨俊、仲源校〕

（接第65页）

写的完全定理证明器的解释器，还有一个把THEORIST的事实、限制和缺席变换为PROLOG的编译器。现在我们正在构造一个基于真值维护和依赖制导回溯的系统。我们也在构造一个能做并发一致性检查的系统。

在理论中通过要求我们只使用缺席的基例，避免了缺席中由存在和全称量化变量引起的问题。

10. 结论及今后的工作

本文的主题是，缺席推理问题不是一个逻辑问题，而是如何运用逻辑的问题。这种方法给了我们一种探讨其它问题的巧妙办法。这就是说，出于某种目的，决定什么时候一个脚本“好”于另一个脚本。Poole认为，欲最佳地解决网络的继承问题得最好选择最

具体的理论，即当具体知识和更一般知识可用时，我们最好使用最具体的知识。Goeble提出了公理化框架公理中的多扩充问题是如何在THEORIST框架中通过选择时序最大持续理论加以解决的。对于诊断系统，我们最好选择最合理的解释。

这一理论奠定了THEORIST系统的基础，已被用于不同领域，包括诊断、建立用户模型、类比推理的问题求解、代词辨析和学生的学习障碍诊断。

参考文献（略）

〔付志明 马 简译自《Artificial Intelligence》36（1988）P.27—47，刘毅之仲源校〕