

3W-5

関係データモデルに基づいたモデル構築支援

平石 邦彦¹⁾・黒川 伊保子²⁾¹⁾富士通(株)国際情報社会科学研究所, ²⁾(株)富士通SSL

1. はじめに

問題解決のための方法の1つに問題のモデル化がある。構築したモデルを解析することにより、対象がもつ性質を推定することができる。ORや経営科学の分野ではモデル化による問題解決が主流であり、様々な意思決定モデルが提案されてきた。また、意思決定支援システムにおいても、モデルの構築、管理は重要な機能の1つである¹⁾²⁾。しかし、現実の問題が既存のモデルの表現形式に適用可能かどうかを判断し、モデル化する過程は、人間が行う作業として残されている。

本研究では、関係データモデルに基づいたモデル化の方法を提案する。これは、関係がもつ関数従属性を関数の形で陽に記述することによりモデル化を行うものである。また、構築したモデルの解析に、制約論理言語であるCLP(R)³⁾を用いた例についても述べる。

2. データ構造とモデル

データベースにおけるデータモデル論の目的は、実世界にある構造をもったデータを、どのように形式化(抽象化)して計算機内部の表現に移すかを探究することである⁴⁾。データベースは、このようなデータ構造に従って集められたデータ集合であり、データ構造の実世界における外延(extension)である。一方、データ構造の内包(intension),すなわち、データ構造から実世界で取り得るデータ集合への写像がモデルであると考える。

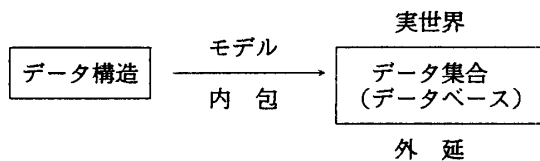


図1 モデルの概念

データ構造として、関係データ・モデルを考える。関係Rは、属性値集合 A_1, \dots, A_n の直積の部分集合 $R \subset A_1 \times \dots \times A_n$ として定義される。これを $R[A_1, \dots, A_n]$ で表す。

関係の外延は、どのような状況を対象としているかにより異なる。たとえば、関係'学生'[学生番号, 氏名,

住所]を考えると、関係'学生'の外延は、どの学校を対象としているかにより、また学校が同じでも年度により異なる値をもつ。状況を表す記述(この場合、学校や年度)を指標と呼ぶことにする。関係 $R[A_1, \dots, A_n]$ の内包は、指標 i に $A_1 \times \dots \times A_n$ の部分集合を対応させる写像である。指標を関係の属性としてもたせれば、異なる指標に対応するデータを一つの関係に記述できる。このとき、指標はキー属性の一部になる。たとえば、関係'学生'[学校名, 学生番号, 氏名, 住所]を作れば、複数の学校の生徒を同一の関係に含めることができ、そのキー属性は、学校名と学生番号になる。

関係Rの属性の集合X, Yについて、Xの任意の属性値に対してYの属性値が高々1個しかないとき、YはXに関数従属であるという。関係Rの任意の属性の集合Xについて、つぎが成り立つとき、Rは第3正規形であるという：

'Xに含まれない属性がXに関数従属ならば、Rのすべての属性がXに関数従属である'

関係Rが第3正規形であるとき、キー属性Xに対し、それ以外の属性Yの値はXの値によってのみ一意に決定される。したがって、 $R = \{ \langle x, y \rangle \mid x \in X \wedge y = f(x) \}$ となる関数 $f: X \rightarrow Y$ が存在する。XをYに対する指標と見ると、関数fは、RのYに関する射影 R_Y の内包になる。このような関数fによりモデル(Rの内包)を記述する。

3. モデル記述言語MODEL/R

損益岐点分析を行うモデルをモデル記述言語MODEL/R(Model DEscription Language for Relational data)により記述した例について説明する。モデルの定義は4つの部分からなる。

①属性値集合の定義： $a := b \# X [if c]$ により、データ型bで変数Xに関する条件式cを真にするものがデータ型aをもつことを定義する。

②関係スキーマの定義：

$$r := [a_1, \dots, a_n] : [a_{n+1}, \dots, a_m]$$

a_1, \dots, a_n は関係rのキー属性の、 a_{n+1}, \dots, a_m はそれ

以外の属性のデータ型である。

③関数の定義：

```
r # [A1, ..., An]:[An+1, ..., Am] :=
  式 in 変数の定義域 where 条件式
```

関係 r の関数従属性を記述する関数を定義する。関数は、他の関係の検索により得られる値に対する計算式の形で定義される。

④値の定義：関係（データ型）に値を割り当てる。

modelschema break-even

```
type
  products:=chr#X; /*製品名 */
  unit :=int#X if X>=0; /*製品の単位*/
  dollars :=real#X /*ドル */
relation
  variable-costs-per-unit
  variable-costs :=[products]:[dollars];
  total-variable-costs:=[]:[money];
  fixed-costs :=[]:[money];
  sales-units :=[products]:[dollars];
  selling-price :=[products]:[dollars];
  sales-dollars :=[products]:[dollars];
  total-sales-dollars :=[]:[money];
  net-income :=[]:[money]
/* 順に、単位当り変動費、変動費、総変動費、
  固定費、販売量、販売価格、売上高、
  総売上高、利益 */
```

```
function
  sales-dollars#[P]:[V] :=
    expr(A*B,V) in
      [selling-price#[P]:[A],
       sales-units#[P]:[B]];
  /* 各製品の売上高=販売価格×販売量 */
  total-sales-dollars#[P]:[V] :=
    sum(A,V) in [sales-dollars#[M]:[A]]
    for [M] in [products#M];
  /* 総売上高=各製品の売上高の合計 */
  variable-costs#[P]:[V] :=
    expr(A*B,V) in
      [variable-costs-per-unit#[P]:[A],
       sales-units#[P]:[B]];
  /* 各製品の変動費=単位当り変動費×販売価格 */
  total-variable-costs#[P]:[V] :=
    sum(A,V) in [variable-costs#[M]:[A]]
    for [M] in [products#M];
  /* 総変動費=各製品の変動費の合計 */
  net-income#[P]:[V] :=
    expr(A-B-C,V) in
      [total-sales-dollars#[P]:[A],
       total-variable-costs#[P]:[B],
       fixed-costs#[P]:[C]]
  /* 利益=総売上高-総変動費-固定費 */
```

```
data
  products#[deluxe, standard];
  variable-costs-per-unit#[[deluxe]:[45],
  [standard]:[35]];
  fixed-costs#[[]:[300000]]
```

end.

4. モデルの解析

MODEL/Rを解釈・実行するシステムはPSI上のESPおよび制約論理プログラム言語のCLP(R)で構築している。ここでは、CLP(R)を用いた解析について述べる。

break-evenモデルで損益分岐点（収益が0となる点）における販売量と価格を求める場合を考える。

```
breakeven(A, B, C, D):-
```

```
  solve(net-income#[[]]:[0],
        [sales-units#[deluxe]:[A],
         sales-units#[standard]:[B],
         sales-price#[deluxe]:[C],
         sales-price#[standard]:[D]],R).
```

述語breakevenは、損益分岐点における販売量と単価の関係を求める。solveの第1引数には求める値、第2引数には入力データを入れ、第3引数には計算の途中で求められた値が返される。solveはつぎの順序で値を探索する。

①値の定義に存在するか？

②入力データに存在するか？

③計算の途中で既に求められているか？

④関係の関数を計算する。i)関数が参照している他の関係の値を同様の手続きで求め、ii)その値により計算を行う。

製品の単価を与えたときの販売数は、

```
?- breakeven(A,B,80,50).
A = 8571.43-0.428571*B
A >= 0
B >= 0
```

により求められる。結果が一意に定まらないときは、式が返される。販売総数が与えられている場合は、

```
?- breakeven(A,B,80,50), A+B=15000.
A = 3750
B = 11250
```

となる。計算に方向性を持たないので、入出力を自由に設定できる。たとえば、販売数を与えたときの単価は、

```
?- breakeven(3750,11250,X,Y).
X = 230 - 3*Y
```

により計算できる。入力を段階的に変化させるには、つぎの述語stepを用いる。

```
step(Var, From, To, Step):-
  From <= To,
  (Var = From ; step(Var, From+Step, To, Step)).
```

```
?- breakeven(A,B,80,X), A+B=15000, step(X,30,60,5).
```

これにより、Xは30から60までの5刻みの値をとることが定義される。

5. まとめ

モデル構築を支援するために、関係データモデルに基づいたモデル記述方法を提案した。制約論理プログラム言語と組み合わせることにより、柔軟な問題解決が可能になる。

なお、本研究は、第5世代コンピュータプロジェクトの一環として行われたものである。

《文献》

- 1) R.H.Sprague, W.D.Carson: Building Effective Decision Support Systems, Prentice-Hall(1982).
- 2) J.J.Elam(ed), Decision Support Systems, Vol.2(1986).
- 3) N.C.Heintze et al.: Constraint Logic Programming: A Reader, 4th IEEE Symp.on Logic Programming(1987).
- 4) 有澤: データベース理論, 情報処理叢書2, 情報学会(1980).