解　説

データとデータ処理の構造
——情報空間モデルと FORIMS——*

小　林　功　武**

1. はじめに

ACM などの計算機学会での現在の話題の事項を挙げるならば,"高水準言語", "ソフトウェア工学" それに "データベース" となろう. これらはそれぞれ異なった目的を追っているが, データとデータ処理の構造の明確化を主張する点において互いに密接に関連している.

筆者は一つの高水準言語として 1962 年に提起された情報代数[1]を拡張し, データとデータ処理の抽象モデルとして情報空間モデルを構成した. 約7年にわたって, モデルの構成と平行して開発された実験システムとしてFORIMSがある. この稿では両者を概説し, システム設計の一方法を述べる.

2. 値集合とその上の演算

データ処理の基本単位となるのは種々の値集合である. 値集合には自然数, 整数, 有理数などの数の集合, 真理値集合, あるアルファベットの連結によって生成された記号列の集合など, 比較的単純なものもある. これらの値は計算機中で固定長または可変長のビット列で表示される. これらの値集合の上で種々の演算(関数) が定義されている. 値集合 $V_k(1\leqq k\leqq m)$ の上の演算 f は

$$f:\ V_1\times V_2\times\cdots\times V_m\to V_f$$

とかける. $V=V_1=\cdots=V_m=V_f$ であれば, f は値集合の中の演算子で, V は f を持つ代数系である.

ビット列で表示された値の集合は, 数学的な意味の値集合を近似してはいるが必ずしも一致しない. 値集合の上の演算もビット列の集合の上で定義されビット列を値とする演算に変換されるが, これはもとの演算の結果のビット列表示と必ずしも一致しないことに注意する必要がある.

値集合にはさらに複雑なものがある. 値集合が1次元ないし多次元の配列の集合でこの上に種々のベクトルや行列演算が定義されていることもある. 配列の要素はより単純な値集合の元である. 可変長の1次元配列はその上で定義されている演算に応じてリストやスタックと呼ばれる. 文章やグラフを値とする値集合を扱わなければならないこともある.

拡張可能言語や構造的プログラミングの議論と関連し, これらの値集合をその上で定義されている演算と併せて形式的に定義しようという試みが盛んである. アプリケーションに応じて種々なデータ・タイプを扱う必要を生ずるが, データベース技術の立場から, レコード, ファイル, データベースなどの概念は特に重要である.

3. データベースの基本モデル

データベースの抽象モデルとしていくつかのものが提起されている. データ構造をそれ以上分割できないデータ単位 n 個の間の関係であると把える E.F. Codd らのリレーショナル・モデル[2], すべてを単位データ(それ以上分割できるかどうかは問わない)の間の2項関係と考える M.E. Senko の DIAM[3]や J.R. Abrial のデータ・セマンティクス[4]は代表的なものである. 実用的にはデータ単位間の n 項関係としてのレコードと, レコード間の2項関係を併存させるモデルが有用で, DBTG[5]の基礎概念などがこれに当たるが, 形式化が不十分であった.

情報空間モデルでは $V_0, V_1, \cdots, V_n$ の n 個の値集合のデカルト数

$$\mathfrak{B}=V_0\times V_1\times\cdots\times V_n$$

をモデルのフレームワークとして採用する. V_0 はタイプと呼ぶ特別な値集合で, $V_k(k\geqq 1)$ は任意の値集

* Information Space Model and FORIMS by Isamu KOBAYASHI (Advanced Research, Nippon Univac Sogo Kenkyusho, Inc.)

** (株)日本ユニバック総合研究所

合である．V_0, V_1, …, V_n の全部に Ω (定義不能) という値を加えることによって，すべての情報をその中に写すに足る十分大きなフレームワークを得ることができる．計算機が扱うことのできるのは，このように定義された情報空間であり，かつそれ以上のものでない．

$\mathfrak{B}$ は原点を $(\Omega,\Omega,\cdots,\Omega)$ とし，V_0, V_1, …, V_n の各軸が原点で直交する $n+1$ 次元空間と考えることができる．V_0 軸に截片 p で直交する超平面の部分集合をファイルと呼ぶ．p はファイル名である．いくつかのファイルの集りをデータベースという．

4. レコード間の関係

前章に定義した情報空間は情報代数の属性空間やリレーショナル・モデルのフレームワークとほぼ同等の概念で，わずかにその幾何学的イメージを与えたものである．ところでDBTG流モデルとリレーショナル・モデルの大きな相違はレコード間の2項関係を考慮するか否かにある．両者の主張のそれぞれに首肯し得る点があるが，議論の噛み合わないのは困ったものである．

レコード間の2項関係を写すものとして

$$\mathfrak{W}=W_0\times\mathfrak{B}\times\mathfrak{B}\times W_1\times\cdots\times W_m$$

を考えよう．W_0 は関係のタイプ，2個の $\mathfrak{B}$ は情報空間自身で2項関係の始点と終点を示し，W_1 以降は2項関係の性質を示す値の集合である．W_0 軸に截片 q で直交する超平面の部分集合を構造と呼ぶ．

$A\subset\mathfrak{B}$ の元をノードとし，構造 $B\subset\mathfrak{W}$ の元をアークとする有向グラフが考えられ，この有向グラフの性質を使って構造の複雑さを分類することができる．順序関係や木構造，ネットワーク構造などの概念は一般に使い方がまちまちで，しばしば議論の混乱の種になるが，グラフとしての形式的分類によって，第Ⅰ型，第Ⅱ型，第Ⅲ型の厳密な定義ができる．また，階層構造の定義も重要である．

さて，$\mathfrak{W}$をそのままの形で扱うことはできないからこの表現についての工夫が必要となる．これを何らかの方法で $\mathfrak{B}$ あるいはその物理的表現に埋め込む方法を考えるのである．基本的には三つの手段がある．その第一は，$\mathfrak{B}$ をその見出しの集合 V_I で置き換えるものである．各点に陰に与えられた見出しとしてそのポインタの集合 V_P を用いることが多い．

$$\mathfrak{W}=W_0\times V_I\times V_I\times W_1\times\cdots\times W_m$$

はそのままの形で $\mathfrak{B}$ の部分空間となるから，レコード間関係は，始点，終点の見出しを持ったレコードとして扱えることになる．このようなレコードの集合を構造ファイルといおう．

第二の方法は，始点レコードに終点レコードの見出し，あるいは終点レコードに始点レコードの見出しを与える方式である．一般にリスト表示と呼ばれる．注意すべき点はこうして付加される属性の値集合が V_I でなくその冪集合 2^{V_I} となることである．2^{V_I} の元は可変長のポインタ配列である．可変長配列を含む可変長レコードをそのまま表示してもよいが，多くの場合レコードの固定長を保つために種々の工夫がなされ，それぞれに応じてリスト表示の変型ができる．この変型には，いくつかのファイルと構造の組と別のいくつかのファイルと構造の組の論理的同値関係が利用される．

第三の方法はレコードの物理的順序配置による方法で第Ⅰ型構造のみに適用される．

第二，第三の方法では原則として W_1, …, W_m に対応する値は表示されない．

5. レコード関数

データ構造の抽象モデルはデータベース・ファイルの設計の基礎を与え，さらにデータベース・マネジメント・システムにあってはデータ記述言語とそのプロセサの設計指針となる．同様にして，情報空間を対象に行われるデータ処理の構造を考える．

データ処理の最少単位は種々の値集合の上で定義されている演算の実行であるが，情報空間の中ではこれらの演算を1個ないし複数個のレコードの組に対して計算されるある関数の値を求めることと解釈される．そのベースとなるものは，定数関数つまり $\mathfrak{B}$ のすべての点 x に対しある定数を与えるものと，点の k 座標，つまり k 番目の属性値，および $\mathfrak{B}$ のある部分集合 A に対して点が属しているか否かにより真偽値を与える関数である．これを基本点関数という．

値集合の上で定義された種々の演算はレコード関数の上の演算に拡張される．このような演算を使って基本点関数からもっと複雑な種々の点関数が生成される．いくつかの点の順序づけられた組を線というが，線関数の生成も基本点関数を種々の演算子で結合することによって行われる．リレーショナル・カルキュラスは論理線関数のことである．その生成に定量子が用いられることもある．点の順序を無視した集合を領域というが，点関数に Σ, Π, 最大，最小，平均，分散

や∧，∨などの演算子を適用して領域関数を生成することができる．

基本点関数と演算子および演算順序を制御する記号の組とある生成文法に従って生成されたレコード関数全体は与えられたシステムの保有する計算能力を示すものと考えられる．

6. 情報処理のための集合演算

情報処理とは情報空間の中の一つの順序づけられた領域の組(入力) から他の順序づけられた領域の組(出力) への変換であると考えられる．この変換は複数個の入力領域から1個の出力領域への変換を組み合わせたものであることは容易に理解できる．したがって，情報処理の本質は m 項の集合演算であるといえる．このような集合演算をある有限個の集合演算の手順的組み合わせとして表現できないだろうか．

情報代数で与えた5種の集合演算

(1) $u(A_1, A_2)=A_1 \cup A_2$

(2) $i(A_1, A_2)=A_1 \cap A_2$

(3) $d(A_1, A_2)=A_1-A_2$

(4) $b[\lambda, \beta](A_1, A_2, \cdots, A_m)$
$=\beta(\{l \mid l \in A_1 \times A_2 \times \cdots \times A_m \wedge \lambda(l)=\text{'true'}\})$

(5) $g[\rho, \gamma](A)=\gamma(A/\rho)$

に演算順序制御のための操作

(6) If $A=\phi$, then ___, else ___.

を加えたものは基本集合演算の組として適当なものである．u, i, d はそれぞれ合併集合，共通集合，差集合を求める通常の2項集合演算である．b はバンドル操作と呼ばれる．λ は $A_1 \times A_2 \times \cdots \times A_m$ で定義された論理点関数，β は

$$\beta=(\lambda_1, \lambda_2, \cdots, \lambda_n)$$
$$\lambda_k : A_1 \times A_2 \times \cdots \times A_m \rightarrow V_k$$

の形のもので点生成関数と呼ばれる．Codd の ALPHA 言語では λ がリレーショナル・カルキュラス，β がターゲット・リストに相当する．$m=1$ の場合，k は λ を検索条件とする検索操作となるが，$m \geq 2$ の場合はいくつかの領域から条件 λ に合致する点を1個ずつ取り出した順序づけられた点の組を作るものでファイルの照合操作に相当する．また λ が述語論理のある形であるところから，自然言語へのインターフェイスとしての期待もあり，さらにデータベース上への推論機能の導入への糸口ともなるものである．b 操作はかなり複雑なものであるが，これをよりエレメンタリな演算に分割する方法がデータベースの検索問題となる．

g はグランプ操作と呼ばれる．ρ は A の上で定義された任意の点関数で，A を ρ の値で類別して得られたクラスの集合が A/ρ である．γ は

$$\gamma=(\mu_1, \mu_2, \cdots, \mu_n)$$
$$\mu_k : 2^A \rightarrow V_k$$

の形の点生成関数である．g はサマライズ操作を代表しているものと考えてよい．

2^B は u, i, d, k, g の5種の演算が定義された代数系であり，これを情報代数系と呼ぶことにしよう．これでデータベースの基本的なアブストラクションができた．

7. レコード間関係を使った検索

B を A の上で定義された構造とするとき，しばしばつぎの6種の検索操作が必要となる．

(1) $s_1(A, B)=\{y \mid o(y) \in A \wedge y \in B\}$

(2) $s_2(A, B)=\{y \mid \delta(y) \in A \wedge y \in B\}$

(3) $s_3(A, B)=\{\delta(y) \mid o(y) \in A \wedge y \in B\}$

(4) $s_4(A, B)=\{o(y) \mid \delta(y) \in A \wedge y \in B\}$

(5) $s_5(A, B)=\{x \mid x \in A \wedge \exists y \in B(\delta(y) \neq x)\}$

(6) $s_6(A, B)=\{x \mid x \in A \wedge \forall y \in B(o(y) \neq x)\}$

$o(y)$, $\delta(y)$ はそれぞれ2項関係 y の始点，終点である．これらはレコード間2項関係に沿って行われる検索である．構造が **4.** で述べたもののうちどの方式で表現されているかによって異なるが，第Ⅰ型構造が物理的なレコードの順序配置で表示されている場合を除いてはこれらの操作は前節の b 操作の特殊な場合となる．しかしながらこの形の検索は頻繁に利用されるものであり，これを前章の基本集合演算に加えておいた方がよいかもしれない．事実，ほとんどのホスト言語システムでは b 操作のうち，これらの操作だけに注目している．

8. 言語設計の立場から

6. および **7.** で情報空間の上でのデータ処理を表現する演算をあきらかにした．これらは，データ処理を記述する高水準言語の一つの候補となる．実際，カジュアル・ユーザが会話形処理に当たって利用する会話形言語としては，この集合論的モデルに基づく言語が適切であると思われる．

しかし，このレベルの言語は一方でバッチ処理の記述のために適当なものでない．定形的なバッチ処理はさらに非手順的なパラメータ言語で指示することが望ましい．また一方においてプログラマが種々のプログ

ラムを開発するために用いるための言語としても適当でない．このレベルの言語の使用は効率の悪いデータ処理をもたらすものである．

前者の解決にはより高水準な言語の開発を必要とし，後者のためにはより低水準のものが欲しい．

高水準言語としてはファイル・ユーティリティやレポート・ライタの開発とその制御言語としての非手順言語が一つの方向であろう．さらにユーザの開発するルーチン業務用バッチ・プログラムの制御コマンドがこの言語に容易に組み込めるような仕組みがあるとうまい．残念ながらこの水準で有限個のコマンドですべてのデータ処理をカバーすることはできない．

高水準言語の別な方向として自然言語がある．可能ならば推論機能を含む事実検索システムと併せて追求したいテーマである．

下位言語への方向は計算機の内部，外部の両記憶装置の相違を意識して，基本集合演算をいわゆるパイプト・モードの操作に分解することによって得られる．基本集合演算のそれぞれからいくつかのパイプト・モード操作が得られ，その一部はデータベース・サブルーチンとして，一部は既存のホスト言語によって実現可能である．検索操作に関しては b 操作の分割によって得られるものと，s_k 操作の分割によって得られるものがあるが，既存のホスト言語システムの多くは後者しか持っていない．b 操作あるいはそのパイプト・モード操作のパラメータとして与えられる検索条件の評価，そしてその検索を実行する最適アルゴリズムの決定がかなり厄介であるが，これをホスト言語に委ねることができないのが辛いところである．

情報空間モデルの適用によって，理論的な裏付けを持った各レベルのデータ操作言語とそのプロセッサの設計が可能となる．

9. FORIMS プロジェクト

1968 年前後はいわゆる MIS 論議の盛んなころであった．混とんとした議論の中で筆者は情報検索との関連と当時ようやく話題となりつつあったデータベース・マネジメント・システム（この名前も未だ定着していなかった）に目をつけ，IDS，IMS，TDMS，MARK IV などの既存あるいは開発途上のシステムを調査し，より優れた機能を目指して FORIMS フェーズ 0 の設計を行った．1969 年(株) 日本ユニバック総合研究所の設立とともにプロジェクトを設定，1970 年にホスト言語レベルと非手順言語レベルの両者を持ち，階層第Ⅱ型構造を扱うフェーズ 0 を完成した．フェーズ 0 は磁気テープにおかれたデータベースを前提としたが，ついで直接アクセス記憶装置と任意の構造を扱うフェーズ 1 の開発にかかり，1972 年に完成した．

当初は，UNIVAC 1100 シリーズのための標準ソフトウェアを目指していたが，DBTG 案に基づく DMS 1100の開発が進められてきたので，当初の方針を変更し，むしろ高度の機能のテスト・ベッドとしての実験システム作りを主眼とし，会話型処理のための集合演算ステートメントを持つフェーズ 2 の開発を目標として 1973 年にフェーズ 2.1 が完成している．

このシステムの特色は情報空間モデルをよりどころとしているところである．逆にモデルの開発過程でプロジェクトからの刺激がかなりとり入れられた．とくにモデルの物理的表現についてのノウ・ハウの蓄積が大きい．

FORIMS は基本ホスト言語として FORTRAN を採用したが，これは既存の膨大な FORTRAN ライブラリをシステムに取り入れることを容易にするためであった．データベース・サブルーチンは FORTRAN のレコード入出力を形成するが，この中に任意の検索条件（但し定量子の扱いは未だインプリメントされていない）による条件検索，任意の構造に沿った検索機能が入っている．構造は構造ファイルによる取り扱いを標準なものとしているが，2 層の階層第Ⅱ型構造をリング表示で扱う機能を追加すべきかも知れないと思っている．会話型言語は基本集合演算を反映しているものである．

最上位の非手順言語はファイルの新設，更新，レポート・ライタの他，データベース管理者用の種々のユーティリティ・プログラムのためのコマンドがある．その一つにマクロ・プロセッサがあり，中位，上位言語の拡張可能性を与えている．

FORIMS は種々の情報システムのテスト・ベッドとして研究所内で利用されている．

10. まとめ

情報空間モデルおよび FORIMS プロジェクトで蓄積されたノウ・ハウは，とくに大規模システムの設計に役立てられる．過去の出力オリエンテッドな設計に対し，ファイル・オリエンテッドなデータベース的アプローチを与えることによって，システム開発のコストを最少にとどめながら，融通性に富む“軟い”設計

の手段を提供するわけである．これは必ずしも既存のデータベース・マネジメント・システムを使うことを意味しない．むしろシステム設計の結果，たまたま利用できるデータベース・マネジメント・システムがあれば採用するという態度を保有すべきである．

扱うべき情報とその上のデータ処理の論理的構造を考え，これを表現する最適な物理的手段を考え，多段の言語レベルとしてインプリメントして行くデータベース的アプローチは，いわゆるトップ・ダウン・アプローチとは異なるものであるが，システム工学の一つの方法として定着して然るべきと考える．

FORIMS プロジェクトを一貫してリードした千葉君を始めプロジェクトのメンバー，物心両面で長期間サポート頂いた日本ユニバック（株）に感謝したい．

11. 情報空間モデルと FORIMS についての文献

情報空間モデル全体については論文 6)～9) がある．データベース検索問題については論文 10)，11) を参照されたい．論文 12) は条件検索の，論文 13) は構造に沿った検索の拡張を論じている．データ構造の物理的表現については論文 14)，汎用データベース・マネジメント・システムの設計については論文 15) がまとめられた．

情報空間モデルを使ってリレーショナル・モデルと DBTG モデルの関連を述べたもの[16]もある．

FORIMS プロジェクトについては解説[17]，システム仕様については仕様書[18),19)]を参照されたい．その他，システムの故障回復についての論文[20)]がある．

大規模情報システムとして国土情報システムや文献情報システムにデータベース的アプローチを採用した例もいくつかあるが，ここでは割愛する．

参 考 文 献

1) CODASYL Development Committee: An Information Algebra: Phase I Report, Comm ACM Vol. 5, No. 4, pp. 190～204 (1962).
2) E.F. Codd: A Relational Model of Data for Large Shared Data Banks, Comm ACM Vol. 13, No. 6, pp. 377～387 (1970).
3) M. E. Senko et al.: Data Structure and Accessing in Data Base Systems, IBM System Journal Vol. 12, No. 1, pp. 30～91 (1973).
4) J. R. Abrial: Data Semantics, Data Base Management, pp. 1～59, North-Holland (1974).
5) CODASYL Data Base Task Group: April 71 Report (1971).
6) I. Kobayashi: A Formalism of Information and Information Processing Structure; Revised Report, The Soken Kiyo Vol. 5, No. 1, pp. 57～122 (1975).
7) I. Kobayashi: Information and Information Processing Structure, Information Systems Vol. 1, No. 2, Perqamon Press pp. 39～49 (1975).
8) I. Kobayashi: An Information Space Model for Data Bases, Nippon Univac Sogo Kenkyusho, Inc., RM-030 (1976).
9) 小林功武：データベースの情報空間モデル，日本ユニバック総合研究所 (1976).
10) I. Kobayashi: An Optimal Data Base Search Strategy for Retrievals on One file, to appear in The Soken Kiyo Vol. 6, No. 2 (1976).
11) I. Kobayashi: A Data Base Search Strategy for Retrievals on Two or More Files, to appear in The Soken Kiyo Vol. 6, No. 2 (1976).
12) I. Kobayashi: Some Enhancements on Settheoretic Data Manipulation Languages, to appear in The Soken Kiyo Vol. 6, No. 2 (1976).
13) Y. Chiba: An Extension of Structural Retrieval in Data Base System, Proc. nd USA-Japan Computer Conf., pp. 359～364 (1975).
14) I. Kobayashi: Physical Representation of Information Structure, The Soken Kiyo Vol. 5, No. 1, pp. 123～158 (1975).
15) I. Kobayashi: Toward a Better Design of Generalized Data Base Management Reptems, The Soken Kiyo Vol. 6, No. 1 (1976).
16) I. Kobayashi: DBTG Model, Relational Model and Information Space Model of the Information Structure, Proc. nd USA-Japan Computer Conf., pp. 329～334 (1975).
17) I. Kobayashi: Introduction to FORIMS Project, The Soken Kiyo Vol. 5, No. 1, pp. 51～56 (1975).
18) K. Kohri and Y. Chiba: FORIMS Phase 2 Design Specification, The Soken Kiyo, Vol. 5, No. 1, pp. 177～210 (1975).
19) 日本ユニバック総合研究所： FORIMS Phase 2.1, S 5050170 (1973).
20) S. Iizuka and Y. Chiba: GERM: General Error Recovery Model of Shared Data Base, The Soken Kiyo Vol. 5, No. 1, pp. 211～226 (1975).

（昭和 51 年 5 月 27 日受付）