



ANSI/SPARC データ・ベース・システム概念*

石田 喬也**

1. まえがき

情報という極めて人間系に密着したものを管理対象とするデータ・ベース管理システム (以後 DBMS) は、単にそのハードウェアやソフトウェアだけでなく、それに参与する様々な人の役割と目的にまで立ち入った、トータルの視点で論じられるべきである。

CODASYL DBTG レポート¹⁾を頂点とする DBMS に関するこれまでの各種の提案や実用製品は、データ記述言語、データ操作言語、データ構造などの裏付けにより、DBMS を具体化して示してきた。これらは確かに、従来のファイル・システムから処理の次元を一段高めた点で、大きく貢献したといえる。しかし見方を変えれば、具体化を急ぐ余り DBMS の本質議論を短絡してしまった感を否めない。また最近脚光を浴びているリレーショナル・モデル²⁾にしても、データ構造やデータ操作言語に議論の焦点があり、DBMS の全体を論じるものではない。

このような現状では、DBMS の発展方向、あるいはまたその標準化の方向を正しく描くことはできない。すなわち、DBMS それ自体のシステム概念の確立こそ、今なによりも急がれているといわざるを得ない。ANSI/X 3/SPARC の DBMS 問題調査グループの成果が注目され、そしてまたここで取り上げようとするのはまさにこの点にある。

この SPARC グループは、もともとは DBMS 標準化可能性を調査する目的で発足されたが、結局は DBMS 全体のアーキテクチャのあるべき姿を求める、原点にもどった議論の展開となった経緯は興味深い。その第 1 次中間レポート³⁾で DBMS の機能要素構成の全体図を描いて後、第 2 次中間レポート⁴⁾でそれぞれの機能要素および機能要素間インタフェースを詳細

に論じている。またその活動状況を報告する論文も現われてきている^{5),6)}。

未だ最終レポートには至っていないが、第 2 次中間レポートに本質的部分はすでに盛り込まれているとされており、以下ではこのレポートに基づき、ANSI/SPARC が指向する DBMS のシステム思想について述べることにする。

2. 情報表現のモデル化

現実世界の情報をいかに正しくかつ効率的に計算機上に写像するか、すなわち情報表現モデルを確立することが DBMS 構築の基礎である。

現実世界に存在するものを実在 (Entity) と呼ぶことにすれば、実在と実在に関する事実を極力ありのままに記号化表現することが、そのための第一歩である。ひとつの企業にとって関心ある実在と実在に関する事実の記号化表現データのすべての集まりを、その企業にとっての概念モデルと名付ける。

概念モデルを構成するデータは、あくまで概念上の存在であり、その実体 (値) は計算機の内部記憶空間内に位置を占める。概念モデルのデータが内部記憶空間上に写像されたデータのすべての集まりを、その企業にとっての内部モデルと名付ける。ただしここで想定される内部記憶空間は、あくまで論理的なものであり、実際には内部モデルの構成データは、物理記憶媒体上にその物理特性に従い記録される。

一方企業における特定のアプリケーションにとっては、関心があるのは概念モデルの構成データの特定部分集合であるにすぎない。この部分集合を、そのアプリケーションにとっての外部モデルと名付ける。

以上に挙げた 3 種の情報表現モデルに従えば、ひとつの企業にとってのデータ・ベースは、ひとつの概念モデルとひとつの内部モデルと、その企業のアプリケーションのタイプの数だけの外部モデルによって表現されるといえる。

* System philosophy for data base management systems proposed by ANSI/SPARC by Takaya ISHIDA (Computer Works, Mitsubishi Electric Corp.)

** 三菱電機(株)計算機製作所

各モデルはより正確には、その構成データの集合と、構成データ自体をタイプ別に記述するメタ・データであるデスク립タの集合から成り立つ。このデスク립タ集合をスキーマと呼び、各モデルに対応して、概念スキーマ、内部スキーマ、外部スキーマがそれぞれ定義される。

この3レベルの情報表現モデルあるいはスキーマ思想こそ、ANSI/SPARC DBMS システム概念を最も特徴づけるポイントとなっている。これは明らかに、GUIDE/SHARE DBMS 要望レポート^{7),8)}における、論理一実在一蓄積の3レコード・タイプ間データ写像方式提案を継承発展せしめたものである。とりわけ概念モデルをシステム内に明確に位置づけて、これまで軽視されがちであった情報のセマンティックス表現を情報表現の中核に据えるシステム思想^{9)~11)}に、体系化した点が高く評価される。

概念モデル、特に概念スキーマの存在意義は、次の4点に要約できる。

- (1) すべての情報参照のための、安定した拠り所を提供する。
- (2) 情報が、どこで、誰により、どのようにして利用されるか、という情報管理目的データ・ディクショナリ/ディレクトリ¹²⁾問題に解決方向を与える。
- (3) データの安全性ならびに完全性を保護する上での最終判断基準を提供する。
- (4) 2レベル写像方式により、外部モデルと内部モデルを隔離してデータ独立性を高める。(図-

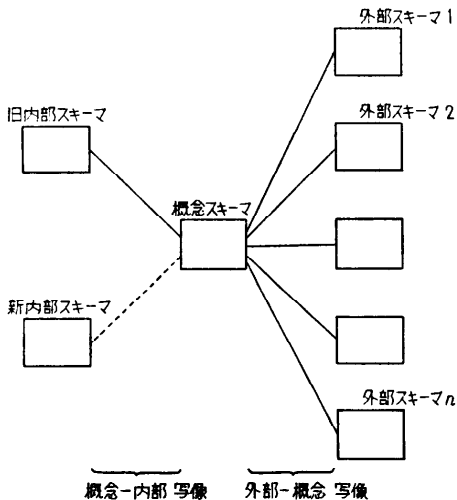


図-1 外部-概念-内部モデル間写像

1参照)。

3. システム・アーキテクチャの全貌

提案されている DBMS アーキテクチャを、主要部分を抜粋して図-2に示す。SPARCグループがこのようなシステム・アーキテクチャを描くに至った動機は、DBMS 構成機能モジュール間のインタフェースを洗い出し、その役割を明確にして、それぞれの標準化可能性を考察することにある。当初は³⁾データの物理レベルの記憶、アクセスの領域にまで立ち入って、41種類に及ぶインタフェースを洗い出した。しかし現在では、物理レベルの領域は蓄積管理と名付ける、データ・ベース管理とは独立したシステムに属するものとする見方によっており、データ・ベース管理の領域に属するものとして、28種類のインタフェースを挙げている。ただしインタフェース番号自体は、当初の番号づけに従っている。

図-2において、六角形はDBMSに参与する人の役割、四角形は構成機能モジュール、(一)太棒印はインタフェース、矢印線は情報の流れ、三角形はデータ・ディクショナリ/ディレクトリ、をそれぞれ表わす。

人の役割は管理者とユーザに大別される。ただし、役割は個々の人とは独立して決められるものであるこ

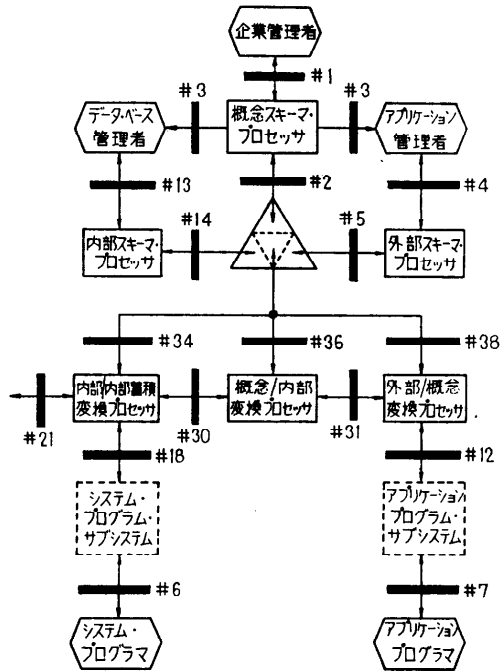


図-2 DBMS アーキテクチャ (抜粋)

とに留意する必要がある。管理者はそれぞれの管理目的に合わせて、必要とするデータとデータ間関係、ならびに安全性、完全性、回復等のための制御方式、さらには、保持データの版、ステータス、ユーザ、責任所在等の管理上の必要項目を記述する情報（スキーマ情報）をシステムに与え、またそれを維持する役割を果たす。企業管理者は概念モデルを、データ・ベース管理者は内部モデルを、アプリケーション管理者は外部モデルを、それぞれの管理対象とする。またユーザについては図-2では、内部モデルのユーザをシステム・プログラマ、外部モデルのユーザをアプリケーション・プログラマ、でそれぞれ代表させている。

概念スキーマ・プロセッサ、内部スキーマ・プロセッサ、外部スキーマ・プロセッサ、はそれぞれ概念、内部、外部の各モデルに対応してスキーマを、生成、更新あるいは表示する機能を受け持つ。データ・ディクショナリ/ディレクトリは各スキーマ情報を集中保管することにより、システム構成要素全体を統合するかなめの役割を果たす。

アプリケーション・プログラマ（プログラム）からのデータ・アクセス要求は、DBMS 外のアプリケーション・プログラム・サブシステムを経て、外部/概念変換プロセッサにより概念モデル・レベルに変換され、それが概念/内部変換プロセッサにより内部モデル・レベルに変換され、さらにそれが内部/内部蓄積プロセッサにより蓄積管理システムへのデータ・アクセス要求に変換される。蓄積管理システム下で実際にアクセスされたデータは、これと逆の流れをたどって、アプリケーション・プログラマに返される。

システム・プログラマ（プログラム）からのデータ・アクセス要求は、DBMS 外のシステム・プログラム・サブシステムを経て直接、内部/内部蓄積変換プロセッサに渡される。アプリケーション・プログラムにとっての、上述したデータ・アクセスにおける迂回は、それだけプログラムのデータ独立性を高める。システム・プログラムにとってのそのルートのバイパスは、データ独立性を犠牲にしつつ、データ・アクセスの効率ときめ細かさを高める上に有効となる。

4. 概念モデルの構成

概念モデルの最小構成単位である概念フィールドは、実在に関する企業にとって関心ある最小不可分の特性を表現する。概念フィールドは、その意味表現上の役割と、その値が選ばれる定義域の2面から定義づ

けられる。定義域自体は、その意味、表現形式、正当性規定、無効規定などによって、任意の値がその定義域に正しく属するか否かを判別できるだけの厳密さをもって定義されていなければならない。さらに概念グループを、一括的に扱う必要がある概念フィールド/概念グループの任意関係結合として定義することができる。アレイ (Array) は概念グループの一例である。

ひとつの実在単位での、概念フィールドあるいは概念グループの全体は、無構造あるいは任意に複雑な構造で関係結合されて概念レコードを形成する。

任意の概念レコードは、それよりはより単純な構成の概念レコードの結合として、したがって一般に概念レコードのネストの形で定義づけることができる。ネストの繰り返しによって到達する基底概念レコードは、リレーショナル・モデルの第3次正規形¹³⁾に類するイメージとなるであろう。

互いに関係しあう概念レコードは、無構造あるいは任意に複雑な構造で関係結合されて、各関係に対応して概念プレックスを形成する。概念プレックスと他の概念レコード/概念プレックスの結合は、さらに新しいタイプの概念プレックスを形成する。DBTG レポートにおけるセット (Set) は、概念プレックスの一例である。

概念レコードあるいは概念プレックスの、実在の特定の集合に対応した、オカレンス(値)・レベルでの関係結合はひとつの概念レコード・セットを形成する。概念レコード・セット自体のオカレンスは、その異なる世代あるいは版である。概念レコード・セットの関係しあうものすべての集合が、概念データ・ベースを形成する。

現実世界で認識される関係は、余りにも多様であり、そのままの形をモデル化することは不可能である。概念モデル（外部モデルと同様）にあっては、関係は識別可能対象の選択道順を示す結合である、とみなす形に単純化して扱われる。ハイアラキ、ネットワークなどは、この意味での関係を具体的に表現する一例である。

概念スキーマ情報は、まず企業管理者によって、概念スキーマ記述言語を使ってそのソース形式（#1: 図-2 のインタフェース番号に対応、以下同様）が記述される。概念スキーマ・プロセッサはそれを受け取り、その記述の文法ならびに矛盾性をチェックした後、それをオブジェクト形式（#2）に変換する。概念/内部変換プロセッサをはじめとして、概念スキーマ

マ情報を必要とするシステムの他の部分には、この形式で知らされる。またその内容を知る必要がある人に対しては、知る権利がある内容のみに限定して、それぞれの使用目的に適した表示形式(#3)に編集されて表示される。特にアプリケーション管理者ならびにデータ・ベース管理者にとっては、このインタフェースは不可欠である。

5. 内部モデルの構成

内部モデルの内部記憶空間としては、限界がない複数の原点を持つ1次元空間が仮想される。この空間には、外部記憶媒体の種類により相違する物理的特性を除いた、アドレス方式、近傍性などの記憶媒体に共通する性能上に関する特性は、ひと通り反映されているとみなされる。

内部モデルの最小構成単位である内部フィールドは、そのオカレンスが空間単位(バイト等)の連続ストリングとして内部記憶空間に位置を占める。内部フィールドのひとつ以上の集まりが、物理的アクセスの単位である内部レコードを構成する。内部レコードのオカレンス・レベルの集合が内部レコード・セットであり、そのひとつ以上の集まりが内部データ・ベースを形成する。

内部データ・ベースを構成する際に考慮されることは、もっぱら経済上あるいは性能上の問題である。そのために内部スキーマでは、空間領域割付方式、データ編成方式、レコード挿入方式、レコード消去方式、レコード検索方式、などが詳細にわたり指定される。

内部記憶空間は互いに素な連続領域であるひとつ以上のスペース区域に分割され、内部レコード・セットは同一タイプの内部レコードに対するデスク립タが一樣な領域であるひとつ以上のフォーム区域に分割されているとみなされる。内部レコードのオカレンスを内部記憶空間上に配する場合には、それに適切なスペース区域とフォーム区域を割りつけなければならない。他の区域からの複写や、意識的な重複レコード保持などがその際性能上工夫されるべき点の一例である。

概念レコード・セットから内部レコード・セットへの写像方式はデータ・ベース管理者により決められる。両者のレコード間での、フィールド対応やレコード・キー変換方式、冗長記憶データに対する波及アルゴリズム、仮想データの实体化アルゴリズム、等がその際規定される。

内部スキーマ情報は、データ・ベース管理者により

ソース形式(#13)で記述され、それが内部スキーマ・プロセッサによりオブジェクト形式(#14)に変換される。それを知る必要があり、知る権利がある人に対しては、使用目的に適した表示形式に編集されて表示される。その主たる利用者はデータ・ベース管理者ならびにシステム・プログラマである。

6. 外部モデルの構成

外部モデルは概念モデルの部分集合空間であるため、構成要素である外部フィールド、外部グループ、外部レコード、外部プレックス、外部レコード・セットはそれぞれ概念モデルの対応構成要素と、それが外部データに関するものであることを除き全く同様に定義される。

各外部スキーマでは、その外部モデルを使用するユーザの言語タイプからは独立した形で標準的に定義されることが望ましい。たとえばユーザがリレーショナル・モデルの流儀で特定の外部モデルを使用したいときでも、その外部スキーマはリレーションを外部レコード・セットに、行を外部レコードに、属性を外部フィールドに、対応づけて標準形表現されるであろう。

外部レコード・セットの概念レコード・セットへの束縛方式としては、外部レコード・セット定義時にそれに束縛する特定の概念レコード・セットを対応づける方式と、外部レコード・セット定義は一般的にしておき、必要時点で特定の概念レコード・セットに束縛する指示を与える方式、の2種類が必要とされる。

外部レコード・セットは、それに束縛される概念レコード・セットの部分集合である条件を満足する限り、構造、表現形式、配列順序などは任意でよい。指定しない特性については、概念データ記述が既定値を与える。

データ独立性は、概念レコードを枢軸として、外部レコードを内部レコードに写像することで実現される。デスク립タ・レベルでこの写像を、プログラム・コンパイル時、プログラム実行開始時、あるいは外部レコード・セットのオープン時に決める形で実現されるのは静的データ独立性である。それはまた、束縛される内部レコード・セットがそれぞれ単一のフォーム区域から構成される場合に限定される。真に望ましいのは、外部レコードの参照の都度その写像を決める方式であり、動的データ独立性を実現する。この方式は複数のフォーム区域から構成される内部レコード・セットへの束縛を可能ならしめる。

外部スキーマ情報は、アプリケーション管理者によりソース形式(#4)で記述され、それが外部スキーマ・プロセッサによりオブジェクト形式(#5)に変換される。個々のユーザには使用言語タイプに応じたフォーマット化がスキーマ・フォーマット変換プロセッサにより自動的に行われて、そのホスト言語形式が渡される。ユーザ・プログラムはそれをそのまま中に複写することにより外部モデルと結合される。

外部モデルのユーザは5種類に大別される。アプリケーション・プログラマには、ホスト言語タイプ別にデータ操作言語インタフェースが提供される。レポート型ユーザには1日の訓練で、問い合わせ型ユーザには1時間の訓練で、更新型ユーザには2ないし4時間の訓練で、またパラメータ型ユーザには単にインプットすべきパラメータの種類とフォーマットの教育で、それぞれの用法をマスターできる程度に易しい問題向きエンド・ユーザ言語インタフェースが必要とされる。

7. む す び

SPARC レポートを、それが提案するシステム・アーキテクチャとその表現モデルに重点を置いてながめてみた。これら以外に、安全性保護、完全性保護、エラー回復、データ独立性などのテーマについても執拗な価値ある解析がなされている。内容の詳細についてはいくつか検討の余地が残されているとはいえ、DBMS 全体を初めて総合的に定義づけたものとしてのこのレポートの功績は非常に大きいと断言できるであろう。ISO における DBMS 標準化の検討が、SPARC レポートをその出発点とする方向に向っている¹⁴⁾のも、極めて自然な動きであるように思われる。

最後にこの執筆にあたり、SPARC 関係資料の入手等で色々御助力を得た、情報処理学会データ・ベース言語研究委員会、藤中恵委員長、植村俊亮幹事に深く感謝致します。

参 考 文 献

1) Data Base Task Group Report to the CODASYL Programming Language Committee, ACM,

New York (1971).

- 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) ANSI/X 3/SPARC Study Group on Data Base Management Systems Interim Report 74-01-25, Draft prepared by C. W. Bachman (1974).
- 4) ANSI/X 3/SPARC Study Group on Data Base Management Systems Interim Report 75-02-08, ANSI DOC. No. 7514 TS 01 (1975).
- 5) C. W. Bachman: Trends in Database Management, Proceedings of a two-day Symposium "Implementations of CODASYL Data Base Management Proposals", the British Computer Society, pp. 200~222 (1974).
- 6) T. B. Steel: Data Base Standardization—A Status Report, Proc. ACM-SIGMOD Conference, San Jose, Calif., pp. 149~156 (1975).
- 7) The Joint GUIDE and SHARE Data Base Requirements Group: Requirements for a Data Base Management System, GUIDE, Chicago (1970).
- 8) 石田喬也: データ・ベースへの要望について, 情報処理, Vol. 14, No. 2, pp. 118~126 (1973).
- 9) M. E. Senko: Specification of stored data structures and desired output results in DIAM II with FORAL, International Conference on Very Large Data Bases, Framingham, Massachusetts (1975).
- 10) P. P. Chen: The Entity-Relationship Model—Toward a Unified View of Data, ACM Transactions on Data Base Systems, Vol. 1, No. 1, pp. 9~36 (1976).
- 11) 石田喬也: データ・ベースのセマンティックスとインプリメンテーション, 情報処理学会データ・ベース研究会資料, 74-10 (1975).
- 12) P. P. Uhrowczik: Data Dictionary/Directories, IBM Systems Journal, Vol. 12, No. 4, pp. 332~350 (1973).
- 13) E. F. Codd: Further normalization of the data base relational model, Courant Computer Science Symposia, Vol. 6, Data Base Systems, Prentice-Hall, New York (1971).
- 14) 情報処理学会データ・ベース言語研究委員会報告, 情報処理, Vol. 17, No. 6 (1976).
(昭和51年5月31日受付)