解　説

高水準データモデル

ポスト・リレーショナルをめぐるデータモデルの研究動向†

宇田川佳久††

1. はじめに

データベースの目的は，必要なデータを，必要とする利用者にだけ，タイムリに提供することである．その萌芽は 1950 年代にまでさかのぼることができる[1]．1960 年代にはデータベースの考え方が普及し，1970 年代からは，データベース・システムモデルの確立[4]，汎用データベース管理システムの開発や利用技術の進歩をみた．ファイルがもっぱら個人によるデータの活用を支援するためのツールであるのに対し，データベースはデータを組織的に活用するためのツールと位置付けることができる．データベースとファイルとを比較した場合，

- データの共用による有効利用を推進できる
- データの標準化を推進できる
- データの冗長性を少なくできる
- データの機密保護に貢献する
- データの一貫性管理が容易になる

といった特徴がある[8]．

データを共同利用する，すなわち，データを組織的に活用するためには，データベースが使いやすいものであることが重要なポイントになる．そのためにデータ構造の単純化，操作演算の簡潔化が追及されてきた．1970 年代のリレーショナル・モデルは，このようなニーズに合致したデータモデルである[2]．リレーショナル型データベースは商用化され，多くのアプリケーションに適用されてきた．特に，固定長の文字・数値データを処理の単位とするアプリケーション（在庫情報管理，人事情報管理など）に適用したときに威力を発揮している．

1980 年代になると，計算機，とりわけワークステーションとパーソナルコンピュータが普及し，アプリケーションも CAD, OA やグラフィクスに及んできた．当然，扱われるデータも質的に変化してきた．従来の文字・数値データに加え，不定長のテキストデータ，図形・画像・音声データ，さらに，これらのデータを組み合わせた構造をもったデータが処理の対象になってきたわけである．リレーショナル・データモデルによってこれらのデータを扱うことは難しく，新しいデータモデルの研究が盛んになってきた[3),14)]．本文では，このように複雑かつ多様化したデータを扱うデータモデルを総称して高水準データモデルと呼んでいる．

本文 **2.** では，データベースの基本機能について述べる．**3.** では，データモデルの変遷と高水準データモデルが解決すべき課題を示している．**4.** では，データベース管理システムの構成と分散計算機環境への対応について述べている．

2. データモデルとデータベースの基本機能

データベースは，複数のユーザまたは応用プログラムによって共同利用されるデータの集まりと考えることができる．ファイルが，もっぱら，個人が使うデータを扱うのに対し，データベースは，複数の利用者からのデータの共同利用を目標としていることが，両者の根本的な違いである．本をデータに例えるなら，個人の机の上に散在している本の山がファイルに相当し，図書館に整然と並べられている本の集まりがデータベースに相当するわけである．

データの共同利用を促進するためには，ファイル・システムよりも複雑な仕掛けが必要になる．具体的には，次のような基本機能がデータベースに提供されている．

† High-Level Data Model——Research Trends in Data Models toward Post-Relational Model by Yoshihisa UDAGAWA (Information Systems and Electronics Development Lab., MITSUBISHI ELECTRIC Corp.).

†† 三菱電機(株)情報電子研究所

（1） データモデルの提供．管理しようとするデータの構造（データ項目とデータ項目どうしの関連性など）を表現する手段と，データを操作する演算を合わせたものをデータモデルと呼んでいる．図書館に例えるなら，本の分類や並べ方，それに，借用・返却の手順を合わせたものがデータモデルに対応する．

データモデルが必要なのは，少なくとも二つの理由がある．第1に，データをアクセスするためにデータモデルが必要だからである．データは，ビットパターンとして計算機に記憶されているが，このビットパターンは，利用者にとって意味のあるデータ表現とはほど遠いものである．データモデルは，計算機に記憶されているビットパターンを，利用者にとって意味のあるデータ表現として操作することを可能にする役割を果たしている．第2に，不正なデータ・アクセスからデータベースを守るのに役立つからである．文字型のデータ領域に数値データを書き込んでしまうようなことは，データモデルによって阻止することができる．

（2） 二次記憶装置に記憶されているデータに対する効率的なアクセス機能．この機能には，データのインデックス付け，空き領域の管理，主記憶とのバッファリング機能などが含まれる．

（3） データの排他アクセス機能．複数の利用者が，同時に同じデータを更新すると，現実的に不都合なことが起きる．これを防ぐために，一時的に，特定の利用者に特定のデータを専有させる機能などが提供されている．

（4） 応用プログラムに対する共通言語インタフェース．応用プログラムは，同一のプログラミング言語で書かれているとは限らない．したがって，個々のプログラミング言語とは別に，データベース専用の言語が提供されている．リレーショナル型データベースの標準言語 SQL[7] がその例．

（5） バックアップとリカバリ機能．データベースは，データの共同倉庫であるから，システム・ダウンと復旧処理への対策が必要になる．そのために，データベースのリカバリのためのユーティリティ・プログラム群が提供されている．

（6） セキュリティ機能．権限のない利用者からアクセスされないように，データに対するアクセス権の管理をする必要がある．また，データベースをダンプし解読される場合の対策として，暗号化してデータを記憶するなどの機能もある．

3. データモデルの変遷

3.1 階層モデル

データベースの考え方は，1960年代前半に確立し，中盤にはデータベース管理システムが商品化されている．このころは，ディスク装置が一般に普及した時期で，計算機の応用分野は，科学技術計算から大量のデータ処理が要求される事務処理分野へと広がりつつあった．データの処理は，主記憶容量の制限から，レコード単位で行われることが多かった．

階層モデルは，データを階層的に記憶し，データへのアクセスもこの階層構造に従って行うものである．**図-1** は，羽田発のフライト・スケジュールの一例を階層モデルで表現したものである．階層モデルでは，データを階層的に記憶しているために，アクセスの順番が画一的になる．たとえば，図-1 では，会社名を指定し，この会社が運行しているフライトの情報（行き先，便名，出発時刻，到着時刻）をアクセスできるように記憶している．しかし，これ以外の順番でデータをアクセスしようとすると，操作が繁雑になり，かつ効率も悪くなる．言い換えれば，階層モデルは，記憶構造に合致したアプリケーションには大変都合が良いものであるが，記憶構造と少しでも違う使い方をしようとすると複雑な操作が必要になる．

3.2 ネットワーク・モデル

図-2 は，図-1と同様のデータをネットワーク・モデルで表現したものである．このモデルは，文字どおり，ネットワーク構造に基づいて現実世

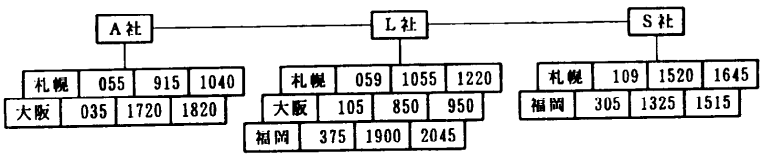

図-1 階層モデルの例

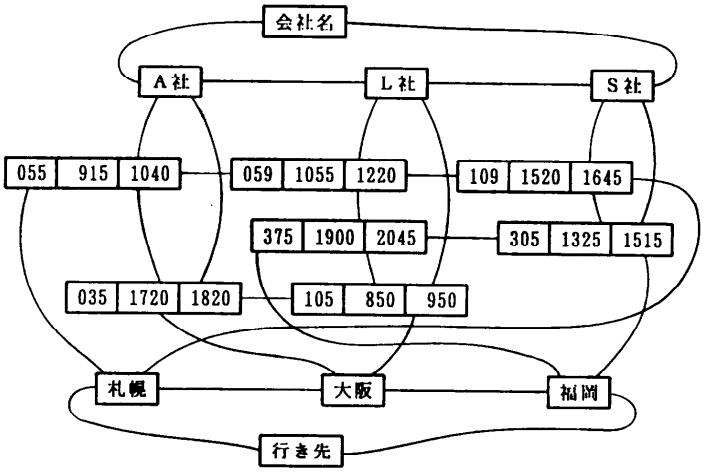

図-2　ネットワーク・モデルの例

界のデータを表現しようとするものである．ネットワーク・モデルは，階層モデルよりも多様なアクセス・パスを設定できる特徴がある．たとえば，図-2 では，会社名と行き先のいずれからでも，フライト情報をアクセスすることができる．

ネットワーク・モデルは，1970 年代を通してデータモデルの主流になった．しかし，データには依然として上下関係があり，上位のデータから順にポインタをたどって，下位のデータをアクセスすることを想定している．したがって，ネットワーク・モデルも階層モデルと同様に，あらかじめ設定された順番とは違った順番でデータをアクセスしようとすると，操作が繁雑になり，かつ効率も悪くなる．

3.3　リレーショナル・モデル

1970 年にリレーショナル・モデルが提案された．1970 年代後半になると，IC メモリが普及し，大容量の主記憶が利用できるようになった．リレーショナル型データベースは，このようなハードウェアの進歩に支えられて発展したデータベースと考えることができよう．階層モデルやネットワーク・モデルが，データ項目どうしを“ポインタ”によって結び付けることによって現実世界のデータを表現しているのに対し，リレーショナル・モデルは，データ項目の“集まり”によって現実世界のデータを表現するものである．**図-3** は，図-1 と同様のデータをリレーショナル・モデルで表現したものである．階層モデルあるいはネットワーク・モデルにおける“ポインタ”の概念は，計算機における“ポインタ”の延長であり，データを検索することは，すなわち，ポインタをたどることであった．しかし，現実世界のデータの間には，ポインタの概念がないことが多い．たとえば，“15：00 から 15：30 に福岡に到着するフライト”といったように，特定の条件を満たすデータの“集まり”を選び出すことのほうが自然であることもある．リレーショナル型データベースは，このような現実世界において自然なデータ操作を可能にしたもので，1970 年代から現在に至るまで研究および製品開発が盛んである．

3.4　リレーショナル・モデルの限界

1980 年代に入ると高度なグラフィック機能を備えたワークステーション (WS) が普及してきた．WS は，構内ネットワーク (LAN) で接続されていて，高速なデータ交換が可能である．このような計算機環境の変化は，アプリケーションの発展をうながし，CAD, OA (文書処理) やグラフィック関係のソフトウェアが充実してきた．技術的には，分散処理と扱うデータのマルチメディア化が促進されている．一方，1980 年代はじめには，リレーショナル型データベースの普及期を迎え，多くのアプリケーションに適用された．この過程で，リレーショナル・モデルの限界も明らかになってきた[8),14)]．

第 1 に，データ構造が固定的であり，データの一貫性も限定された範囲でのみ行われていた．リレーショナル・モデルは事務処理を前提にした，ただ一種類のデータ構造（リレーション）しか提供していない．リレーションというデータ構造

フライト・スケジュール

会社名	行き先	便名	出発時刻	到着時刻
A社	札　幌	055	9　15	10　40
L社	札　幌	059	10　55	12　20
S社	札　幌	109	15　20	16　45
A社	大　阪	035	17　20	18　20
L社	大　阪	105	8　50	9　50
L社	福　岡	305	13　25	15　15
S社	福　岡	375	19　00	20　45

図-3　リレーショナル・モデルの例

で，データベース化しようとする対象をすべて表現しなければならない．一方，CAD や OA といったアプリケーションで扱うデータは，リレーションの中にリレーションが入った入れ子構造をしていたり，データの階層が明示的に決まっていたり，データの順番に意味があったりすることがある．こういったデータをリレーションによって表現することは，非効率的であったり，困難であることさえある．また，リレーショナル・モデルにおけるデータの一貫性は，リレーションの特定の項目のデータ値に関してのみ管理されている[7]．一方，アプリケーションによっては，不定個のリレーションに対してデータの一貫性を管理しなければならなかったり，一般的な手続きを使わなければ一貫性条件を記述できないこともある[8]．

第2は，データ型の限界である．リレーショナル・モデルでは，扱うデータが固定長の文字・数値データに限られているために，図形・画像・音声といった，いわゆる，マルチメディア・データの扱いに対応することが難しいことが指摘されている．単に，マルチメディア・データを記憶し，表示するだけであれば，非常に長いビット列（たとえば1メガ・ビット）を一つのデータ型として扱うことで対応できる．しかし，図形・画像データに限っても，データ型とともにデータの構造を含めた対応が必要であることが多い．たとえば，回路図には同形のモジュールが何回も現れ，モジュールはさらに小さなモジュールによって表現される．したがって，回路図を表示するときも，モジュール・レベルの表示から，ゲート・レベルの表示まで，いくつかのレベルが考えられる．マルチメディア・データの扱いは，高水準データモデルの重要課題になっている．

3.5 高水準データモデル

本文では，リレーショナル・モデルの限界を克服することを目指したデータモデルを総称して高水準データモデルと呼んでいるが，具体的には，以下のようなデータモデルが含まれている．

（1） 関数データモデル

現実世界のデータを実体と関数によって表現しようとするデータモデルである[11]．実体は，現実世界の識別できる“もの”に対応する．実体に関数を適用することによって，実体の属性を操作することができる．関数を追加することによって，データモデルを拡張することができることから，拡張可能データモデルと呼んでいる研究者もいる[6]．

（2） オブジェクト指向データモデル

オブジェクト指向プログラミング[12]における概念（クラス，インスタンス，継承，メソッド）を取り入れたデータモデルである．リレーショナル・モデルのような数学的な定式化は行われていないものの，1989年に出された，オブジェクト指向データベース宣言[5]によって，オブジェクト指向データモデルの枠組みが定められている．

（3） マルチメディア・データモデル

テキスト，音声，図形，画像，ビデオといったマルチメディア・データ[9]を扱うデータモデルを総称して，マルチメディア・データモデルと呼んでいる．現状では，文字・数値データに加え，画像や図形データを扱うものが多く，CAD データや地図データの管理に適用されている[10]．

（4） 演繹データベース

データから別のデータを導出するルールやデータの一貫性管理に関するルールもデータベースに蓄えることにより，柔軟かつ高度な問合せに対応しようとするデータモデルである[14]．

4. データベース管理システムの変遷

4.1 ファイル・システムと OS

2. では，データベースの基本機能について述べたが，このような機能を実現するもの（通常ソフトウェア）を，データベース管理システム (DBMS) と呼んでいる．DBMS の発展過程をソフトウェア構成の側面から説明すると次のようになる．計算機が利用され始めたころ，利用者は必要な計算機リソースをすべて自分のプログラムで管理しなければならなかった（図-4）．二次記憶装置に記憶されているファイルを使う場合も例外ではない．利用者は，ファイルの物理的な性質（記憶ブロックの物理的な位置や記憶容量，ヘッドの移動指定），ファイルの論理的な性質（データ項目の数や各データ項目の型など）を操作するプログラムを書いた後に，応用プログラムを書くことができた．

その後，ファイルの物理的な性質は，計算機システム固有のものであるから，共通プログラムとして提供されるようになった．計算機リソースを

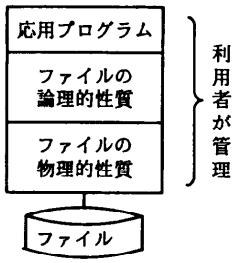

図-4　ファイル・システムによるデータ管理

一括管理する目的で，オペレーティング・システム（OS）が開発され，ファイルの物理的な性質はOSによって管理されるようになったわけである（図-5）．OSの出現によってもたらされたもう一つの効果は，プログラムの多重処理である．一つの計算機で，見かけ上，同時に複数のプログラムを動かせるようになった．ファイル処理も見かけ上，同時に複数実行できるようになった．しかし，注意しなければならないのは，OSのデータ管理機能はデータの共同利用を目的としているわけではないことである．ファイルに記憶されているデータは，通常一個の特定のプログラムに割り当てられている．OSは，そのようなプログラムを複数，同時に実行させているのである．

4.2　データベース管理システム

データの共同利用を可能にするために，データベース管理システムが開発された（図-6）．その基本機能は，2. で述べたとおりである．利用者からみたときの特徴は，データベース中のデータを操作するための言語を，特定のアプリケーションとは独立して提供していることである．この言語は大きく2種類に分類されている．一つは，データベースの構造を定義するデータベース定義言語（DDL），もう一つは，データの操作（検索，挿入，更新，削除）をするデータベース操作言語（DML）である．1975年に，データベース・システムのアーキテクチャを定めたANSIのレポート[4]によれば，DDLは，データベースの運用管理をする個人（あるいは組織）が使う言語であり，DMLはデータベースの利用者が使う言語と位置付けられていた．しかし，実務では，データベースの構造を利用者が定義したいこともある．1989年に標準化されたリレーショナル型データベース言語SQLでは，一つの言語体系の中でデータ構造の定義と操作ができるようになっている[7]．

4.3　ワークステーションの普及

WSの普及は，データベースのアーキテクチャにも影響を与えている．従来のアーキテクチャでは，1台の計算機で応用プログラムからデータアクセスまでを実行するものであった．これに対し，複数のWSをLANで結んだ，いわゆる分散環境では，複数の計算機によってプログラムの実行を分担することが可能になる．現在，最も一般的なのは，二つのカテゴリの計算機を使った，いわゆる，クライアント・サーバ・モデルに基づいて処理する方式である．

図-7に示したように，クライアント側では，応用プログラムの実行とデータの論理的性質を管理する．一方，サーバ側は，DBMSコマンドの実行とファイルの物理的性質を管理している．ここで問題になるのは，DBMSコマンドの最適化やデータの一貫性管理をどちら側で実行するかとい

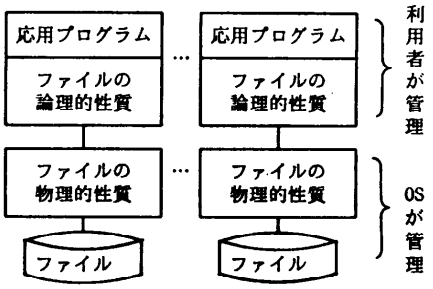

図-5　OSによるデータ管理

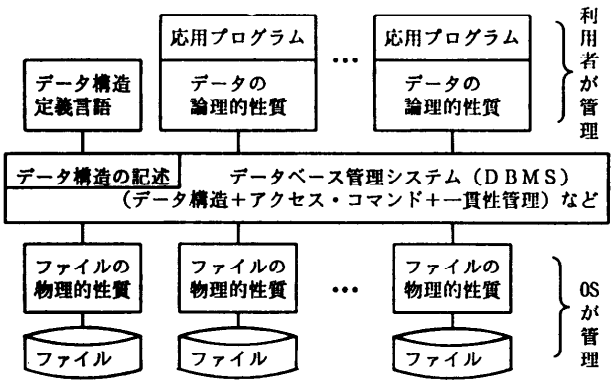

図-6　DBMSによるデータ管理

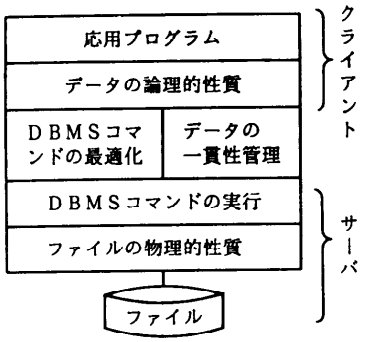

図-7 クライアント・サーバ・モデルによるデータベース処理の例

うことである．この判断は，クライアントとサーバの処理能力や，LAN の通信速度に依存するものであるが，近年，高速なデータベース・サーバが開発され，DBMS コマンドの最適化やデータの一貫性管理をサーバ側で実行できるようになりつつある．

分散処理アーキテクチャを採用することにより，データベースの処理能力および機能の拡張が，集中処理よりも容易になると期待されている[13]．複数の計算機で処理することにより，性能が確保できれば，より複雑なデータの一貫性管理が可能になる．また，新しいデータ型の追加も，そのデータ型を処理する専用の計算機を追加することで行えるようになる可能性もある．応用の高度化にともなうデータモデルへのニーズは，より多くのハードウェア資源を必要としている．分散処理アーキテクチャは，高水準データモデルが理論を越えて現実のものとなるために必要なハードウェア資源を提供し得るものである．

5. おわりに

本文では，アプリケーションからのニーズと計算機技術からのシーズの側面からデータモデルとデータベース管理システムの発展を述べた．データモデルの議論は，1950年代にまでさかのぼることができるが，主流となったデータモデルは，その時代のアプリケーションからの要求やハードウェアの処理能力によって決まってきたとの見方もできよう．

近年，計算機のデータ処理能力を使うアプリケーションは，ますます広がっており，扱うべきデータも多様化し複雑になっている．より多くのハードウェア資源が必要になってきているが，これらのハードウェア資源は，分散処理によって提供されつつある．今後のデータ利用システムの開発基盤として，表現力に富み利用者にとって使いやすいデータモデルと，このデータモデルを実現する DBMS の開発が必要とされている．

参考文献

1) 植村俊亮：データベースシステムの基礎，オーム社 (1979)．
2) 上林彌彦：データベース，ソフトウェア講座 18，昭晃堂 (1986)．
3) データベース・システム——最新技術と応用の展望，ソフト・リサーチ・センター (1987)．
4) ANSI/X3/SPARC Study Group on Data Base Management Systems, Interim Report, ACM FDT, 7, 2 (1975).
5) Atkinson, M. et al.: The Object-Oriented Database Manifesto, Proc. of Deductive and Object-Oriented Databases, pp. 40-57 (1989).
6) Batory, D. S. et al.: Implementation Concepts for an Extensible Data Model and Data Language, ACM Trans. Database Syst. Vol. 13, No. 3, pp. 231-262 (1988).
7) Database Language SQL2, Draft Proposal ISO/IEC DP 9075 (Jan. 1989).
8) Date, C. J.: An Introduction to Database Systems, 5th ed., Addison-Wesley (1990).
9) Grimes, J. et al.: What is Multimedia ?, IEEE Comp. Graphics & Appli., Vol. 11, No. 1, pp. 49-52 (1991).
10) Orenstein, J. A. et al.: PROBE Spatial Data Modeling and Query Processing in an Image Database Application, IEEE Trans. Softw. Eng., Vol. 14, No. 5, pp. 611-629 (1988).
11) Shipman, D.: The Functional Data Model and the Data Language DAPLEX, ACM Trans. Database Syst. Vol. 6, No. 1, pp. 140-173 (1981).
12) Stefik, M. et al.: Object-Oriented Programming: Themes and Variations, The AI Magazine, Vol. 6, No. 4, pp. 40-62 (1986).
13) Tutorial: Distributed Database Management, IEEE Computer Society Press (1984).
14) Ullman, J. D.: Principles of Database and Knowledge-Base Systems, Vol. 1, Computer Science Press (1988).

(平成3年2月26日受付)

宇田川佳久（正会員）

1954年生．1982年東京大学大学院博士課程修了．同年三菱電機(株)入社．現在，同社情報電子研究所主事．エンジニアリング・データベースなどの研究開発に従事．工学博士．電子情報通信学会，人工知能学会各会員．