

マルチメディアデータベースシステムの アーキテクチャ

牧之内顕文、石川 博、鈴木文雄
(富士通研究所)

1. Gardian [1] によれば、データ(ベース)管理システムは、順編成ファイルを基本にした第1世代(1950年代)、直接アクセスファイルを中心とした第2世代、CODASYL 1971 勧告 [2] で代表されるアクセスモデルシステム(1960年代初期から1970年代中期まで)を経て現在は関係データモデルに基づく第4世代に進化してきた。これらのデータ(ベース)管理システムは主に型の決まったデータ("formatted data")である数値や文字データを扱う事務処理(ファイル処理)用として発展してきた。

一方、コンピュータの計算処理以外への応用が図られるにつれ、数値や文字データに加え、テキスト、グラフ、イメージ、音声などのデータも扱う応用が出現してきた。文書処理やCAD/CAM はその典型的な例である。しかし現在のところこれらの応用分野では、個別処理毎に個別のファイルを作成しておればよかった時代からマルチメディアデータを「共用」して処理する段階にきている。特にCAD分野では「設計」の共有化を計り、設計作業の合理化を進めようとの動きが出て来ている。

マルチメディアの処理を分類してみると次のような特徴が浮かび上がってくる。

(1) メディアは各々、特有の構造とそれに対する演算(操作)を有する。

そしてそれらのメディア間の差異は非常に大きい。

従来の数値・文字データでも構造や演算の差は存在していた。しかしその差は小さく、人間にとってみれば実質的な意味を持たず、むしろ計算機側の制約から来るものであった。しかしマルチメディア処理ではメディア間の差異はユーザにとって本質的である。例えばテキストの処理では編集操作が重要でありかつその処理に欠かせない本質的な操作である。しかもその「編集操作」は応用分野によって随分違ったものになる。

(2) メディアの中には非常に複雑な構造を持つものがある。

テキストなどがその一例である。利用によっては「章」、「節」などを操作単位として識別出来ねばならない。さらに、異なるメディアが互いに関連を持って一つの複雑な単位を構成する場合もある。テキストの中に図表や写真といった異なるメディアが含まれているような場合もある。それらもやはり一つの操作単位(実体)として識別されねばならない。そしてそれら実体はそれを含む実体と要素の関係にある。

データベース管理システムの発展経過が示すように、データそのものを格納・管理するだけでは不十分であり、その「メタデータ」あるいは「利用」に関する情報をも併せて管理することが重要である。これが所謂「データディレクトリ/ディクショナリ(DD/D)」の考え方である。これはマルチメディアデータベース管理システムにも適用されるべきである。

(3) メディアの利用に関する情報も又一つのメディアである。

さて、人工知能の研究では知識表現について種々の考え方が発表されて来た[4]。なかでも「フレーム」は

(A) 複雑なデータ構造を自然に表現できる、

(B) 付加手続き ("attached procedure") によりデータの動的側面を表現出来る、

ことにより非常に強力な表現能力を持つ。

このフレームを「オブジェクト」の基本構造としたオブジェクトの世界に、個別 (インスタンス)、クラス、メタクラスの3階層を導入し、「類化」や「汎化」による抽象化を実現した知識表現システムがある ([5]、[6])。これらのシステムではオブジェクトは言わば一つの抽象化された動的なデータであり、各オブジェクトは固有の行動様式を持つ。互いに関連するオブジェクトを一つのオブジェクトに抽象化することも可能である。また、オブジェクトは「メッセージ」により互いに交信し、オブジェクト群全体で一つの調和ある行動をとる。

我々は自然言語インタフェース K I D 開発への MINERVA の応用経験 [6] からこのオブジェクト指向アプローチがマルチメディア DBMS にも有効であろうとの知見を得た。それは以下の理由による。

(1) フレーム構造をオブジェクトの核構造にしたことで、オブジェクトは複雑なデータ構造およびデータ間の関連をも充分表現できる。

(2) 付加手続き機構でメソッドを実現し、そのメソッドによりメディアの動的振る舞いを記述できる。

(3) 「汎化」などがオブジェクト間の一つの関係として定義でき、それに関連する動的性質 (例えば属性の継承) をも一つのメソッドとして実現出来る。

フレームを核としたオブジェクト指向知識表現はこのように非常に強力であるがデータベースの観点からは次のような弱点がある。

(A) オブジェクトの大量化に対する配慮を欠いている。

(B) 伝統的なデータ処理 (例えば統計処理) に対する配慮がない。

「事実 (データ)」も知識であり、その処理も又一つの知識形態であることを考えるとこの事は従来の知識表現システムの大きな欠点であると考ええる。特にマルチメディア DBMS の観点からは致命的な欠陥である。

以上より、我々はオブジェクトフレーム表現を知識表現法とする「知識ベース管理」システムの上にマルチメディア DBMS を構築することが可能であり、オブジェクトフレームの関係データベース上での実現が最も妥当であるとの結論を得た。

以下では我々が考えている知識ベースシステム SENJU の構成を中心にマルチメディア DBMS のアーキテクチャについて述べる。

2. 知識ベースシステム

知識ベースシステムは三階層構成をとる (図1)。この三階層構成は ANSI / SPARC のデータベースアーキテクチャ [7] に似ている。ANSI / SPARC アーキテクチャの概念モデルが図1のオブジェクトレイヤで仮定されるオブジェクトフレームモデルである。オブジェクトフレームは拡張 RDB に写像される。この拡張 RDB モデルが ANSI / SPARC アーキテクチャの内部モデルに対応する。応用レイヤは知識ベースを使用する応用の個別知識に関する。これは ANSI / SPARC の外部モデルに当たる。

応用レイヤの応用知識はデータベースのビューに近い。しかしビュー以上であ

る。むしろDD/Dに近い。DD/Dにはデータ処理プログラムやそれが使用するデータに関する情報が蓄えられる。この類似から言えば知識ベースシステムはいわばDD/DとDBMSとの両者が管理すると仮定されている情報を統一的に管理するシステムであると言えよう。

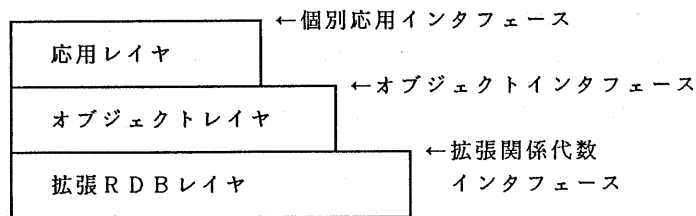


図1. 知識ベースシステムの構造

3. オブジェクトレイヤ

オブジェクトレイヤに於ける基本知識表現はフレームである。フレーム〔8〕は図2に示す形をしている。

フレーム

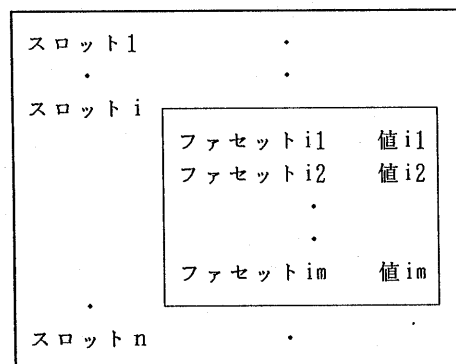


図2. フレームの形

フレームは1個以上の「スロット」からなる。スロットもまた複数の「ファセット」を持ちうる。値もまた単値ばかりでなく多値（単値の集合）もある。このように一つの項目の下位項目が繰り返すことが大きな特徴である。しかも値として他のフレームの名前を持つことによりフレームの中にフレームを格納する構造も可能である。この御蔭で非常に複雑な構造をフレームで表現することができる。

オブジェクトレイヤに於ける管理・操作対象は「オブジェクト（事物）」である。オブジェクトは世界の物、事、行為、事象、事件等の実在の抽象化された実体を表現する。オブジェクトはこの実体を持つ属性の三つの側面を表さなければならない。

（1）実体の性質（“property”）

(2) 実体と他の実体との関係 (“relationship”)

(3) 実体の振る舞い (“behavior”)

一つの実体を取り上げて、それが持つ性質、関係、そして振る舞いを特定するのは応用分野毎に行わなければならない。しかも、一つの応用分野を固定したとしても、ある実体の何が性質で何が関係であるか、あるいは何が振る舞いであるかを決めることが困難な場合があるかもしれない。特に性質と関係の区別は恣意的でありうる。しかし、ともかく実体は上記三種類の側面を持っており、その象徴であるオブジェクトはそれらを過不足なく記述出来ねばならない。

このようなオブジェクトをフレームで実現するのは簡単である。一つのフレームを一つのオブジェクトに対応させる。スロットが性質、関係、又は振る舞いを表現する。性質を表すスロットは（例えば人の年齢）はその性質を代表する実際の値（例えば25才）を値として与えられる。関係の場合は、値として他の実体（即ちフレーム）の名前を持つ。振る舞いは値にプログラム（あるいは関数）が格納される。

オブジェクトレイヤのオブジェクトは三種類に分類される。一つは「個別 (“instance”) オブジェクト」であり、これは個別の実体（例えば学生のA君）を表す。類似の個別オブジェクトが属するクラスを「クラスオブジェクト」と呼ぶ。クラスオブジェクトの持つ属性はそのクラスに属する個別オブジェクトの共通の属性やそのクラス自身に関する属性を持つ。クラスオブジェクト自身もまたクラスに属する。そのクラスを「メタクラス」と言い、やはり一つのオブジェクトである。このようなオブジェクトの三階層構成は他のシステムにも見られる（〔5〕、〔9〕）。

オブジェクトはメッセージを受け取るとそれを解釈し、その指示に従って行動する。場合によっては受け取ったメッセージを転送したり、新しいメッセージを作り他オブジェクトにそれを発信する。このように関係するオブジェクト間でメッセージをやりとりして、その交信に参加したオブジェクトが全体として統一のとれた動きをする。

受け取ったメッセージによって起動される活動を「メソッド」と呼ぶ。これは上述したようにフレームのスロットの値としてプログラムを与えることにより実現される。その時のスロット名がメソッドの名前になる。

オブジェクト間の関係として「所属関係」がある。これは一つの個別（あるいはクラス）オブジェクトがクラス（あるいはメタクラス）オブジェクトに「属する」ことを示す。又、クラス（メタクラス）オブジェクト間には「包含関係」（あるいは「スーパー・サブクラス関係」とも言う）がある。これは集合の包含関係と同じ意味である。これら二つの関係は知識ベースシステムSENJUの基本システム関係であり、一般のいわゆるオブジェクト指向の知識表現システムにも見られる。これら関係は知識表現の立場からは類化と汎化による抽象化として重要な役割を担う。

マルチメディアDBを扱う為にSENJUでは他に二つの主要な抽象化概念を導入する。一つは「進化による抽象化」であり、他の一つは「集約による抽象化」である。前者はオブジェクトが時間と共に変化する様態を記述するものであり、後者はオブジェクトとオブジェクトとの要素関係を表現するものである。

従来のDBでは、データのスキーマは時間的に変化しないものと仮定されていた。しかし、設計DBの分野ではスキーマそのものの変化を記述することが必要

である。知識ベースシステム S E N J U ではすべての階層の知識をオブジェクトフレームと言う同一の形式で表現できるため、オブジェクトの時間的変化の様式を一つ定めればそれによってすべての階層での知識の時間的変化を表現できる。

4. 拡張 R D B レイヤ

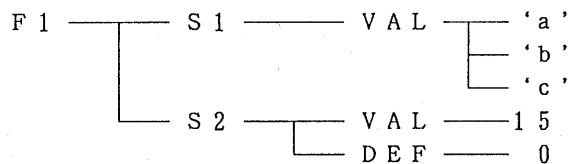
知識ベースシステムの知識格納構造として関係データモデルを採用した。関係データモデルは

- (1) インタフェースが簡単であること、
- (2) 既存の関係データベースを活用できること

の二つがその理由である。

オブジェクトレイヤではフレームの名前による直接アクセスが典型的なフレームアクセス法である。また、特定のフレームを検索した後、そのフレームのロットやファセット自身を探索することもありうる。即ち、データベース操作の常識で言えば検索対象になりえない属性名も検索対象となりうる。このことを考慮すれば、図3に示した格納法が標準的であろう。我々はこの格納法を(フレームの観点から)「標準法」と名付ける。

(フレーム) (ロット) (ファセット) (値)



フレーム名	ロット名	ファセット	値
F 1	S 1	VAL	'a'
			'b'
			'c'
	S 2	VAL	1 5
		DEF	0

図3. フレームの標準格納法の例

上述したように、フレームには繰り返し項目や多値の概念があるので、関係モデルを入れ子構造をゆるすように拡張する〔10〕。この拡張したモデルは N F²

(Non First Normal Form) 関係モデル〔11〕と呼ばれる。図3はこのモデルに従って書かれた。

この標準格納法の問題点は同一の属性の下に異なるタイプのデータが出現することである。これは、従来の関係モデルで暗に仮定されていた同一属性－同一ドメイン－同一データタイプの関係と折り合わない。これはフレームがデータタイプレス言語であるLisp文化圏で育った概念であるからであろう。一方、関係データモデルがCOBOL等のデータタイプを強く意識するプログラミング言語を利用する環境から発展してきたことが上記の仮定を導く大きな要因であるものと思われる。

この問題を解決するためには「タグ付値」を値のタイプとして導入することが考えられる。即ち、元々ファセットは値の種類を示すタグであるから、ファセットと値を一体化してタグ付値とするという考えである。一方、(テーブルの)属性値のタイプが他の属性値によって決定されるような「タイプ依存性」を導入することも考えられる。

標準格納法の他の問題点は値の意味がタプル毎に異なることである。従って所望する値を求めるのに必ずスロットに関する条件を書かねばならない。しかも場合によっては従来なら不必要であった結合演算を施さなければならない。

このような欠点は、標準格納法では実体の一つの属性をテーブルの一つのタプルに写像することに由来している。これは関係データモデルの考え方、即ち実体を(原則として)一個のタプルで表現することに矛盾する。(関係データモデルから言えば標準格納法は標準ではない。)

標準格納法に対して、「転置格納法」を導入する。これはフレームのスロットをテーブルの属性に対応させる(図4)。この格納法は関係モデルの思想と一致するので標準格納法に見られた不都合は避けられる。(但し、同一属性の下に異なるタイプの値が現れることについては解決できない。)

フレーム名	S 1	S 2
F 1	(VAL) 'a'	(VAL) 15
	(VAL) 'b'	
	(VAL) 'c'	(DEF) 0

図4. 転置格納法によるフレームF1(図3参照)の表現

標準格納法が、その欠点にもかかわらず棄てがたいのはクラスオブジェクトの格納法として適しているからである。クラスオブジェクトはテーブルに、個別オブジェクトはそれが属するクラスに対応するテーブルのタプルに各々対応づけられる。特定の応用分野をとってみても、テーブルの数が数個ということはない。この少なからぬテーブルに関する情報(即ち、クラスオブジェクトそのもの)を格納するのにそれらテーブルの個数だけのテーブルを創成することは無駄である

う（しかもそれらテーブル中に格納される数は精々数個である）。

オブジェクトにはプログラム（関数）もデータの一部として格納される。このため、「長いフィールド」を関係モデルに導入する必要がある。この長いフィールドはマルチメディアDBへの応用を考える時、より必要性が高まる。テキストやイメージ等をデータとして蓄積・管理しなければならないからである。

5. 応用レイヤ

このレイヤは他の二つのレイヤと趣を異にして、特定の知識構造を実現するものではない。むしろオブジェクトフレームを使って特定分野の「知識」を蓄積・管理する。

例えば、マルチメディアDBMSはこのレイヤに実装される。あるいは、統計DBMSも統計に関する知識をこのレイヤに与えることにより実現される。もし両分野の知識が実装された場合、両者に関する知識を使って、マルチメディアデータ上での統計処理を行うことも可能である。

このようなことが可能になるのは、オブジェクトフレームの表現能力が高いことによる。しかも、その表現能力を活かしてすべての階層のオブジェクトを同一の形式で表現する「自己記述方式」をとることが可能なことがSENJUの柔軟さを保証する。この自己記述方式は拡張関係DBレイヤにも適用する。例えば、アクセス法としてのインデックスを（効率的に）実現する手段としてNF²テーブルが有効である。

6. 終わりに

マルチメディアDBシステムを構築するのにより広い枠組みを用意した。それは知識ベースシステムである。そこではAI分野で知識表現法として発展してきたフレームとプログラミングパラダイムとして提唱されたオブジェクト概念、そしてDB分野のエース関係データモデルが統合される。これにより、知識とデータの真の融合が達成される。

我々はこの知識ベースシステムのテストベッドとしてマルチメディアDBMSを考えている。それは、マルチメディアDBの扱うデータは構造が複雑であるばかりでなく、その利用知識がデータベース化されねばならないと考えるからである。

本論文では知識ベースシステムSENJUのアーキテクチャの概略を述べた。現在、各レイヤのインタフェースを設計中である。さらに拡張RDBレイヤを実装する高速RDBエンジンの基本構想を固めつつある。これらについては機会を見て順次発表して行きたい。本論文でほとんど触れなかったマルチメディアDBそれ自身（例えば、狭い意味でのモデル）についても検討してきた。オブジェクトフレームモデルはそのための表現能力を十分持っていることが確かめられている。これについても別の機会に紹介したい。

最後に、拡張RDBレイヤには分散機能〔13〕が最初から実装される。これはマルチメディアDBMSの使用環境を考慮すれば当然の選択であろう。

謝辞

本論文の基本的考え方は筆者達の責任に帰する。マルチメディアデータベース研究開発グループの他のメンバはここに記した考え方の展開に寄与した。感謝の意を表したい。

参考文献

- [1] Gardian, G.: "Towards the fifth generation of data management system" in "New application of data bases" Academic Press, London 1984
- [2] CODASYL DATA BASE TASK GROUP: "Report of the DBTG" ACM New York Ed., April 1970
- [3] Codd, E. F.: "A relational model of data for large shared data banks", Comm. of the ACM, Vol. 13 No. 6, June 1970
- [4] Barr, A. and Feigenbaum, E. A. 編、田中幸吉、淵一博 監訳: 「人工知能ハンドブック」第I巻, 共立出版社
- [5] Steik, M. and Bobrow, D. G.: "The LOOPS Manual: A data oriented and object oriented programming system for Interlisp" Xerox Knowledge-Based VLSI Design Group Memo KB-VLSI-81-1, March 1981
- [6] 泉田義男 他: 「対象世界のモデルを利用したデータベース検索システム」情報処理学会データベース・システム研究会43—2、1984年9月
- [7] ANSI/X3/SPARC Study Group on Data Base Management System: "Interim Report FDT 7 No. 2" 1975
- [8] Winston, P. H.: "Artificial Intelligence" Second Edition, Addison-Wesley Publishing Company, Massachusetts
- [9] 亀田雅之 他: 「Prologによるオブジェクト指向フレームシステムの試作」情報処理学会第27回全国大会(昭和58年後期)昭和59年3月
- [10] Makinouchi, A.: "A consideration on normal form of not-necessarily-normalized relation in the relational data model" Proc. of the Third International Conference on VLDB, October 1977
- [11] Deppisch, U. et al.: "The storage component of a data base kernel System" Technical Report DVS1-1985-71, Technische Hochschule Darmstadt Fachbereich Informatik, 1985
- [12] Haskin, R. L. and Lorie, R. A.: "On extending the function of a relational database system" Proc. of 1982 ACM SIGMOD International Conference, 1982
- [13] 手塚正義 他: 「関係型均質分散データベースシステムRDB/DV」情報処理学会データベース・システム研究会47—3、1985年5月