

解説



地図・図面情報処理における マルチメディアデータベース†

嶋 田 茂††

1. はじめに

官公庁・地方行政体における行政計画の策定や、電力・水道・ガスなどの公共事業体における設備計画の立案などの業務においては、地図や図面情報を計算機により高速に処理し、目的とする地域における各種の検索がインタラクティブに自由に行えることが求められている。このような要求を実現しようとする、一般に、データ容量が大きくかつデータ構造が複雑となるため、従来十分満足のいく応答特性を備えたシステムの実現が難しく、その実用化は遅れていた。しかし最近、エンジニアリングワークステーションなど高性能で安価なグラフィックシステムの出現と、光ディスク・CD-ROM などメディア媒体技術の進展により、商用システムが相次いで発表されるようになり、本格的な地図・図面情報処理システムの普及時期が到来しつつある¹⁾。このような状況において、今後さらに必要となる地図・図面情報処理システムの機能のあり方をさぐるため、われわれの研究室では、知識型地図情報処理システムのプロトタイプ (GENTLE) を試作し、その評価を行ってきた²⁾。その結果、地図・図面情報処理システムの今後の課題としては、次に示す三つがあると考えている。

(i) マルチメディア化

地図・図面情報処理システムの商用化の変遷をたどると、各システムには、その時期までに実現されたメディア媒体技術が確実に反映されていることが分かる。たとえば初期のシステムでは、磁気ドラムやディスクを用いてメッシュデータもしくはベクトルデータといった単一種のメディア情報を扱う処理系であったのに対して、最近では光ディスク・CD-ROM など多様なメディア媒体の出現によりラスタ画像とベクトル

データとを融合させるといったマルチメディア情報化の傾向にある。この場合単にラスタ画像の上にベクトル図形を重畳させるといった単純な処理系ではなく、たとえば、ベクトルで表現された等高線の群に対し、この地域に対応した航空写真画像を部分的に切り出し、さらに3次的に再合成するといった密接な関係構造化を行う意味でのマルチメディア情報化が必要となってきている。

(ii) 高度分散処理化

最近のワークステーションの急速な進歩により、分散処理化が検討されるようになってきた。この分散処理はワークステーション間の水平分散処理と、メインフレームとワークステーション間の垂直分散処理との二つの面がある。そして、単に処理の分散化ができるという段階に留まらず、その処理効率が格段に向上することが重要で、そのための技術課題が山積している。

(iii) 知的インタフェース

システムとのマンマシンインタフェースを向上させる一つの流れとして、最近マルチウィンドウを備えたシステムが多くなってきた。しかし単にウィンドウが多数表示できるだけでは、かえって操作を複雑にすることになりかねない。むしろシステムユーザが望む処理内容を予測したシステム環境を整えること、たとえばデータベーススキーマにそった最適な検索手順を推定する技術などが重要となってくる。これによりシステムへの要求が多少あいまいなものでも推論検索が可能となり、自然言語検索や会話音声による検索など、知的インタフェースへの展開が可能となる。

この三つの課題は密接な関係にあるため、単独の課題として対応することは難しい。本稿ではこのうち (i) のマルチメディア化の観点から議論を進める。特にデータベースの見地から考えると、多様な媒体内のデータを、データベース内部へ転送した後のマルチメディア情報の扱いが重要と考えるため、以降本稿で

† Multi-media Database for Maps and Drawings by Shigeru SIMADA (Central Research Laboratory, Hitach Ltd.).

†† (株)日立製作所中央研究所

用いるマルチメディアとは、マルチメディア情報を指すものとする。

2. 地図・図面情報処理の状況

2.1 システムの開発動向

現在までに実用化が進められてきた地図・図面情報処理システムの開発状況を概観する。表-1 は代表的なシステムの開発状況を年代順に並べ、各システムの特徴を次のような項目について分類したものである。

メッシュ型とポリゴン型・・・地図情報の記憶形態に関する分類である。メッシュ型では、地域を一定範囲のメッシュ（たとえば1 km 四方など）に分割し、その範囲を属性付きの代表点で記憶するのに対して、

ポリゴン型では属性との対応を付けるべき範囲の境界を、ポリゴンとして詳細に対応させる。

出版型と検索型・・・出版型のシステムでは、地図や図面データそのものを作成することを目的としており、出版物または CRT への表示が鮮明であることが要求される。一方検索形のシステムでは、各種の数値・文字データを図面上で関係付け、各種の検索が可能となるような構造化が要求される。

ベクトル型と画像型・・・ベクトル型では、図面中の図形を従来の CAD システムで用いられていたように、ベクトル座標系列として記憶するのに対して、画像型では圧縮コード化されたラスタ画像として記憶する。

表-1 代表的な地図・図面情報処理システム

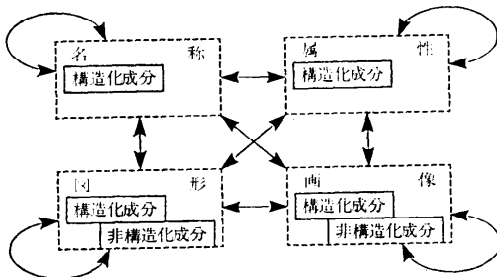
完成時期	システム名称	主な開発機関	システム構成	分類		
1970初	CGIS	カナダ政府	MF	MS	PB	VC
	DIME	米統計局	MF	MS	PB	
	UIS	建設省	MF	MS	PB	
1981	GDBS	IBM	MF	PO	RE	VC
	TOGIS	東大生研	MC		RE	IM
1982	ISLAND	国土庁・富士通	MF	MS	RE	VC
	ARC/INFO	ESRI 社(米)	MC	PO	RE	VC
	INFOMAP	シナコム社(米)	MC	PO	PB	VC
	Total Utility Mapping System	東京ガス・国際航業	MC	PO	RE	VC
1983	ODYSSEY	ハーバード大学	MF, MC	PO	RE	VC
	WING	日本電気	MF, MC	PO	RE	VC
	UDAMS	三井造船・アジア航測	MC	PO	PB	VC
1984	HMAP-B	ゼンリン・日立	MF+ADG	PO	PB	VC
	COMPUS	玉野総合コンサルタント	MC	PO	PB	VC
	TOUMAS	国際航業	MC	PO	PB	VC
1985	ARISTOWN	富士通	MF	PO	RE	VC
	FAMOS	富士電機	MC+WS+ADG	PO	PB	VC
	MIC	社会調査研究所	WS+OD		RE	IM
	OGIMAPPING	応用技術	WS	PO	RE	VC
1986	MARVIS	パソコ	PC	PO	RE	VC
	MAGELLAN	STELLA SYSTEM(日)	PC	PO	RE	VC
	ZMAP	ゼンリン	PC+OD	PO	RE	VC
	HMAP-D	日立	MF+WS	PO	RE	VC
	HMAP-F	日立	WS+OD		RE	IM
	AQUA-MAP	日立	WS	PO	RE	VC
	図面管理システム	三菱電機	WS+ADG	PO	PB	VC
	総合防災情報処理システム	名古屋市・富士通	MF	PO	RE	VC
	地中情報処理管理システム	大阪ガス・日本 IBM	MF	PO	RE	VC
1987	地図利用情報管理システム	東芝	MC	PO	RE	VC

略号 MF: Main Frame, MC: Mini-Computer, WS: Work Station, ADG: Automatic Digitizer, OD: Optical Disc, PO: ポリゴン型, MS: メッシュ型, PB: 出版型, RE: 検索型, VC: ベクトル型, IM: イメージ型

以上の分類に対して地図・図面情報処理システムの変遷と最近の動向を探ると次のようになる。まず従来の地図システムは、国勢調査などの統計データを、2次元的に可視化するためのいわゆるマッピングシステムが主流であったのに対して^{9),9),10)}, 1980年代からは、より詳細で、かつ自由度の大きな記述のできるポリゴン型が主流となっている^{4)-6),9),10)}。これはコンピュータシステム技術の進展に負う点が多く、タブレットのような安価な図形入力手段が実用化されたこと、データベース技術の進展によって、ポリゴンでも実用レベルの対話応答速度が達成できるようになってきたことなどの理由が考えられる。また最近の動向として、出版型システムで作成された市販の電子化地図を用い、その上に施設や計画図面を重畳させて利用する検索型のシステムが多くなってきている^{1),8)}。また急速な画像情報処理技術の向上により、光ディスクを利用した画像型のシステム^{7),10)}も多くなる傾向にある。さらにベクトル型と画像型とを融合させたマルチメディア指向のシステムへの要求が高まりつつあり、しかもすでに述べたように、画像とベクトルとを重畳表示させるというような単純な要求だけでなく、各部分要素に意味的な関係構造を与えるといった高度な要求へと発展しつつある。そのキー技術となるのがマルチメディアデータベース技術であると考えられる。

2.2 マルチメディア化に対する要求

地図・図面情報処理システムにおけるデータベースの備えるべき機能としては、単にマルチメディアが管理できるというだけでなく、行政計画や設備計画における編集や検索が、対話的にしかも高速に実行可能でなければならない。一般的に処理速度は、一度に処理対象とするデータの容量に依存することが多いので、



名称：地名・機関名などの固有名詞
 属性：人口密度・配管口径などの数値
 図形：地形図・設計図面などの図形及びテキスト成分
 画像：画像化された地形図や航空写真及び情景写真など
 図-1 地図・図面情報システムにおけるマルチメディアモデル

データベース構成上の工夫が要求される。

地図・図面データベースの構成は、図-1に示すように図形・画像・名称・属性の4つの要素が互に関係構造をもったモデルとして考えることができる。その各要素の具体的な内容として、名称は地名・機関名など固有の名称を示す文字データ、属性は人口密度・配管口径などを示す数値データ、図形は形状を記述するための座標データ、画像は航空写真やスキャニングされた地図などの画像データをそれぞれ示す。特に名称データを独立の要素として扱うのは地図の場合、地名や建物名称からの高速検索が重要視されるからである。このようなデータ構造に対し検索や編集などの処理は、図-1の矢印で示すようなメディア間に及ぶすべての関係について処理可能であるのが望ましい。

このようなメディア別の分類の観点とは別に、図形や画像データに関しては、構造化成分と非構造化成分の分類が考えられる。ここで構造化成分とは、ほかのメディア情報と関係付けられた成分のことである。そしてこの構造化成分の大きさが全体のマルチメディアデータベースの容量の大きさを左右することが多い。そこでシステムの要求機能を満足させるのに必要十分な範囲だけに構造化を限定する必要がある²⁾。たとえば前節の分類において、出版型のシステムでは個々の図形成分に幾何的な構造情報（たとえば点・線・面のような）を与える必要はなく、建築物・道路などといったマクロな図形群の単位に表示属性を関係付ける程度でよい。また検索型のシステムにおいても、検索対象だけを構造化すればよく、地図としての雰囲気を示すための要素（たとえば塀・垣根・崖記号など）の一つ一つを構造化する必要はない。このような工夫によるマルチメディアデータベース容量の削減の効果は、個々の検索時間の短縮に直接寄与することになる。

3. システム構成とマルチメディアデータベース

3.1 地図・図面情報処理システムの構成

マルチメディアデータベースの特性は、その構造の良否以上に、システム構成の影響を強く受けるため、本章ではシステム構成の立場からみたデータベース構造の要求について考察する。

まず地図・図面情報処理システムの構成として、最近特に要求される内容を分析すると次のようになる。

(1) 水平分散処理の要求

地図・図面データベースは一般に大規模となり、た

たとえば人口20万人程度の小都市における上下水道の管理業務だけでも全体で数ギガバイトオーダーとなる。このような大規模なデータベースから、必要な地域のデータを抽出する処理は、一般に負荷が重く応答性が劣化することが多い。そこで、あらかじめ担当地域別にデータベースを分割して、ワークステーション上のローカルデータベースに担当地域のみを登録し、それ以外は必要に応じてほかのワークステーションの情報を利用するような水平分散の要求が多くなりつつある。

(2) 垂直分散処理の要求

地方自治体や公共事業体などにおいては、住民登録・工事履歴などすでに大規模なデータベースとして中央集約的に管理されている場合が多い。この既存のデータベースを検索し、その内容をワークステーション上の地図の上にマッピングし、各種計画の立案支援を行うような垂直分散の要求もまた高まっている。

このような要求以外にも、たとえば同一性能を実現するのに要するハードウェアコストの比が低いことと

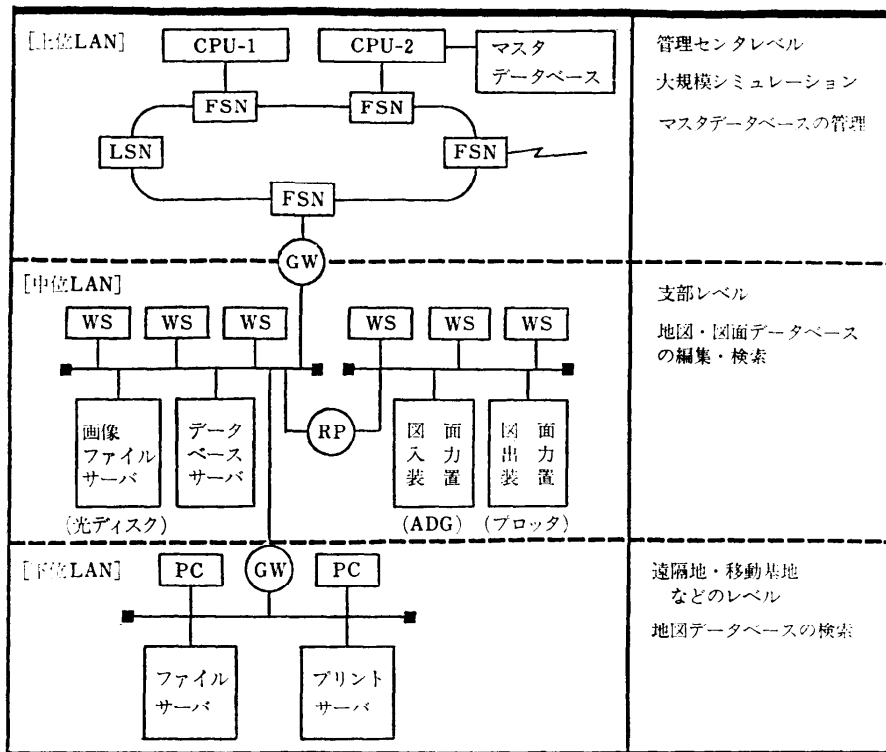
か、データベース変更にもなう一貫性管理を局所化するなどの要求がある。水平・垂直分散処理を実現するシステムの構成案としてはいろいろ考えられ、実際にはデータ規模、トランザクション頻度、最小アクセス時間などの条件を分析して決定される。ここでは設計上の条件を厳密に考えないことにし、むしろマルチメディアデータベースを参照するすべての様式が含まれるようなモデルを想定する。その一案として、図-2に示す3階層のLANによる水平・垂直分散処理システムが考えられる。このモデルの各部の位置付けと処理内容は次のようになる。

【上位 LAN】

本部のセンターレベルの位置付けにあり、大型の地図や図面の初期入力や、各ワークステーションから転送された編集後のデータの一貫性保持のための整理、及び大規模なデータによるシミュレーションなどを行う。

【中位 LAN】

支部レベルの位置付けにあり、各ワークステーション



FSN; Field Service Node GW; Gate Way WS; Work Station
 LSN; Loop Service Node RP; RePeater PC; Personal Computer

図-2 階層化分散処理システムの構成案

ンにはそれぞれ中規模のハードディスクを備え、通常の検索や編集に不足しない程度の範囲を覆うデータを記憶する。そして各種の検索・編集などをワークステーション単体で行う。

【下位 LAN】

遠隔地や移動基地などからの検索要求を処理するレベルにある。パーソナルコンピュータや安価な端末によるネットワークを構成し、上・中位 LAN から必要なデータを転送し、検索主体の処理を実行する。

以上のモデルにおけるシステムの中核は、中位 LAN に接続されるワークステーション群であり、マルチメディアデータベースは、単純なワークステーションレベルのローカルデータベースの特性のほかに、上位または下位のデータベースとの連携を考慮した分散的な特性を備える必要がある。このような分散データベースの一般論に関してはほかの文献にゆずり、本稿では後に示すような知識ベースの活用による分散型マルチメディアデータベースの実現方法を紹介する。

3.2 地図・図面自動入力インパクト

ここで、前節までに触れた地図・図面情報の計算機への入力方式について補足しておく。すでに述べたように、地図・図面データの計算機への入力で問題となるのは図形の構造化成分であり、今までのベクトル型システムにおいては、その実用化を左右するのは、この地図・図面データの入力効率の良さにあるといっても過言ではない。

このような要求を反映して、最近各種の自動図面入力システムの実用化の報告がなされるようになった^{11)~15)}。しかしいずれも、特定の地図や図面を対象とし、かつ図面の書き方に制約を設けたり、認識の条件を規定したりするものが大部分であり、既存の地図や図面を実用レベルの認識率で入力できるものは皆無

に近い。またこのような自動入力にはミニコンによる専用システムや、専用ハードウェアによる場合が多く、汎用計算機によるものは少なかった。しかし最近、ランレグス圧縮コード画像を直接用いた高度なバイプライン構成による高速画像処理方式が開発され、従来 CPU 時間にして数時間のオーダを必要としていた処理が、数分のオーダで実行可能となっている¹⁶⁾。これは通常のデータ処理のレベルに達したことに相当し、ほかに負荷分散を求めることなく、汎用大型計算機でも十分実用的な地図・図面の自動入力が可能な段階になったことを示す。したがって特に既設の大型計算機が利用できるような環境のもとでは有利となる。

前節のシステム構成の立場から考えると、上位 LAN レベルにおける大型計算機により、このような地図・図面の初期入力を行うのが一つの方法である。

4. マルチメディアデータベースの構成方法

今まで述べてきた地図・図面情報処理のためのマルチメディアデータベースを実現するのに必要な技術課題は、すでに研究が進んでいるエンジニアリングデータベース (EDB と略称する) の中心的課題と同一の内容を含んでいる¹⁶⁾。現在までの EDB の実現方法に関する各種の研究のアプローチを分類すると、表-2 のようになる。これらと比較すると、まず第1のアプローチでは、一ほかに他のシステムとのデータの互換性は考えないことが多く、新たな機能の要求が起こるたびに専用データベースの構造をそれに合わせて変更することが多くなる。したがって開発工数もまた大きくなると考えられる。一方第2のアプローチでは、データ形式のフォーマット変更を加えることにより、簡単にマルチメディア化を図ることができる。しかし、データベースの設計が適当でない場合には、システムの応答性がかなり劣化する危険性がある。さて第

表-2 エンジニアリングデータベース (EDB) 研究の主なアプローチ

項番	アプローチの内容	実現方式の特徴	拡張性	応答性	難易度
1	問題向き専用の DBMS の開発	従来の CADシステムで採られているアプローチで、要求される機能に対し最適なデータ構造を定義し、それ専用の DBMS を備える。	△	○	△
2	汎用 DBMS を利用した実現	汎用 DBMS で管理可能なデータ形式にすべてのデータをフォーマット変換して管理するか、または図形・画像などの特殊な部分だけを専用の DBMS で管理する。	○	△	○
3	新たなマルチメディア DBMS の開発	図形・画像を含むマルチメディアを高速に管理することの可能な汎用 DBMS を新たに開発する。	○	○	?

DBMS: Database Management System

3のアプローチは、全く新しい汎用のマルチメディアデータベース構造を目指すものではあるが、現在の技術レベルでは開発におけるリスクが大き過ぎ、実用レベルに至るまでの開発期間が相当長くなることが予想される。そこで以下本稿では、これらのうち第2のアプローチを中心にして解説を進める。

4.1 汎用データベース管理システム適用上の問題点

最近になって、メインフレーム用に加えパーソナルコンピュータやワークステーションなど中・小型システム向け汎用の関係形式データベース管理システム(以降 RDBMS と略記)が多く発表されるようになってきた¹⁷⁾。しかし地図・図面データベース管理用として直接適用することは一般に難しい。その理由を考えると、次のような項目がある。

(1) 従来の RDBMS では、扱うことのできるレコード長の範囲も狭く、画像や図形など長大なデータ形式をとるメディアへの対処が十分ではない。

(2) あらかじめ定義されたデータベース構造に対し、その枠を超えた変更が頻繁に発生するので、データベース操作や管理命令が複雑になる。

(3) データのもつ意味に関する表現方法が不十分なため、スキーマ構造の管理の大部分を応用プログラムのレベルで対処しなければならない。

これらの問題点を解決するための方法として、すべてのデータを RDBMS で処理可能なデータ形式に変換し、これらを一元的に管理する「一元管理的アプローチ」と、図形・画像など RDBMS では管理が難しいメディア情報を専用の管理システムで行い、外側の概念レベルでの管理を RDBMS で行う「多元管理的アプローチ」とが考えられる。以下この両者のアプローチの代表例と特徴について比較する。

一元管理的アプローチ・・・このアプローチの代表

例としては、すでに日本 IBM の GSDPS(1981)⁴⁾ や、日本電気の WING (1982)⁶⁾ などが発表されており、名称・属性管理のほか、図形要素に至るまで完全に一元的な関係構造化を行っている。このアプローチによれば、単一の管理手続きによりすべてのデータのメンテナンスが可能となるといった省力効果が存在するほかに、柔軟で無駄の少ないデータ定義が可能といった関係形式そのものも特性が期待できる。ところが RDBMS では汎用・多重検索を目標に作成されるため、管理プログラムの計算機上での負荷は大きく、検索の応答速度も遅いなどの問題が残っている。しかも一般に、このようなデータ定義の柔軟性と検索の高速化を両立させる関係構造を設計することは非常に難しい。

多元管理的アプローチ・・・最近進展の著しい CAD システムメーカーや地図情報処理システムメーカー(たとえばインターグラフ社や ESRI 社など)が採用している方式である⁹⁾。これは RDBMS では直接扱いにくい図形や画像データを専用のテーブルで管理し、その上に RDBMS をかぶせることにより、ほかのメディア情報(名称や属性など)との関係をつける方式である。この場合、専用データベースの構造を工夫することにより、全体として一元管理的アプローチよりもコンパクトでしかも高速検索の可能なデータベース管理システムとなる可能性を秘めている。しかしデータ操作に関する自由度が少なくなることもありうる。たとえば図形に幾何的な操作を加えようとした場合、管理システムを RDBMS から専用の管理システムに切り換える必要があるほか、図形管理の手続きを応用プログラム側で記述しておかなければならない。また一元管理方式では容易に表現可能であった、点一線一面の幾何構造化を図るのが難しくなる。

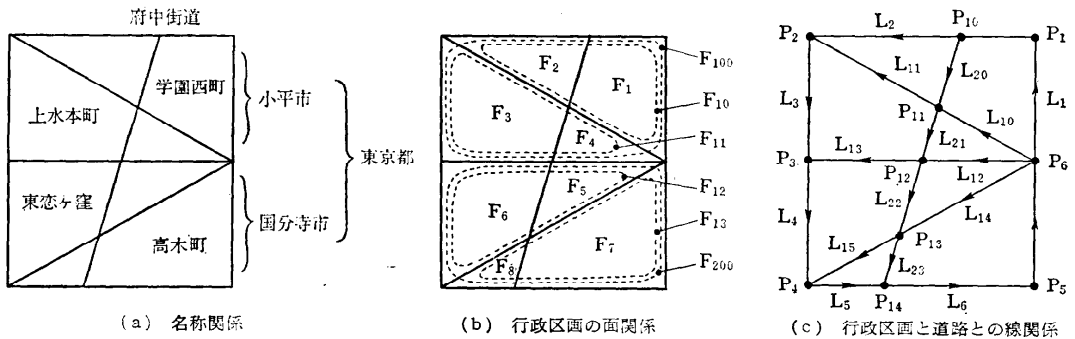
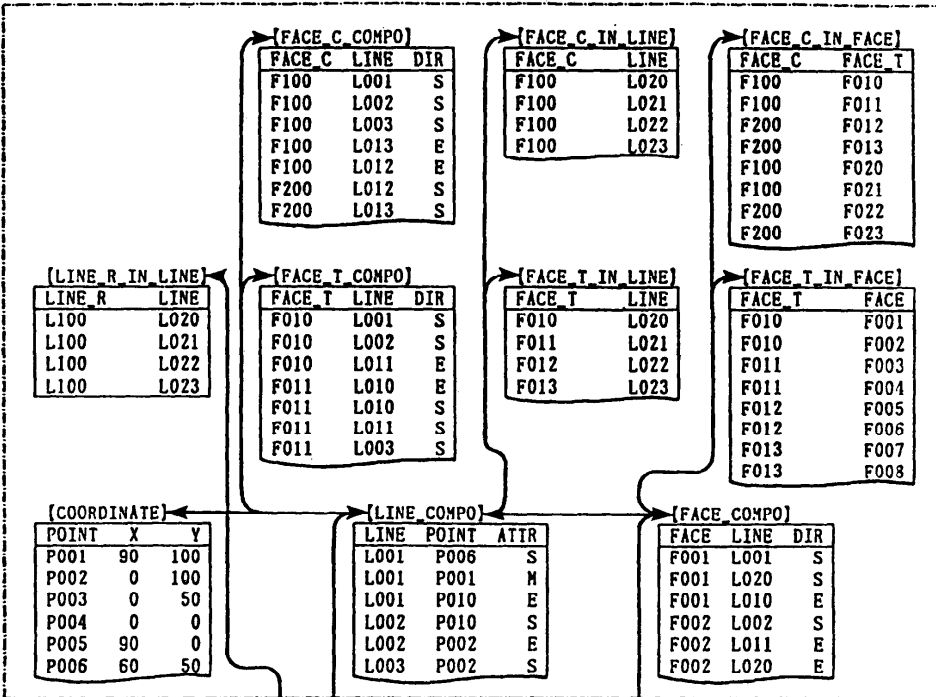
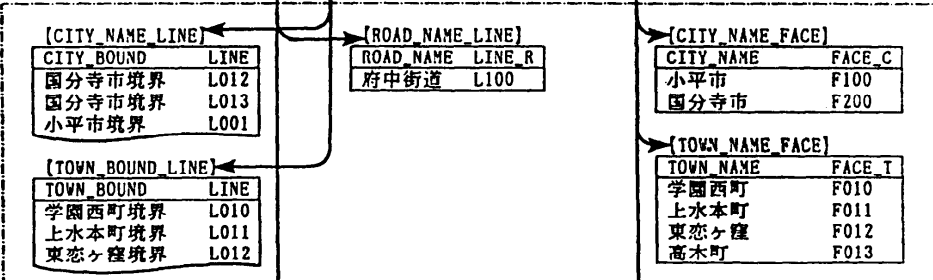


図-3 簡略化した地図のモデル

【図形-図形 関係テーブル】



【図形-名称 関係テーブル】



【名称-名称 関係テーブル】

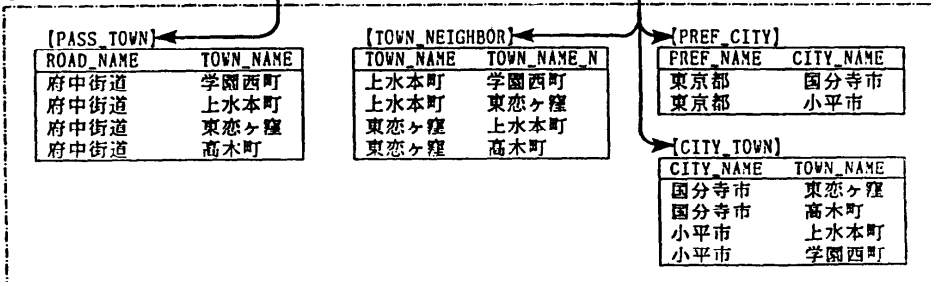
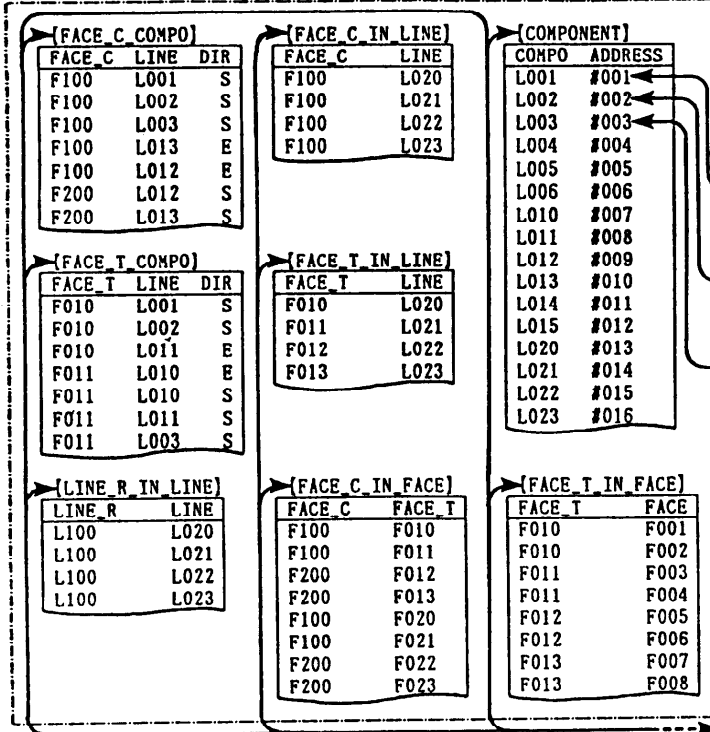


図-4 一元管理的アプローチによる記述例

【図形-図形 関係テーブル】



【図形専用テーブル】

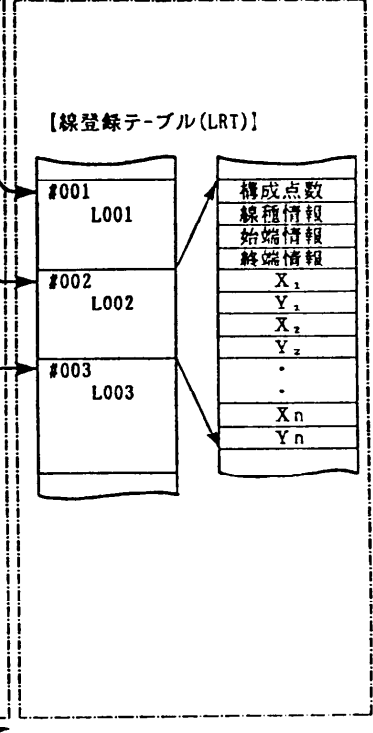


図-5 多元管理的アプローチによる記述例

4.2 具体例による各アプローチの比較

両者の特性を具体的に把握するため、図-3 に示すような簡略化した地図のモデルについて考える。このモデルでは、(a)に示すように、東京都が小平市と国分寺市だけからなり、その中央に府中街道が通るような小さな世界を想定している。一方図形の概念としては、行政区界を示す図形を(b)に示すように面一面の関係で、行政区界を示す記号を(c)で示すように点一線の関係でそれぞれ記述している。この地図のモデルを一元管理的アプローチにより記述すると図-4 のようになり、多元管理的アプローチにより記述すると図-5 のようになる。ただし、図-5 において、図-4 と重複する部分は省略した。

一元管理的アプローチによる記述例と問題点

まず最初に図-4 における小平市の図形の定義に着目すると、小平市を示す面 (F100) は、学園西町を示す面 (F010) と上水本町を示す面 (F011) とを包含する関係にあるほか、行政区界境界線を示す線群 (L020, L021, L022, L023) で構成される関係にもある。これらの面や線群を具体化して描画するための座標値を得るための関係処理を行うと、同一の記憶

位置に書かれた座標として求めることが可能で、多重定義のない効率的な定義が可能となっている。しかしたとえば「上水本町を示せ」といった簡単な検索要求を満足させるにも、長大な検索手順を要し、検索時間が膨大になる問題がある。その原因は、検索の対象とする図形そのものではなく、要素間の関係を格納した多数のテーブル群 (FACE_COMPO, LINE_COMPO, COORDINATE) をたどる必要があり、名称から対応する図形の座標データが直接得られないためである。

多元管理的アプローチによる記述例と問題点

多元管理的アプローチでは、図形-図形関係テーブルの一部を図形専用テーブルに置き換えている。すなわち、一元管理的アプローチで用いていたテーブル群 (COORDINATE, LINE_COMPO, FACE_COMPO) をやめ、図形専用テーブル (LRT) を用いる。図-5 において、LRT は構成点数・線フラグ・座標値などを順に並べた一次元の配列の構造である。この図形専用テーブルの内容については、直接に関係データベース化するための変換は行わず、むしろ関係成分化を行うためのテーブル (COMPONENT テーブル) を介して、間接的な関係化を図る。これによって、名称や

属性から図形への検索は、汎用の DBMS 内の文字・数値情報処理として扱うことができる。さらに検索結果を直接座標データに換えることができるので、座標データを得るためのオーバーヘッドは軽く、高速検索が可能となっている。

しかし、図形定義の柔軟性から考えると問題がある。たとえば小平市に相当する面を定義する方法として、上水本町と学園西町に対応した二つの面から構成されるという解釈を与える場合を考える。一元管理的アプローチでは二つの面が共用する線 P₂P₆ の座標は、単一の記憶場所を参照すればよいのに対して、多元管理的アプローチでは、図形専用テーブル LRT でそれぞれ連続して管理しているので、共用する線を二重に定義せざるを得ない。

4.3 マルチメディアデータベースの新しい構成

以上の比較検討から、データ操作上多少難が残るものの、多元管理的アプローチの方が、マルチメディアデータベースを構成するのに有利であることが結論される。そこでこの多元管理的アプローチを基本にした新しいマルチメディアデータベースの構成例を提示する。図-6 にその内容を示す。この図の最下位レベルには、座標・テキスト・画像などメディア別の専用のデータ管理システムを配し、各データは一定範囲に分割したページ別に管理される。これらのメディア別専用管理のうち、たとえば座標の場合には図-5 で示した線登録テーブル (LRT) の形式で、隣接した座標点

が物理的に連続するように管理される。そして各メディアの管理の単位別にポインタを設け、それらを関係データベース管理システム (RDBMS) における各種のメディアの実体として用いる。RDBMS では、図形の幾何的な構造化のほか、すでに存在する属性データとの関係付けや、名称から高速にアクセスするための関係構造化などが行われる。

このような RDBMS とメディア別の専用管理システムとの連携による多元管理の効果としては、文献2) で示されたような、名称から図形を検索する速度の比較実験結果がある。それによると、物理的な座標データの連続性により、RDBMS 特有のソートマージ処理が省略されるため、RDBMS による一元管理処理に比べ約2桁以上の高速検索が確認されている。さらにこの新しいマルチメディアデータベースの構成に対して、図-6 の左側に示すような知識ベースを配している。この知識ベースには、今まで述べてきたマルチメディアデータベースのスキーマを中心とする意味構造を記憶しているほか、次章で述べる日本語インタフェースのための検索要求の意味抽出フレームや、道路・建物といった地図の主題に関する各種の知識及び検索後の結果を地図として見やすく表示するためのグラフィックス制御ルールなどが記憶される。

マルチメディアデータベースのユーザには、この知識ベースが直接見えるように構成されており、たとえばユーザがデータベース構造を知らなくても、データ

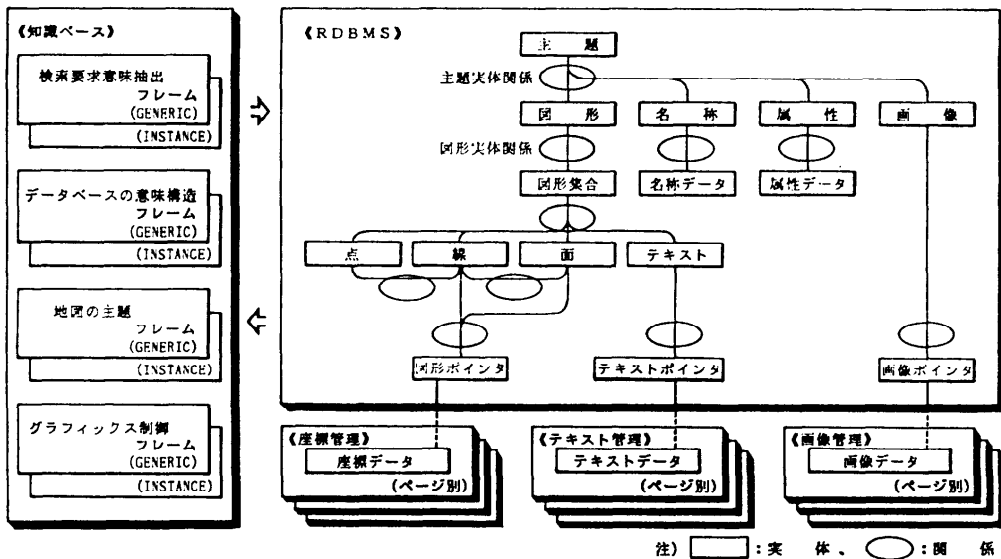


図-6 新しいマルチメディアデータベースの構成案

ベース意味構造フレームの自動起動により、目的とするデータが格納されているテーブルが存在するかどうかを推論することが可能となる。この推論メカニズムの詳細は次章で触れる。

なお以上述べたように、名称から座標といったトップダウン的な検索ではなく、座標から名称といったボトムアップ的な検索を行う場合の速度を向上させる方式も重要となる。この高速化手法としては BD 木構造化¹⁹⁾ やバケット構造化²⁰⁾ など、空間的に広がった座標データを局所的に分割管理するためのインデックス構造化を行う各種の手法がすでに存在する。このようなインデックス構造は、マルチメディアデータベースの最下層のメディア別専用管理システム内に効率よく配置可能で、このことも多元管理方式の優位な点の一つと考えられる。

5. 知識ベースによる検索の高度化

前章では、知識ベースを最上位に配置した新しいマ

ルチメディアデータベース管理システムの構成方式について概略を述べた。本章では特にその知識ベースに着目し、その役割とそれを用いた推論検索方式について補足するとともに、推論検索を用いた検索の高度化について述べる。

5.1 知識ベースの役割

冒頭でも述べたように、地図・図面情報処理システムの今後の課題として、マルチメディア化と高度分散化及び知的インタフェースの三つが考えられる。前節 4.3 で提案したマルチメディアデータベースにおいて、その最上位に配置した知識ベースはこれら三つのいずれの面においても重要な役割を果たす。この知識ベースの構造としては、対象指向のフレームにルール記述を可能にしたハイブリッド型を用いることができ、図-7 に示すような多重世界を容易に定義することが可能となっている。知識ベース内の各知識は、図-7 に示すように概念スキーマ・外部スキーマ・内部スキーマの三つの世界に分類でき、各知識の果たす

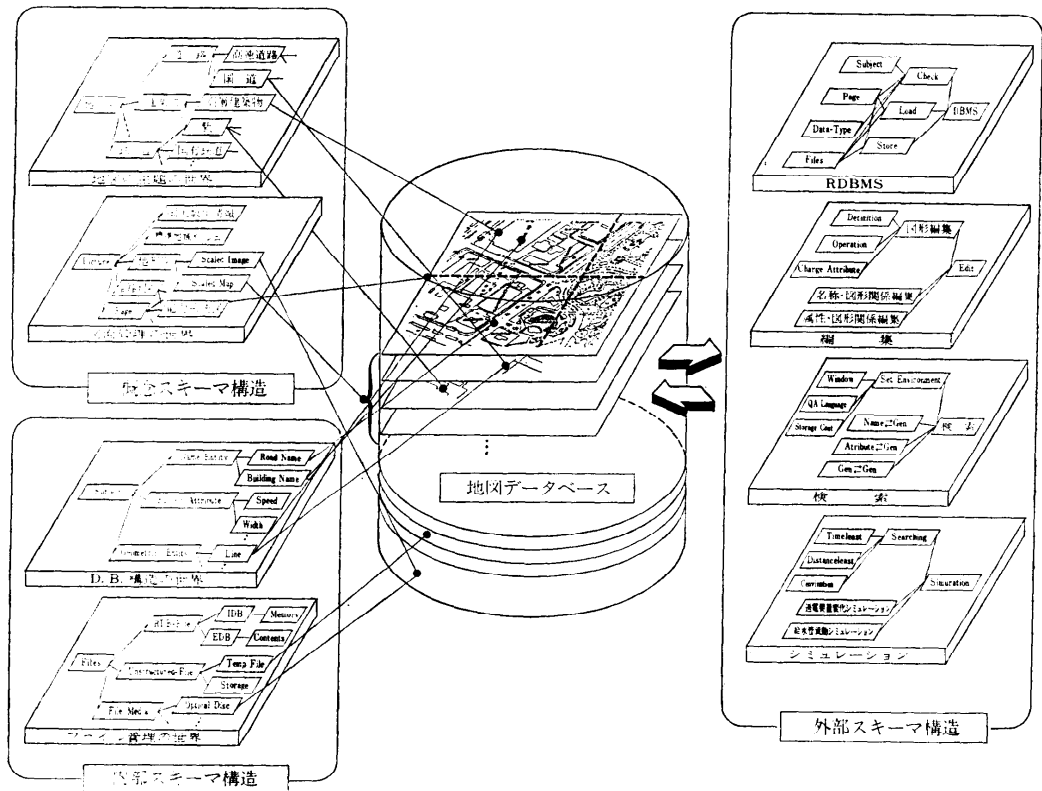


図-7 地図データベースの多面的解釈のための知識構造 (各種スキーマのフレームによる記述)

役割を次のように考えている。

(i) マルチメディア化への対応

すでに存在するデータベースの構造やスキーマ情報を内部スキーマ構造へ記憶することにより、メディアの種類別に散在するデータ管理を統一的に扱うことが可能となる。

(ii) 高度分散化への対応

各ワークステーションに分散して存在するローカルなデータベースの構造を知識ベースに記憶させ、どの位置に所望のデータがあるかを推論させることにより、効率のよい分散システムを実現できる。この内容は、上記マルチメディアへの対応用に設けた内部スキーマ構造知識を拡張することにより可能となる。

(iii) 知的インタフェースへの対応

ユーザにとって親しみやすい多様な世界（たとえば、地図は道路・建物などで構成されるといった主題の世界や、点・線・面といった幾何的な特性を明示した編集の世界など）を外部及び概念スキーマ構造知識に記憶することにより、ユーザの要求する処理内容に適した専用の環境が多様に定義できる。特に主題の世界として常識（たとえば、最高制限速度は高速道路の場合 100 km/h、一般道路の場合 40 km/h）を記憶させることにより、会話調の日本語などで指定された曖昧な検索要求を精密化するような推論が可能となる。

5.2 ハイブリッド型知識ベースを用いた検索の高度化

ハイブリッド型知識ベースを用いた検索の例として、文献 2) で実現した会話調日本語からの推論検索の内容について紹介する。説明をより具体化するために、処理対象とする地図は前章図-3 のモデルとし、検索要求の例文として「小平市上水本町に含まれる府中街道を赤で示せ」について説明する。この例文は、地名を副主部及び連体修飾部を備えた主部と、手段を示す連用修飾部を備えた述部とで構成される複文となっている。検索処理は、この検索要求文の「意味抽出部」と、意味抽出後の結果に基づくマルチメディアデータベースの「推論検索部」とに分かれる。これら二つの部分に前節の知識ベースの役割をあてはめると、意味抽出部は(iii)に相当し、推論検索部は(i)(ii)にそれぞれ相当する。以下各部の内容について詳しく述べる。

意味抽出部

この部分はさらに構文解析部と要求分析部とに分かれる。

《構文解析部》

この構文解析部では、いわゆる自然言語の翻訳問題としてはとらえていないため、自然な日本語で問題となる係り結びの曖昧性などの課題は扱っていない。むしろ問題となるのは、「小平市上水本町」といった地名や「府中街道」といった道路名称など、検索要求文に現れる固有名詞の扱いである。ここでは、これらの固有名詞を日本語辞書の中に記憶させないで、各名詞の接尾語や接頭語に着目した地名シンタックスを用いることにより、地名または道路名称であることを推定し、具体的な内容は未知のまま文法とのマッチングをとっている。この地名シンタックスの例としては、「～市～町～丁目～番地」や「～街道」、または「北～」などが考えられる。これにより日本語辞書の大幅な節約になり、構文解析が高速に行われる効果をもつ。

《要求分析部》

構文解析の結果得られる構文解析木は、検索要求文の形式に依存するため、後続の処理で扱いにくい。そこで、図-8(a)に示す形式の標準的な要求分析フレーム(QAF)に変換する。この変換過程で明確化されるのは、結果の取り扱い(PROCESS)、検索対象(OBJECT)、実行条件(WHERE)の3項目である。

図-8(a)のQAF-Genericでは、この3項目すべてが満たされるまでデーモン式起動により主題フレームの参照による不足情報の推論や、ユーザへの再問い合わせの発行などが実行される。そしてその結果を図-8(b)の様式のQAF-Instanceとして定式化し後続の処理に渡す。例文の分析後の結果としては、

{PROCESS: 線図形表示,

OBJECT: 町名称 (=「小平市上水本町」) を有する
町に包含される道路名称 (=「府中街道」)
を有する道路,

WHERE: 赤色の実線}

となっている。

推論検索部

前段までのQAFの抽出レベルでは、単に目標とする検索の対象が明確にされただけで、具体的なデータベース検索の可能性に裏付けられた意味構造には至っていない。そこで図-9に示す形式のデータベースの意味構造知識を参照して、目標とする検索対象に至るまでの具体的な検索手順を推論する。その結果を図-10に示す形式の要求解釈フレーム(QTF)に記憶する。以下このQAFからQTFを推論するまでの過程を要求例文を用いて具体的に説明する。

```

:QAF Generic
(FRAME
  (FRAME-NAME QAF)
  (FRAME-TYPE DEMON)
  (IF-LIST
    (*PARSE-TREE ((#IF-NEEDED (GET-PARSE-TREE))
                       (#PROP (NOT(NIL)))))
    (THEN-LIST
      (*PROCESS (#IF-NEEDED (FIND-PROCESS) ))
      (*OBJECT (#IF-NEEDED (FIND-OBJECT) ))
      (*WHERE (#IF-NEEDED (FIND-CONDITION))))))

```

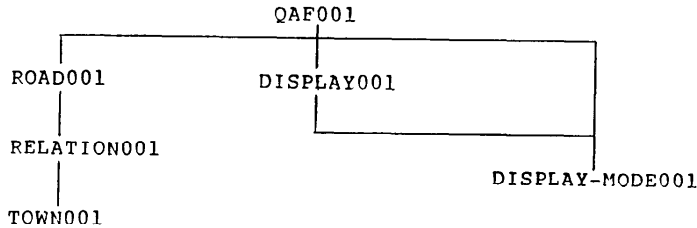
(a) QAF の GENERIC の内容

```

:QAF Instance
(FRAME
  (FRAME-NAME QAF001)
  (FRAME-TYPE DEF)
  (IF-LIST NIL)
  (THEN-LIST
    (*NODE (#VALUE (ND03)))
    (*PROCESS (#LINK (DISPLAY001)))
    (*OBJECT (#LINK (ROAD001)))
    (*WHERE (#LINK (DISPLAY-MODE001))))))

```

(b) QAF の INSTANCE の例



(c) QAF の INSTANCE 間の体系
 図-8 要求分析フレーム (QAF) の内容

まず QAF で明確になっている OBJECT に着目し、検索の開始位置を決める。そこでデータベースの意味構造フレームを参照すると、町名称と面の関係を扱う 'TOWN_NAME_FACE' テーブルと、道路名称と線の間を扱う 'ROAD_NAME_LINE' テーブルとが存在すると分かるので、検索開始点はこの各テーブルにおける TOWN_NAME カラムと ROAD_NAME カラムであると推論できる。そしてさらに、この両開始点から求まる町を示す面 (FACE.T) と、道路を示す線 (LINE.R) との間に包含関係を有するということが検索条件として推論される。

次に QAF の PROCESS に着目し、検索の終端位置を決める。PROCESS として得られているのは「線図形表示」であるため、最終的な検索対象としては、RDB フレームの参照により、LINE_COMPO テーブルの座標を示す COORDINATE カラムであると推

論できる。

以上により検索の開始点位置と終端位置及び検索条件が明確になったので、この後はデータベースの検索の世界における巡航問題として扱うことが可能となり、Prolog のようなバックトラック可能なプログラム言語で容易に推論が可能となる。Prolog を用いてこの推論を実行し、検索プランとして求めた結果を図-11 に示す。(a)は上述の条件を無視した場合、(b)はその条件を付加した場合、(c)はその中で最も検索テーブル数の少ないものを選んだ場合をそれぞれ示している。

さらに高度な推論検索

今まで述べてきた推論検索方式においては、推論過程での検索条件として図-9 の RDB017 に示すように、FACE.T_IN_LINE という面と線の包含関係があらかじめ記憶されたテーブルを用いているが、一般

```

:RDB Generic
(FRAME
  (FRAME-NAME RDB)
  (FRAME-TYPE DEMON)
  (IF-LIST
    (*PROCESS (#VALUE (CHECK-RDB))) )
  (THEN-LIST
    (*TABLE ((#NAME (GET-RDB-NAME))
              (#PLACE (CHECK-PLACE) )
              (#PROP (GET-RDB-PROP))) )
    (*FIELDS (#NAME (GET-FIELDS-NAME))) )
)

:
:RDB Instance
(FRAME
  (FRAME-NAME RDB017)
  (FRAME-TYPE DEF)
  (IF-LIST NIL)
  (THEN-LIST
    (*TABLE ((#NAME (FACE_T_IN_LINE))
              (#PLACE (EDB))
              (#PROP (GEO-GEO
                     FACE_T_LINE
                     CONTAIN))) )
    (*FIELDS (#NAME (FACE_T_LINE))) )
)

:
:RDB Instance
(FRAME
  (FRAME-NAME RDB020)
  (FRAME-TYPE DEF)
  (IF-LIST NIL)
  (THEN-LIST
    (*TABLE ((#NAME (CITY_NEIGHBOUR))
              (#PLACE (IDB))
              (#PROP (NAM-NAM
                     CITY_NAME CITY_NAME
                     NEIGHBOUR))) )
    (*FIELDS (#NAME (CITY_NAME
                     CITY_NAME_N))) )
)

:
:IDB Instance
(FRAME
  (FRAME-NAME CITY-NEIGHBOUR)
  (FRAME-TYPE DEF)
  (IF-LIST NIL)
  (THEN-LIST
    (*AKO (#VALUE (RDB020)))
    (*FIELDS (#NAME (CITY_NAME CITY_NAME_N)))
    (*CONTENTS (#VALUE (KOKUBUNJI_SHI KODAIRA_SHI
                        KODAIRA_SHI KOKUBUNJI_SHI))))
)

```

図-9 データベースの意味構造フレーム
(RDB: 外部データベース, IDB: 内部データベース)

```

:QTF Generic
(FRAME
  (FRAME-NAME   QTF (QAF-PROCESS QAF-OBJECT) )
  (FRAME-TYPE   DEMON)
  (IF-LIST
    (*QAF-LIST   (#IF-FILLED (NOT(NIL)))) )
  (THEN-LIST
    (*PROCESS    ((#DEFAULT (SET(QAF-PROCESS)))
                  (#IF-NEEDED (GET-PROCESS))) )
    (*TABLE      (#IF-NEEDED (FIND-TABLE)))
    (*FIELDS     (#IF-NEEDED (FIND-FIELD)))
    (*WHERE      (#IF-NEEDED (CHECK(QAF-OBJECT)))) ) )

```

(a) QTF の GENERIC の内容

```

:QTF Instance
(FRAME
  (FRAME-NAME   QTF001)
  (FRAME-TYPE   DEF)
  (IF-LIST      NIL)
  (THEN-LIST
    (*PROCESS    (#LINK (DISPLAY001)))
    (*TABLE      (#VALUE (ROAD_NAME LINE LINE_R IN LINE
                          LINE_COMP0 COORDINATE)))
    (*FIELDS     (#VALUE (ROAD_NAME LINE_R LINE POINT
                          (X Y))))
    (*WHERE      (#NAME (ROAD_NAME LINE.ROAD_NAME
                          ='HUCHU KÄIDO)))
                  (#LINK (QTFS001 (LINE_R IN LINE.LINE))) ) )

```

(b) QTF の INSTANCE の例

```

:QTFS Instance
(FRAME
  (FRAME-NAME   QTFS001 (LINE))
  (FRAME-TYPE   DEF)
  (IF-LIST      NIL)
  (THEN-LIST
    (*TABLE      (#NAME (TOWN_NAME_FACE FACE T IN LINE)))
    (*FIELDS     (#NAME (TOWN_NAME_FACE T LINE)))
    (*PROCESS    (#VALUE (LINE=FACE T IN LINE.LINE))
    (*WHERE      (#NAME (TOWN_NAME_FACE TOWN_NAME
                          ='JYOUSUI HONCHOU))) ) )

```

(c) 幾何関係がデータベース上に存在する場合

```

:QTFS Instance
(FRAME
  (FRAME-NAME   QTFS002 (LINE))
  (FRAME-TYPE   DEF)
  (IF-LIST      NIL)
  (THEN-LIST
    (*TABLE      (#NAME (TOWN_NAME_FACE)))
    (*FIELDS     (#NAME (TOWN_NAME_FACE T)))
    (*PROCESS    (#LINK (GEOMETRY001 (LINE_FACE T))))
    (*WHERE      (#NAME (TOWN_NAME_FACE.TOWN_NAME
                          ='JYOUSUI HONCHOU))) ) )

```

(d) 幾何関係を演算によって求める場合

図-10 要求解釈フレーム (QTF) の内容

```

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_C_IN_LINE CITY_NAME_FACE CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_COMPO FACE_C_IN_FACE CITY_NAME_FACE CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_COMPO FACE_T_COMPO FACE_C_IN_FACE CITY_NAME_FACE
CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_IN_LINE FACE_C_IN_FACE CITY_NAME_FACE CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_COMPO FACE_T_IN_FACE FACE_C_IN_FACE CITY_NAME_FACE
CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_COMPO FACE_COMPO FACE_T_IN_FACE FACE_C_IN_FACE
CITY_NAME_FACE
. ?)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_C_COMPO CITY_NAME_FACE CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_COMPO TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_COMPO FACE_T_COMPO TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_IN_LINE TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_COMPO FACE_T_IN_FACE TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_COMPO FACE_COMPO FACE_T_IN_FACE TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_C_IN_LINE FACE_C_IN_FACE TOWN_NAME_FACE)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_C_COMPO FACE_C_IN_FACE TOWN_NAME_FACE)

```

(a) 得られた検索プラン群

```

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_IN_LINE FACE_C_IN_FACE CITY_NAME_FACE CITY_TOWN)

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_IN_LINE TOWN_NAME_FACE)

```

(b) "FACE-T-IN-LINE" を使用するものに限定

```

(COORDINATE LINE_COMPO LINE_R_IN_LINE ROAD_NAME_LINE)
(LINE_R_IN_LINE FACE_T_IN_LINE TOWN_NAME_FACE)

```

(c) 使用テーブル数の少ないものに限定

図-11 検索プランの推論内容

にはこのような関係構造を与えることは難しい。そこでこのような幾何条件を直接記述したテーブルがないと判断された場合に、デーモン起動方式によって幾何演算手続きを自動的に起動するような記述をしておくことが可能である(図-10(d)にその記述例を示す)。これによって検索の制約が大幅に緩和される効果がある。

6. おわりに

今後重要となる地図・図面情報処理のためのマルチメディアデータベースに関して、主としてその構成面から技術状況を紹介してきた。とくに現在商用化されている汎用の RDBMS を活用できる多元管理的アプローチとしてのマルチメディア地図・図面データベースシステムと、その実現方法について詳述した。ここでは、さらに知識ベースの導入により、多元管理的アプローチの問題点を克服した新しいマルチメディアデータベースの構造を紹介した。しかしながら、名称からそれに対応した建物図形を検索するといった比較的単純な検索を行う場合にも、①知識ベース管理、②関係データベース管理、③図形専用管理といった3種類のデータ管理システムを使い分ける必要がある。このような処理はシステムオーバヘッドとして間接的に影響するため、処理系が複雑になったわりには、データ定義の自由度が少ないという問題が残っている。

今後、本稿ではあまりふれることのできなかつた直接的なマルチメディアデータベース管理システムによる一元的な管理方式(表-2の3)にも着目し、述語論理と密着したより理想に近い推論型マルチメディアデータベースシステムへと展開していくことも重要な課題となろう。

参考文献

- 1) 地図利用を革新するコンピュータマッピング, NIKKEI COMPUTER 1.6号, pp. 83-92 (1986).
- 2) 嶋田, 江尻: 日本語インタフェースを有する知識処理型マルチメディア地図情報処理システム GENTLE, 情報処理学会論文誌, Vol. 27, No. 12, pp. 1162-1173 (1986).
- 3) Nagy, G. and Wagle, S.: Geographic Data Processing, ACM Comput. Surv., Vol. 11, No. 2, pp. 139-182 (1979).
- 4) 松家, 杉本: 地域計画策定支援システムについて, 情報処理, Vol. 23, No. 9, pp. 818-827 (1982).
- 5) 松山, 三根, レ・ヴェト・ハオ, 長尾: 関数的

検索機能を有する地理情報処理システム MILES 情報処理学会論文誌, Vol. 23, No. 6, pp. 608-616 (1982).

- 6) 三枝, 三沢, 笠原, 霧谷, 斎藤 他: 地理情報処理システム WING, 第26回情報処理学会全国大会論文集, 6M-2~6M-6, pp. 1463-1471(1983).
- 7) Sakauchi, M. and Ohsawa, Y.: A New Interactive Geographical Information System Based on Effective Image-Type Map Representation, the 9th World Computer Congress, IFIP 83 (1983).
- 8) 林, 宮崎, 近藤: 住宅地図情報利用システムの開発, 日立評論, Vol. 66, No. 12, pp. 43-46 (1984).
- 9) 特集, コンピュータグラフィックスを利用した地図情報処理システム, 図形処理情報センター刊 PIXEL, No. 18 の諸論文 (1984).
- 10) 特集, システム化が進む地理情報処理システム, 図形情報処理センター刊 PIXEL, No. 39 の諸論文 (1985).
- 11) 宮武, 松島, 江尻: 平行線抽出手法を用いた地図からの道路情報自動抽出, 電子通信学会論文誌, Vol. J68-D, No. 2, pp. 153-160 (1985).
- 12) 恒川, 下辻: 図面読取装置 TOSGRAPH, 電子通信学会論文誌 (画像処理特集), Vol. J68-D, No. 4, pp. 466-472 (1985).
- 13) 角本, 宮武, 嶋田, 江尻: 実時間色コード化技術を用いた多色図面の自動認識, 電子通信学会論文誌 (画像処理特集), Vol. J68-D, No. 4, pp. 829-836 (1985).
- 14) 長田, 井上, 吉田: 論理回路図の自動入力処理, 電子通信学会論文誌 (画像処理特集), Vol. J68-D, No. 4, pp. 837-844 (1985).
- 15) 大沢, 坂内: 多次元データ構造を用いた図面処理, 電子通信学会論文誌 (画像処理特集), Vol. J68-D, No. 4, pp. 845-852 (1985).
- 16) 宮武, 松島, 江尻: ラン型方向コードに基づく圧縮2値画像の高速輪郭追跡手法, 昭和62年度電子通信学会全国大会予稿 (1987).
- 17) 大保, 他編: CAD/CAM のデータベース, 『CAD/CAM』, bit 増刊号, 共立出版 (1985).
- 18) リレーショナル型データベース16種類を総合評価する, NIKKEI BYTE/NOVEMBER, pp. 102-131 (1986).
- 19) Bentrey, J. L.: Multidimensional Bynary Search Tree Used for Associative Searching CACM, Vol. 18, No. 9 (1975).
- 20) Edahiro, M., Kokubo, I. and Asano, T.: A New Point-Location Algorithm and Its Practical Efficiency—Comparison with Existing Algorithms, ACM Transactions on Graphics, Vol. 3, pp. 86-103 (1984).

(昭和62年4月28日受付)