

オブジェクト指向に基づく 住宅地図データベースシステムの構成

打浪 清一¹, 荒木 信吾¹, 津留 義信²
尾崎 直人², 吉本 祐一², 石井 泰仁³, 山下 泰英⁴

¹ 九州工業大学 情報工学部, ² ゼンリン
³ エーエスエー・システムズ, ⁴ ハ幡コンピュータセンター

ゼンリンが作成した住宅地図を基に、オブジェクト指向データベース管理システムVersant を用いて、オンライン住宅地図データベースシステムを構築した結果に就いて述べる。

印刷用に構築された住宅地図データベースは、印刷には効率的であるが、対象物がオブジェクト単位にまとまって無く、また、各オブジェクト間に位相関係があるが、オブジェクト指向モデルでは位相情報の記述処理に全く無力なので、これらの点を改善した住宅地図データベースシステムを試作した。その方法とシステム概略に就いて述べる。

マンマシンインターフェイスとしては、GUIを用い、特殊な検索用言語を知らなくてもマウスの操作と、キーワード入力で検索や表示を行えるように実装した。

A ONLINE DETAILED MAP DATABASE SYSTEM BY MEANS OF AN OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEM

Seiichi UCHINAMI¹, Shingo ARAKI¹, Yoshinobu TSURU²
Naoto OZAKI², Yuuichi YOSHIMOTO², Yasuhito ISHII³, Yasuhide YAMASHITA⁴
¹ Faculty of Information Science and Systems Engineering, Kyushu Institute of Technology, ² Zenrin, ³ ASA Systems, ⁴ Yawata Computer Center

¹ 680-4, Kawazu, Iizuka, Fukuoka 820, JAPAN

² 3-1, Shinmachi, Nakahara, TobataKu, Kitakyushu, 804 JAPAN

³ MainichiKaikan Buil. 5F, 13-1, Konyamachi, KokurakitaKu, Kitakyushu, 802, JAPAN

⁴ 2-8-13, Chuoh, YawataHigashiKu, Kitakyushu 805, JAPAN

This paper describes a new online detailed map database system based on object oriented method. In map systems topological information such as far-and-near relation and along relation are important, but OODBMS has no abilities to treat these relations. We implement a database system that can be retrieved from various topological properties among objects, and properties of objects. We implement familiar GUI window-base interface. We can retrieve maps by icons and without using object oriented language.

1. はじめに

地図データベースは各構成要素間に位相関係があり、これらの記述処理は現存するどのタイプのデータベース管理システムでも行なえない。しかし、オブジェクト指向データベース管理システムは、マルチメディア向きと言われており、位相的性質を記録処理するには、少ない機能付加で実現できるのではないかと考え、プロトタイプシステムを試作した。ゼンリンが印刷物用に収集し、計算機可読型のデータとして蓄積してきた地図データベースを基に、そのデータ構造を変換し、必要な情報を付加して、種々の利用が出来るオンライン住宅地図データベースシステムを構築した。以下にその概要を述べる。

2. 要求分析とシステム分析

2.1 地図の利用法と特徴

紙に印刷された地図と計算機が管理する地図データベースの使われ方を比較する。現在の印刷物の地図の欠点としては、以下のものが挙げられる。

- (1) 1頁毎に区切られて、境界部分は読みにくい。
- (2) 住所から検索時に、索引無しには検索にくい。
- (3) うろ覚えの住所で検索事が難しい。
- (4) 人名やビル名、企業名からは検索せない。
- (5) 地形的な特徴から検索と能率が悪い。

データベースシステムの長所としては

- (1) 所有していないても、DBセンターにアクセスする事により入手可能。
- (2) 種々の観点から、条件に合う地図を入手可能。
- (3) 検索した地図を、必要な情報のみを表示する形に、編集して、出力する事が可能。

2.2 現在の地図データベース

現在のゼンリンの地図DBの特徴は以下の通り。

- (1) メッシュ構造：地表がある大きさの矩形に区切りメッシュと呼び、それを単位として管理を行う。メッシュで管理することにより、検索処理や描画処理などで高速化が期待できる。
- (2) レイヤ構造：地図上の图形を同じ意味ごとにまとめてひとつのレイヤ（層）として管理する。目的の地図はこれらレイヤの中から必要なものだけを重ね合わせた合成図として表現できる。
レイヤはほぼオブジェクトの分類と一致しているのでクラスを作成する場合の参考になる。

以下のようなレイヤーが有る。水城ボリゴン、高道路・有料道路、国道、主要道路、一般道路、JR本線・支線、特殊軌道・モノレール、私鉄本線・支線、河川、海岸線、湖・池・沼・プール、細流・澗河、橋、歩道、耕地面界、地類界、等高線、構団、地形形状など。

以上のデータベースシステムに於ける地図の特徴を生かしたシステムの検討を行った。

2.3 システムの利用環境

次のような使用法の下で有効なシステムの開発を検討した。センターにDBを置き、通信回線を介して地図DBに以下の2方法でアクセスする。

- 1) 利用者は、ワークステーション(WS)、パソコン、或いは電話で、センターDBにアクセスし、必要な区画の必要な情報の描かれた地図を通信回線を介して、入手する。

入手形態としては、パソコン通信のような形でデジタルでデータを受信する方法と、FAXにより受信する方法が考えられる。

利用法としては、葬式の会場とか、食事をするふさわしい店とその地図が欲しいというような質問に對して検索、FAXで返事するもの等。

- 2) 利用者は、ゼンリンの地図DBをよく使う地域に関しては、CD-ROMで所有し、自社内で地図データベースを利用している。その後の更新情報を入手したり、所有している地域外のデータを検索し、通信回線を介して入手するか、或いは、FD、テープで送って貰う方法を取る。

2.4 検索質問のパターン

自由な質問が許されれば良いが、4種類を設計し、最初の2種類について実装した。

1) 質問のパターン

- (A) 対象物を指定、対象物を含む地図を検索し、指定倍率、指定範囲、指定項目を表示する。
(対象物を陽に指定。)

対象物としては、点、線、領域、網がある。
ここで、点とは各住宅、駅等、線とは道路、鉄道等、領域とは町、区等、網とは、交通網等を言う。
例a) XX氏の家の近傍1km区画を表示せよ。

- b) 小倉北区魚町1-1-1の近所を表示せよ。
- c) XX氏の家と小倉駅の双方を含む地図を最大倍率で表示せよ。

- (B) 対象物の型を指定、属性を規定し、条件に合う

対象物を検索し、指定倍率、指定範囲、指定項目を表示する。（対象物を陰に指定。）

対象物としては、点、線、領域、網がある。

属性や属性値の指定方法：以下のものが有る。

属性値 数値による範囲 道幅が5m以上

業種 医院 クリニック（類語集の利用）

位相 近い、沿い

計算による 交差点、経路

例a) Y百貨店の近くのイタリア料理店を知りたい。

b) 博多駅から博多日航ホテルへゆく道筋を示せ。

c) 小倉北区内で5階建て以上の建物を地図上に階数と共に建物名を表示せよ。

d) XX氏の家と最寄りのJR駅の双方を含む地図を最大倍率で表示せよ。

e) XX氏の近傍200m以内の駐車場とその電話番号を表示せよ。

f) 八幡北区の国道3号線沿いのガソリンスタンドを表示せよ。

(C) 対象物変数に対し、限量子を掛けた論理表現式を満たす対象物を求め、指定倍率、指定範囲、指定項目を表示する。（対象物を陰に指定。）

例a) 5階建て以上のビルは、全て商店が入っていて、住居ばかりの5階建て以上のビルが無い町を八幡西区内で検索。

b) XX氏の家の近くを通過するバス路線の中でJRの駅を通る路線を表示せよ。

c) XXXを起点に、住所1..住所50を巡回して元の所に戻る最短経路を地図上に表示せよ。

(D) 対象物型を指定、ユーザのDBと結合を取り、条件に合う対象物を求め、抽象化等の処理を行い、結果を表や、地図上にプロットして表示する。

（分散データベース利用）

例a) 小倉北区内にある当店の顧客中で、本年度の買い物上げ額が100万円以上の客に対して、地図上に買い物上げ額を棒グラフで表示してほしい。

2.5 データベース管理システム

[A] DBMSのタイプの選定

DBMSには現在6種類のタイプがある。

(1) 階層型、(2) ネットワーク型、(3) リンクファイル型、(4) 関係型、(5) 転置ファイル型、(6) オブジェクト指向型。現在の地図DBシステムは独自開発型、関係型のDBMSを採用している。

地図データベースの利用法は、ある意味では、定型と言える。しかし、複数タイプの質問に応えるには、階層型は駄目である。ネットワーク型、リンクファイル型なら可能であるが、応答速度の点から、転置ファイル型に劣る。関係型は複数属性の質問も可能であるが、遅いのが難点である。よって転置ファイル型が適している。

現在考慮しているDB環境は、大型計算機ではなく、WS程度のものを考えている。このクラスの上で稼働しているDBMSは、関係型が殆どで、オブジェクト指向型が幾つかある程度である。転置ファイル型はADABASがHPの上で動き始めたという程度であまり使える環境でない。関係型のDBMSは汎用型であるが応答速度が遅い。それ故、転置ファイル型の地図用DBMSの開発を検討したが、時間とマンパワー、費用の件で難しく、まだあまり使用実績はないが、オブジェクト指向型で作るのが現環境の下では良いと判定した。オブジェクト指向では、クラス階層を定義し、その中にメソッドを埋め込むが、メソッドも継承出来るので、差分プログラミングで良く、プログラミングの労力の軽減が期待される。しかし、まだ、プログラマがオブジェクト指向プログラミングになれていないので、そのメリットが充分に生かされるかどうか疑問が残る。また、OODBMSは速度の点で不安が残る。理由は、レコード間の関係は、リンクを用いて表現する。その点では、リンクファイル型に似ている。ところがリンクファイル型と転置ファイル型と比べると、転置ファイル型の方が応答速度が良い。それ故、オブジェクト指向型と転置ファイル型とベンチマークテストをして比較評価しなければ分からぬが、リンクを採用するのはベストとは言い難い。オブジェクト指向型信者は応答速度の遅い関係型と速度比較して勝ったといっているが、これは意味がない。

現在のオブジェクト指向DBMSは、クラス階層を修正すると、それに付随するインスタンスが整合性を保証されない。属性の多重継承は当然必要で、これをサポートし、クラスの変更が出来ないとDBMSとしては、不充分である。このように、現在注目を浴びているが、まだ、あまり実用には供されていないし、不完全である。またDBMSの基本部分が不完全なので、その部分はこちらでプログラムをして、補充する必要がある。

このように欠点も多いが、取りあえずオブジェク

ト指向DBMSを用いる事にした。

[B] オブジェクト指向DBMSの特徴

- ・任意のデータ構造が使用できるので現実世界に即したモデリングが可能。
- ・検索されたオブジェクトを直接プログラム内で利用できるので変換が不要。
- ・プログラム言語と同一の言語でデータベースへの問い合わせが可能。

欠点

- ・データモデルに厳密な数学的根拠がない。
- ・ビューのサポートが貧弱。
- ・スキーマ（オブジェクト指向DBMSではクラスの定義）の変更が柔軟に行えない。
- ・データベース本来の機能（データ独立性やトランザクションの管理など）が未成熟。

[C] オブジェクト指向DBMSの選定

WangのOODBMSに関する比較評価レポートの中で評価の高く、日本に販売代理店のあるものに絞って問い合わせた。回答のあったObjectStore・ONTOS・Versantの3つのオブジェクト指向DBMSについて、より詳しく比較検討した。実際に、地図データベースとしてオブジェクト指向データベースを使用する場合地図データの特徴を考慮して、必要な機能：(1) 言語に関する事、(2)データベース機能、(3)開発環境、(4)サポート体制に関する13項目の質問を行い、その結果を評価、集計して、最も良かったVersantを採用した。

[D] Versantの概要

OODBMS Versantは、米国Versant Object Technology社の製品である。DB記述言語としてC、C++、SmallTalkが選択できる。C++を採用した。以下にVersantの特徴を挙げる。

- (1) PObjectなる永続性クラスが提供される。

(2) 非永続オブジェクトと区別するため、DBに記録するインスタンスの生成には、“new Persistent クラス名”という構文を用いる。

(3) インスタンスの削除は通常のC++の“delete インスタンス名”の構文で行える。

(4) DB内のオブジェクトを通常のポインタで指せず、ポインタに代えてLinkクラスを用いる。

Linkクラスのインスタンスは、ほぼポインタと同等の機能を持つ。

(5) 検索処理はクラスオブジェクトに対してselect メソッドを使用する。検索条件の指定はオブジェクトの属性に対し、数値の指定、数値範囲の指定、文字列の指定とそれらの論理積で行う。

(6) select処理の高速化のため、クラスの指定した属性に対しHashTableとB-treeがサポートされる。

(7) 検索結果は可変長のリンククラスであるLinkVstrクラスのインスタンスで返される。

(8) ネットワークがサポートされているのでLAN接続により、分散DBを構築することができる。

3. システム設計

3.1 システム構成

システム構成を、図1に示す。

(1) 地図データベース

地図DBは、SUN/4上にVersantを用いて構築する。利用者側としては、イーサネットで接続されたWSからDBをアクセスするシステムと、通信回線を介してパソコンからDBをアクセスするシステムの2種類を準備する。

種々のクライアントに対して柔軟にデータを提供できる地図DBの構築を目指し、以下の要件を考慮し作成効率を考え5モジュールに分割した。

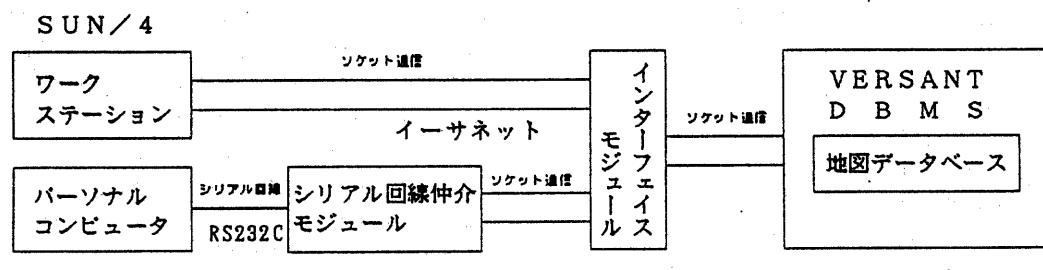


図1 システム構成図

- 1) 遠隔地から電話回線などを用い、地図DBへのアクセスを可能にする。
- 2) LANで接続されているクライアントからも地図DBへのアクセスを可能にする。
- 3) パソコンなどの能力の低いクライアントに対してもDBのもつ全ての機能を提供する。能力的に無理なものはクライアント側で使用を制限する。
- 4) 仕様の変更に柔軟に対応できるようとする。

1)に対応するためには、外部とシリアル回線で通信して、DBとの通信を仲介するモジュールが必要となる。また、4)については、質問に対してDB内部のクラスのメソッドで全ての処理をしていたのでは仕様が変更になった場合、クラスの定義を変更せざるを得ない。よって、各クラスのメソッドはプリミティブな処理を行うだけにとどめ、DBとクライアントを直接接続するのではなく、間にインターフェイスモジュールを入れる。

(2) マンマシンインターフェイス部

関係型のDBMSのEULは、標準化されており、SQLとQBEがあるが、OODBMSのEULはまだ標準化されていない。C++などを使って利用者が検索命令を出せるはずがない。その点ではOODBMSはまだ未完成である。Versantには、ObjectSQLがあるが、もう一つである。

地域限定

福岡県	市郡	区	町村	大字
福岡県	北九州市	小倉北区	魚町	

オブジェクト限定

検索条件： クラス名 オブジェクト名 近傍条件 件数表示 地図描画

処置： AND OR

検索履歴：

検索条件	検索結果
福岡県北九州市小倉北区魚町	121
福岡県北九州市小倉北区魚町, 近傍(国道3号線, 40m, 道路), 種類(ガソリンスタンド), オブジェクト(建物)	2

図2 検索要求画面のルックアンドフィール

ウィンドウ上にアイコンを表示し、マウスでこれを選択する方法は、メニュー形式でシステム主導型であるが、素人には使いやすい方法である。

地図はグラフィックデータであるし、利用者は人がほとんどと考えられるので、検索・表示のためのインターフェイスは、GUIで作製する。

1回の質問で目的の対象物が絞れるとは限らないので、漸次絞って行く事が出来るようにし、検索履歴を持ち、前に戻って検索を続ける事も出来るようになる。検索画面例を図2に示す。

(3) 通信部

マンマシンインターフェイス部での利用者入力と、Versantに於けるデータ検索とはかなりレベルが異なるので、その中間段階のマクロコマンドを設定し、このレベルで通信を行うことにした。このレベルでの通信プロトコルを定め、それを実装した。WS間はソケットで、WSとPC間はRS232Cで結んだ。

地図データ伝送のプロトコルは、(1) 文字、(2) 線、(3) ポリゴン、(4) マークの4種類準備した。

3.2 地図データベースシステム設計

[A] 地図スキーマの設計

オブジェクト指向パラダイムの特徴としては、以下のものが挙げられる。これらを活かすような方法で地図概念スキーマを設計した。

(1) オブジェクト化、カプセル化

地図には線で描かれた图形と、それに付属する属性からなる。图形情報としては、点・線・面などの形状とその位置情報があり、付属情報としては、名前、住所、電話番号などがある。印刷用のDBでは、これらがオブジェクト化されていないのでこれをオブジェクト化する。例えば、道路では、現在、国道レイヤーには国道に面する家と道との境界線が飛び飛びに入っているだけで、国道3号線という対象物に関してのまとまりが無い。そこで各オブジェクト単位にまとめ構造化を行う。

(2) クラス階層と属性継承

is_a_kindで定義されるクラス間関係は上位下位の階層関係をなし、上位クラ

スに記述された。属性、メソッドは下位クラスに継承され流用できる。

クラス階層に関しては、永続性オブジェクトクラスの下に、地図クラスを置き、その下に点、線、面、ネットワーククラスを置く。これらは抽象クラスで、その下に具体クラスを置く。

- ・点クラスは、建物クラスを下位に持つ。

(一組の座標によって管理される)

- ・線クラスは、軌道、河川、道路クラスを下位に持つ。(折れ線で管理される)
- ・面クラスは行政区界、メッシュ等の下位クラスを持つ。(輪郭と塗りつぶし情報をもつ)
- ・ネットワーククラスは、道路、河川、軌道など1つのメッシュに入りきれず、ネットワーク状をなすものについてグローバルな接続関係、位置情報等を持つ。詳細な道路の両側の境界線等は線クラスで持つ。マクロな情報なので、マクロクラスと呼ぶ事にした。
- ・軌道、河川、道路マクロクラスを下位に持つ。
- ・業種を指定し店を探すのを高速化する為、業種階層を設け、これと店名などをリンクで関連付ける。業種階層は第一階層は、公官署、学校、販売業、病院、文化施設、サービス業、その他とし、サービス業の下に飲食業、娯楽...のように4階層の分類を作成した。

- ・人に関しては地理とは別に人クラスを設け、氏名、住所、電話番号、職業等を属性として持たす。建物オブジェクトは人オブジェクトに対する可変長リンクをもつ。
- ・メッシュクラスは前述のメッシュ構造をクラスにしたもので、下位にサブメッシュを持つ。
- ・行政単位クラスはサブメッシュを下位に持つ。
- ・サブメッシュクラスはその矩形領域の内部にある建物オブジェクト等の検索、描画を管理する。
- ・背景：今回は試作であるので、あまり細かなクラス分けはせずにその他の表示物は文字背景、シンボル背景、線背景、面背景の何れかとした。

(3) 差分プログラミング

新しくクラスを追加する際には、作成したいクラスの持つ属性、メソッドから考えてそのすぐ上のクラスを見つけてその下に新しく作る。メソッドも無い部分、異なる部分を作るだけで済む。

(4) クラス階層以外の必要な関連の種類の設定とそのインプリメント法

オブジェクト指向ではis_a_kind_of関係を特別扱いして他の関係は無視するが、地図情報を記述するには、このクラス階層だけでは足りない。他の関係に関してはリンクをつなぐ程度の低いインプリメントしか考えられていない。

is_a_part_of, has_a, topology関係の記述法を決定する必要がある。

表示を考えると、近傍にあるオブジェクトはまとまっていた方がよい。その為に、各メッシュ内にあるオブジェクトをまとめてアクセスし易いメカニズムを導入する。メッシュから、その中にあるオブジェクトに対してリンクを張るか転置ファイルを持つ。今回はVersantが転置ファイルはサポートしていないので、並のオブジェクト指向風にリンクで試みた。

クラス階層図を付録図に示す。

[B] 地図検索法

検索を行なって絞っている途中や、あるメッシュ内のオブジェクト集合など中間結果を格納する容器としてバッグを設ける。

検索依頼は、範囲限定された最上級の行政単位オブジェクトにメッセージを送り、それが下位に送られ、該当するオブジェクトだけが自分をバッグに入れる。

条件を更に追加しながら対象を絞って行き、目的の物が得られたら表示する。

これらの中間結果は夫々、異なるバッグに入れられ、有る程度過去の物まで一時的に保存される。

ワークステーションの場合は、クライアント側にバッグの内容を取り寄せ、スクロール、拡大・縮小に応える。パソコンの場合は、伝送量、伝送時間、記憶容量の点で最低必要限のものしか伝送しない。

[C] 位相検索法の設計

地図DBシステムで最も問題になるのは、「~沿い」、「~の近く」といった位相関係に関する検索である。この様な近傍検索についてどのようなデータの持ち方、検索方法があるかを検討し、各方法を定性的に評価、そのうち、最良の方法については、実装、評価、実現性の確認を行なった。

C.1 近傍検索の方法

「~の近く」という関係は、点と点、線と点、線と線の組合せがある。ここで、「線と点」は「~沿いの」と同じ様な意味合いになる。

点と点の近さは2点間の距離を検索すれば良く簡

单である。線と線の関係としては、並行して走っているとか、直交しているなどの関係があるが、この問い合わせは少ないと思われる所以、点と点、線と線に関する近傍検索を主に検討した。

- 質問の形式としては、次の様なものを想定した。
- ・ 山田氏の家に近いバス停を表示して欲しい。
(点から点の検索)
 - ・ 3号線沿いのファミリーレストランを表示せよ。
(線から点の検索)
 - ・ 小倉駅に近い国道を表示してください。
(点から線の検索)

最後の質問の形式は、比較的少ないと思われる。検索のための方法には、次のようなものがある。

- (1) 転置ファイルを用いる方法
- (2) 転置ファイルクラスを用いる方法
- (3) クラスに近傍の属性を持たせる方法
- (4) 近傍のインスタンス同志を双リンクする方法
- (5) 隣接インスタンス同志を双リンクする方法
- (6) クラスに距離計算メソッドをつける方法

C.2 検索法の決定

検索方法の決定をするためにそれぞれの方法を比較検討して、実現する方法を決定した。

データ量、検索速度、データ変換、範囲指定、地域限定などの観点から比較したものを表1に示す。

この評価結果より、クラスに近傍を求めるメソッドを入れる方法を採用した。

表1 位相検索法の評価

評価項目	データ量	検索時間	データ変換	指定範囲	地域限定
転置ファイル	C	A	D	×	△
区分転置ファイル	D	B	E	△	△
近傍属性付加	C	C	B	×	○
距離属性付加	E	D	C	△	○
近傍リンク	C	C	B	×	○
近傍区分リンク	E	D	C	△	○
隣接インスタンスリンク	B	D	D	△	○
計算メソッド付加	A	E	A	○	○

データ量 A(データ量少)… E(データ量多)

検索時間 A(早い)… E(遅い)

データ変換 A(手間小)… E(手間大)

範囲指定 ○(可) △(有る程度) ×(不可)

地域限定 ○(可) △(有る程度) ×(不可)

C.3 検索法

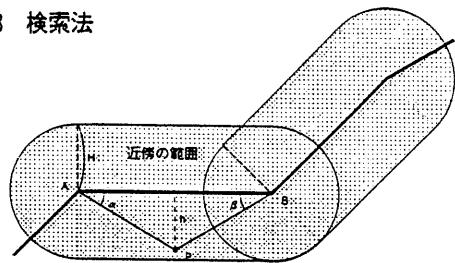


図3 近傍の計算方法

図3のように、道の中央線を構成する線分を包含するトラック状の範囲に建物の重心が存在すれば、その建物はその道の近傍だと判定する。この範囲内の判定は、次式のようになる。

$$0 \leq \cos \alpha \leq 1 \& 0 \leq \cos \beta \leq 1 \& h < H \\ \text{or } AP < H \text{ or } BP < H$$

4. データベースシステムの構築

4.1 データのコンバート

[A] ゼンリンの住宅地図データベースより簡単に抽出できる情報は、以下の通りである。

- (1) 建物について：形状（座標列）、建物の名前、その建物が属しているメッシュ番号
- (2) 道路について：形状（座標列）、その道路が属しているメッシュ番号、（線に関して同様）
- (3) メッシュについて：メッシュ番号、そのメッシュに属す建物、そのメッシュに属す道路。

[B] 近傍検索に必要であるが、住宅地図DBからは抽出できない情報には以下のものがある。

- (1) 道路の中央線 (2) 道路の名前（国道3号線など）
新規に入力の必要な情報には以下のものがある。
- (1) 職業（業種）、(2) 電話番号

[C] データベースの容量

ゼンリンの地図データを元に北九州市小倉北区の地図DBを構築した。データ変換の際には不要データは省き、新規に必要なデータは追加されてデータ容量の正確な比較はできないが、元データ（圧縮済）の2倍弱の容量（約150MB）となった。

4.2 データベースシステムとしての機能

[A] インターフェイスモジュールの機能

インターフェイスモジュールは図1に示したように、クライアントとDBMSの間に位置する。クライアントとインターフェイスモジュールの間では専

用の通信プロトコルを用意しており、インターフェイスモジュールはこのプロトコルを解釈しDBMSに対する要求を実行する。また、DBMSからの処理の結果も通信プロトコルに変換されてクライアントに伝えられる。例えば“国道3号線”的検索がクライアントから要求されると、インターフェイスモジュールに対して“国道3号線”という名前をもつ道路マクロオブジェクトを検索し、見つかった道路マクロオブジェクトにリンクをもつ道路オブジェクトを検索対象地域から選び出し、結果をプロトコルに翻訳してクライアントに返す。

過去の検索結果を記憶し、必要に応じ後戻りし再検索出来るように時限記憶としてのオブジェクトバッグをインターフェイスモジュールに設けた。

[B] 検索処理

検索処理は、建物検索、住人検索、行政単位検索、道路・軌道・河川（線）検索がある。

- (1) 建物検索：住所と建物名を指定する。普通の住宅は建物名が無いので世帯主名を建物名の代用とする。建物名は部分一致検索する。
- (2) 住人検索：住所、建物名、部屋番号、住人名、電話番号、職業から検索出来る。建物名、部屋番号、住人名、電話番号に関しては文字列の部分一致を取り。職業（業種）は簡単な階層的分類を定義し、プルダウンメニューで選ぶ。
- (3) 行政単位検索：行政単位名を指定し、行政単位ごとに文字列の完全一致を取り。
- (4) 道路・軌道・河川検索：道路名を指定。道路名は文字列の完全一致検索を行う。
- (5) 位相検索は前述の方法で行なう。

4.3 マンマシンインターフェイス

OpenWindow、MSWindow上で、以下の検索・表示が行なえる。

[A] 住所からの検索

- (1) 住所を指定して該当地域を検索する。
大きい行政区画はプルダウンメニューで選択。
小さい町丁番等は漢字入力する。
- (2) 画面上で領域指定しその範囲内で種々の検索を行う。

[B] 属性検索

以下の属性を指定して検索する。

- (1) 建物名、(2) 番地まで、(3) 世帯主名、(4) 業種、(5) 電話番号。

[C] 位相検索

- (1) ~沿いを指定して検索する。
- (2) ~の近くを指定して検索する。
- (3) これらを組み合わせて指定しての検索。

[D] 表示

- (1) 検索オブジェクトの強調表示
- (2) 画面のズーミング（スライダー、範囲指定）
- (3) スクローリング
- (4) 背景表示レベルの変更
- (5) 画面上でピックしたオブジェクトの属性表示
- (6) 検索履歴による過去の検索への戻り
- (7) 検索履歴を用いてのAND、OR演算。

5. 検討・評価

時間がなく定量的な評価やチューンナップはこれからであるが、現時点までの評価を述べる。

5.1 検索処理の高速化

Versantが提供する唯一のオブジェクトの検索方法はselectメソッドである。しかし、selectメソッドはある属性値をもつオブジェクトをクラス全体から探し出す処理である。ある特定のオブジェクトだけに対する検索はできない。例えば、探したい建物の住所が分かっている場合、その住所に当たる領域だけを探す方が全ての建物オブジェクトから探し出すよりも高速な処理ができる。特に地図の場合はオブジェクト数が膨大な数になるためこの違いは顕著である。そこで、selectメソッドは極力使用せず、オブジェクトがリンクを辿って検索を行うメソッドを作成するという方法を採った。この方法の利点は特定のオブジェクトに限った検索ができるの他に、検索条件の詳細な設定ができることが挙げられる。例えば文字列の部分一致処理を行ったり、属性値によって別のオブジェクトのメソッドを起動したりなどである。例としてX市のYさんを探す場合を考えてみる。

- (1) X市に属するSubMeshObjをObjBag1に入れる。
- (2) ObjBag1内の全SubMeshObjに対してYさんを探すMessageを送る。
- (3) SubMeshObjは自分がリンクをもっている全ての建物Objに対してYさんを探すMessageを送る。
- (4) 建物Objは自分がリンクをもっている全ての人Objに対してYさんを探すMessageを送る。
- (5) 人Objは自分がYかどうかチェックを行い、そ

うであれば指定されたObjBagに自分を登録する。以上の処理でX市内の全てのYさんが指定したオブジェクトバッグに登録される。

5.2 地図描画処理の高速化

地図描画処理は他の処理に比較して扱うデータ量が非常に多い。図1に従うと描画されるデータはDBMSから一旦インターフェイスモジュールに渡され、それをインターフェイスモジュールがクライアントが処理できる形に翻訳して出力することになるが、この部分にかかる時間は大きいと予想されるため、描画データの変換が必要ないように各クラスの描画メソッドがクライアントとの通信プロトコルに従った形でデータを出力するようにした。また、その変更によりインターフェイスモジュールを介す必要がなくなるため、メソッドの引数にソケットのファイルディスクリプタを渡し、メソッドがクライアントへのソケットに直接出力するようにした。

5.3 C++を使用したデータベースの記述

オブジェクト指向DBMSの特徴として通常のプログラム言語をそのままDB記述言語として使用できることが挙げられるが、逆に言えば一般的なオブジェクト指向モデルをサポートできるDB記述言語を作ることは非常に難しい。オブジェクト指向モデルはかなり抽象度の高いモデルであり、言語によって利用できる機能に差がある。C++で記述されたDBに対してはC++以外の言語で利用することは困難である。よって使用言語の能力がそのままDBの機能を支配する。今回はC++を使用したがその経過で気付いた点を挙げる。

(1) オブジェクト指向とデータベースの不整合

DBの利用では、あるクラスの全オブジェクトの特定の属性の値だけが知りたい場合がある。オブジェクト指向モデルではデータ処理の最小単位がオブジェクトの為、関係のない属性まですべてメモリ内に読み込んでしまうため効率が悪い。

(2) プログラムの生産性

DB内部のクラス定義（スキーマ）さえ決定てしまえば、外部のプログラム開発はたいへんスムーズに行えた。スキーマの設計はかなり困難を要した。スキーマの変更は一般的に柔軟には行えない。VersantではインスタンスがDBに登録されている場合はスキーマの変更は事实上困難である。

(3) データの登録

スキーマが完成したらデータをオブジェクト化してDBへの登録を行う。C++ではオブジェクトを作る際にコンストラクタを呼び出す。コンストラクタでは普通の関数同様、引数を受け取ることができる。開発を始めた当初の構想では、コンストラクタに対していくつかの引数を指定することでオブジェクト自体が自動的に正しいオブジェクト間の関係を確立するということを考えていた。例えば建物オブジェクトであれば、住所と建物名と座標列を与えると住所から属する行政単位を、座標から属するメッシュを探しだし、それらを利用して対応するサブメッシュを見つけて自分を登録する、といったイメージである。ところが、今回のような複雑なクラス階層、オブジェクト間関係がある場合にはこの方法は不可能ではないにしろ大きな困難を伴うことがわかった。まず祖先のクラスのコンストラクタと子孫のクラスのコンストラクタの整合性の問題。もう一つはリンク関係や座標などの引数の増大である。またそれに伴ってコンストラクタ自身の肥大や、コンストラクタ内部でのエラー発生可能性も上がる。つまり、ある程度以上複雑なクラス階層、オブジェクト間関係の一貫性をサポートするにはコンストラクタという機構だけでは見通しのよい設計は行えないのではないかだろうか。最終的に今回の試作ではコンストラクタには引数を渡していない。コンストラクタの仕事は最低限の内部変数の初期化のみとし、残りの処理はメソッドとして用意しておき、全て外部プログラムに任せた。この手法ではオブジェクト作成時に正しい状態が確立されず、オブジェクト内の変数の設定し忘れなどが起こり得るが、複雑なコンストラクタの作成に時間を割くよりは効率がよいと考えたからである。

6. おわりに

システムは稼働しているが、広範囲の検索では応答が遅いので、純粹なオブジェクト指向の実装ではなく、チューンナップが必要である。

本プロジェクトは北九州テクノセンターから支援を受けて、平成3年、4年の2年間に亘って、九州工業大学情報工学部打波研究室、ゼンリン、ASAシステムズ、YCC、北九州情報ひろばの五者でチームを組み開発を行なったものであり、ここに記して深く感謝致します。

プロジェクトの詳細は報告書を参照されたい。
各部の詳細は別の機会に述べたい。

参考文献

- 1) Versant Object Technology : Versant System Reference Manual. (1992)
- 2) Versant Object Technology : Versant C++ Interface Manual. (1992)
- 3) Bjarne Stroustrup: THE C++ PROGRAMMING LANGUAGE, Addison Wesley(1991)
- 4) B. J. コックス: オブジェクト指向のプログラミング, トッパン(1988)
- 5) R. S. ウイナー, L. J. ピンソン: C++ ワークブック, トッパン(1991)
- 6) W. リチャード, スティーヴンス: UN I X ネットワークプログラミング, トッパン(1990)
- 7) 江河, 打浪, 堀川, 石井: 地図遠隔地アクセスデータベースシステムの開発 北九州テクノセンター産学官共同研究成果報告書 92年6月
- 8) 打浪, 尾崎, 山下, 石井: 地図遠隔地アクセスデータベースシステムの開発 北九州テクノセンター産学官共同研究成果報告書 93年6月
- 9) 荒木信吾: 地図データベースシステムの構築法に関する研究, 九州工業大学情報工学部特別研究報告書, 93年3月

付録 データベースクラス階層

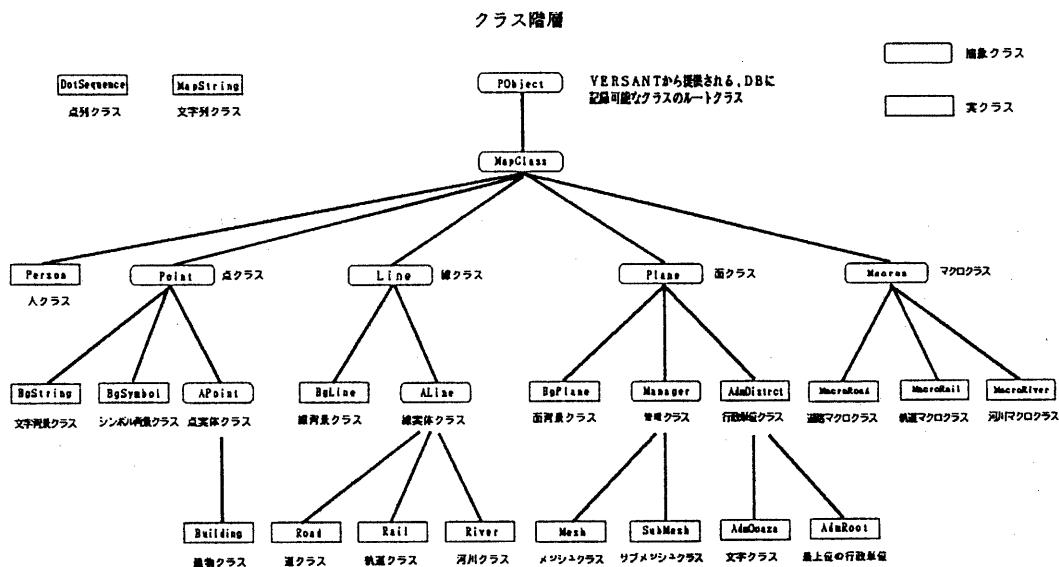


図 住宅地図クラス階層