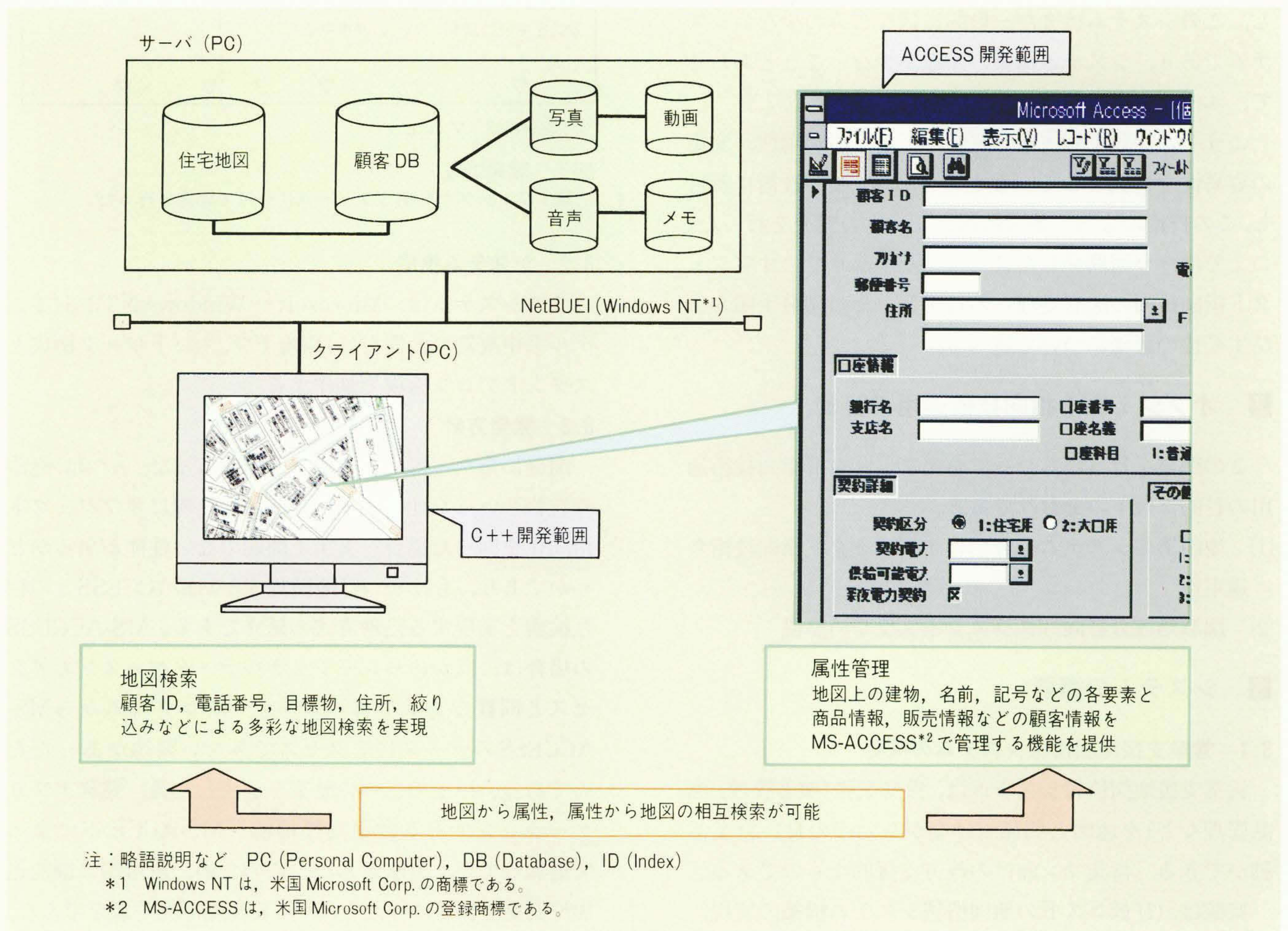


営業支援地図情報システム“HMAP/ET”

Object-Oriented System for Business Market Mapping System

温井和彦* Kazuhiko Nukui 斎藤 浩* Hiroshi Saitō
池田 務** Tsutomu Ikeda



営業支援地図情報システム

このシステムは、市販の住宅地図 (CD-ROM : Compact Disc Read-Only Memory) を利用し、顧客管理を主体とした営業支援地図情報システムが簡単に構築できるものである。

このたび、手軽に情報地図が扱えることをコンセプトに、パソコン (パーソナルコンピュータ) 上で動作するデータベースと地図処理を組み合わせたパソコン用営業支援地図情報システム “HMAP/ET” (Hitachi Mapping Application Support System/Enhanced Territorial Mapping Management System) を開発した。

オブジェクト指向開発技法を積極的に取り入れることにより、途中からの仕様変更および機能追加に対し柔軟に対応できる。また、データ管理の簡略化と先読み処理の並列化の併用により、従来のワークステーション用地図システムに比べて性能レベルは同等以上である。

* 日立製作所 情報システム事業部 ** 株式会社日立エンジニアリングサービス 情報システム部

1 はじめに

地図情報システムは、地図を利用する業務を効率化する目的で、または、シミュレーション処理など高度な業務処理を行う目的でシステム化されることが多い。しかし、このシステムはまだ一般的にはなじみの少ないシステムであり、システム構築の知識が少ないこととあわせて、ユーザーへの普及が遅れていた。

こうした問題を解決する試みとして、再利用性、変更の容易性を特徴としているオブジェクト指向技術に着目し、この技術によって地図情報システムの開発を行った。ここでは、今回開発したクラスライブラリでのオブジェクト指向技術の図形処理への適用と、その開発手順および生産性の評価について述べる。

2 オブジェクト指向技術適用の目的

この開発を行うにあたってのオブジェクト指向技術適用の目的は以下のとおりである。

- (1) 地図情報システム分野でのオブジェクト指向技術の適用性
- (2) 図形処理分野向けクラスライブラリの整備

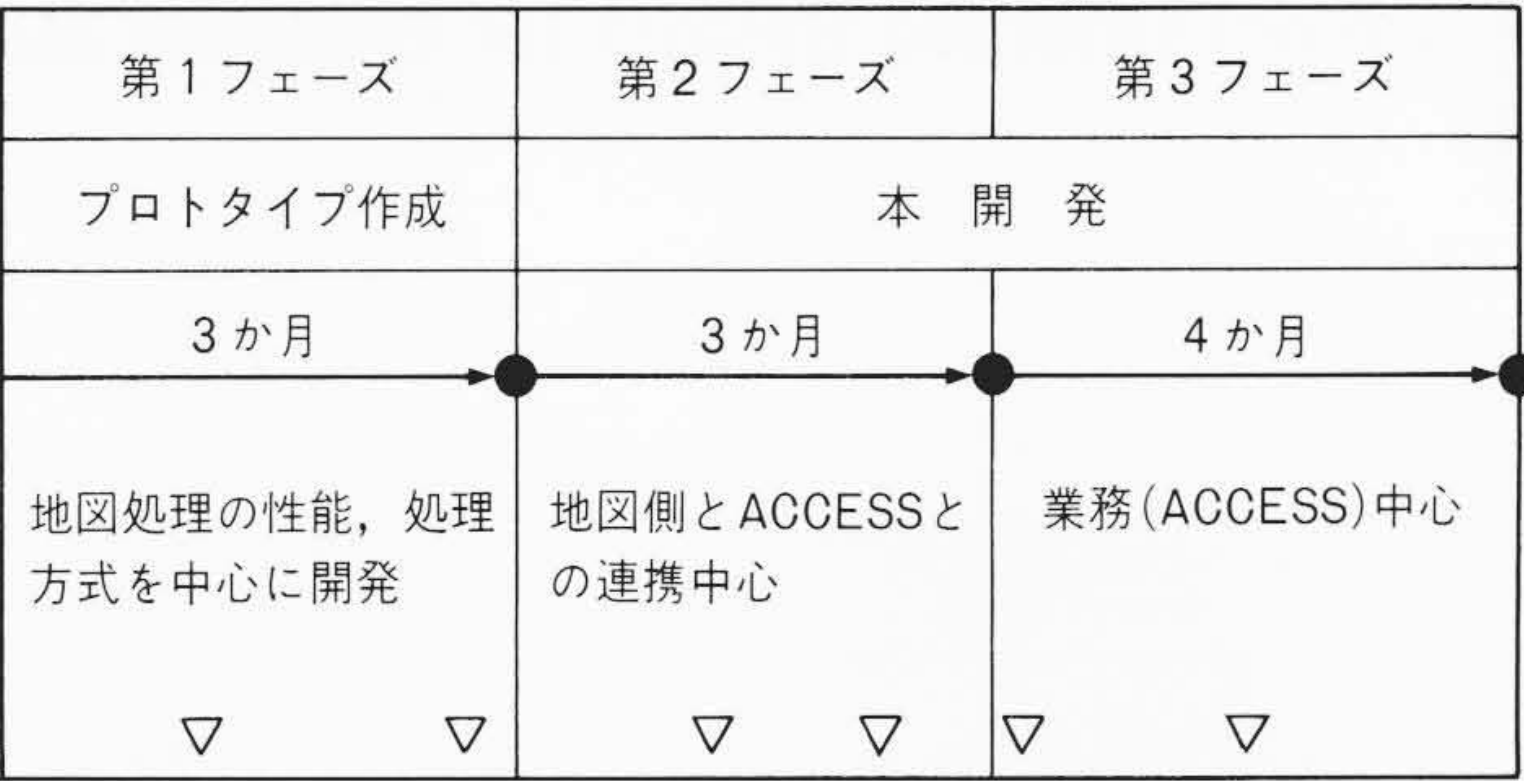
3 システムの概要

3.1 営業支援地図情報システムの概要

営業支援地図情報システムは、営業業務(顧客管理、商品管理など)を地図と関係づけながら効率の良い営業活動ができる、営業マン向けの後方支援的なものである。

特徴は、(1) 低コストの地図情報システム構築の実現、およびだれもが手軽に扱える地図システムのコンセプトの下に、地図処理機能と属性との連携部は標準機能として提供し、(2) 営業情報の処理部にMS-ACCESSを採用することにより、パソコンの知識を持つ人であれば、カスタマイズも可能となるように、マクロ処理およびACCESS BASICのコーディング部を開示するようにしたことである。

さらに、CD-ROMで市販されている電子地図(住宅地図)を利用するユーザーが自動的に属性情報、住所情報、目標物情報を地図上の位置情報と合わせて取り込むことができる機能もあわせて開発(別途変換ツールが必要)し、手間の掛かるリンク作業を行わずに済むようにした。これにより、従来比で $\frac{1}{5}$ から $\frac{1}{10}$ 程度の作業で情報地図のシステム構築ができるようになった。



注：記号説明 ▽ (レビュー)

図1 開発工程
第1フェーズから第3フェーズに分けて開発を行った。

3.2 システム構成

このシステムは、Microsoft^{※1)}Windows NT3.5(インテル系中央処理装置だけ)環境下で、ネットワーク構成とスタンドアロン構成で動作する。

3.3 開発方針

開発にあたって二つの懸案事項があった。一つは、道路や建物といった地物を取り扱う地図処理にオブジェクト指向化を図った場合、実用上問題のない性能が出るかどうかであり、もう一つは地図処理とMS-ACCESSとの相互検索を実現する処理方式の確立である。MS-ACCESSの場合は、ほかのリレーショナルデータベースへのアクセスと同様な方式では、ユーザープログラムからMS-ACCESSのデータにアクセスできない問題があったためである。以上の二つの懸案を解決した後、業務アプリケーションである営業支援機能をMS-ACCESSによって追加していく方法とした。このため、段階的に機能追加や変更が容易に行えるスパイラル型開発手順を選んだ。

3.4 開発日程

スパイラル型の開発手順を実施するにあたり、それぞれの開発フェーズの各イベントで、関連部門とのレビュー(主にオブジェクト指向技術利用の観点)と設計内での性能評価を行いながら開発を進めた(図1参照)。

4 開発環境

開発環境は、パソコンのネットワーク環境下(NetWare^{※2)}3.12J)で開発を進めた。ただし、Windows^{※3)}のOS(Operating System)の流れが16ビット(Microsoft

※1) Microsoftは、米国Microsoft Corp.の登録商標である。
※2) NetWareは、米国Novell, Inc.の登録商標である。
※3) Windowsは、米国Microsoft Corp.の登録商標である。

Windows3.1)から32ビット(Microsoft Windows NT 3.5)へと移り変わろうとする時期でもあり、ターゲットOSをMicrosoft Windows NT3.5と決めたが、32ビットAPI(Application Programming Interface)向け開発言語のVisual C++^{※4)}2.0やMicrosoft Windows NT3.5用MS-ACCESS 2.0がリリースされていない状況下での開発スタートとなり、次の環境で開発を進めた。

開発言語では、地図表示アプリケーション部はVisual C++1.0を、営業支援処理(データベース)部はMS-ACCESS1.1を使用した。

Visual C++1.0にはMFC(Microsoft Foundation Class Library)2.0が添付されており、これをできるかぎり利用した。

また、地図アプリケーションとMS-ACCESSとの連携処理にはMicrosoft Windowsのメッセージシステムを利用して、互いに通信を行いながら動作する方式を確立した。

なお、現在開発中の次期バージョンは、Visual C++ 2.0(MFC3.0)およびMS-ACCESS2.0対応製品とする予定である。

5 地図処理クラスライブラリの構築

5.1 性能に対する考慮

地図処理システムの場合、初期表示、スクロール、拡大・縮小などの性能が評価の大きなウェイトを占める。このため、オブジェクト指向化にあたっては、第一に各地図処理基本機能の性能を見極めることにした。

※4) Visual C++は、米国Microsoft Corp.の商標である。

(1) MFCの問題点

オブジェクト指向プログラミングの成功は、採用した開発言語に付属するクラスライブラリや既存クラスライブラリをどこまで使いこなすかにかかっている。

この開発では、MFCを採用したが、MFC自体はドキュメントビュー(Document-View)構造であり、ドキュメントすなわちキャラクターベースのシステムには最適のクラスライブラリではあるが、地図処理のような図形データを描画するシステムでは、そのほとんどを自前で設計する必要があった。

(2) 地図処理方式のオブジェクト指向化

描画処理上、従来の処理方式との違いは、各要素(オブジェクト)自身が自分自身を地図平面上のどこに描画するかを知っており、地図の描画はすべて描画オブジェクト(CView)に任されることにある。従来の方式では、表示する範囲と表示する際の線種、線色、線幅などを指定して描画処理に移る。オブジェクト指向化の場合、各要素自身で表示範囲に入っているか、線分や円などの種別の認識、表示する際の線種、線色、線幅を自身が認識し表示することになる。この分オーバーヘッドが大きくなる。

このため、描画速度を高速化するために地形図(ファイル)に対応した描画オブジェクト群をキャッシュクラスとして作成しておき、オブジェクトの生成(NEW)、消滅(DELETE)のオーバーヘッドを最小限に抑えるようにした。さらに、このオブジェクト群を管理するコレクションクラスを設け、複雑に相互関係を持つ地図要素群を効率よく管理することにした(図2参照)。

コレクションクラスは、それぞれの図に対応する複数のキャッシュクラスを配列として管理し、地図要素をオ

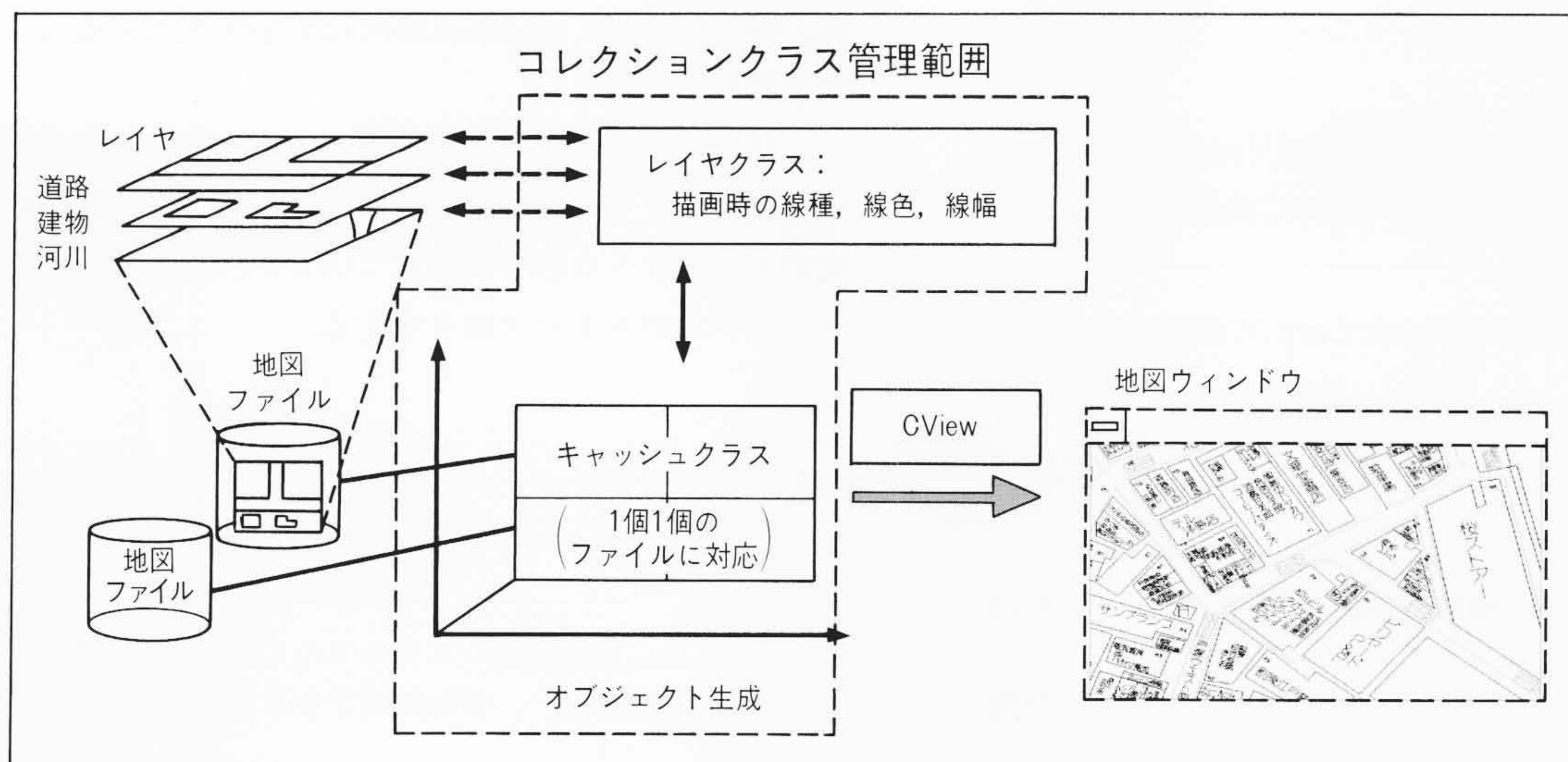


図2 データ処理概念
隣接関係を持つ地図データをキャッシュクラスに取り込み、その後のオブジェクトの管理をコレクションクラスで行うようにした。

表1 地図アプリケーション部の開発規模とクラス数
開発規模をC換算の $\frac{1}{2}$ とすることができた。

C++開発規模	C見積規模	クラス数
23.7キロステップ	52キロステップ	49

ブジェクト化したオブジェクトリストをセットする。ここで、ヒット率を高めるために配列の先頭は最も最近に使用された(ディスプレイに表示された)オブジェクトリストをセットするようにする。

また、配列の最後には最も過去に使用されたオブジェクトリストを置き、新しい地図データを読み込んだときにはこの配列の最後のオブジェクトリストを破棄する(DELETE)ようにした。

この方式により、従来の処理方式に劣らない地図描画方式を実現することができた。

6 MFC利用性の確認

MFCはドキュメントビュー構造であり、今回の開発でモデルビュー(Model-View)構造に作り替えたと前述したが、ここを除くと実質的にはMFCの効用は大きなものがある。例えば、極端な例をあげると、地図をディスプレイに描画する際のウィンドウ定義処理を従来のMS-SDK^{※5)}(Microsoft Windows Software Development Kit)による記述と比較すると、 $\frac{1}{50}$ から $\frac{1}{100}$ 程度のステップ数で済む。新たに作成したクラスも地図という特殊な処理業務であるために、共通的な処理ではMFCの積極的な利用を進めることで開発規模を大きく削減することができた。地図アプリケーション部の開発規模とクラス数を表1に示す。

7 適用効果

7.1 生産性

営業支援地図システムの開発規模はVisual C++の実績で23.7キロステップとなり、従来(C換算)と比較して

※5) MS-SDKは、米国Microsoft Corp.の商標である。

$\frac{1}{2}$ であった。

開発規模が小さくなるのはオブジェクト指向化だけでなく、Visual C++の記述法にも起因するので、単純な規模比較でなく、実現機能の内容によって評価していくべきであると思われる。

今回の地図処理部分のオブジェクト化では、地図の描画機能やスクロール、拡大・縮小などの再利用できるMFCクラスがなく自前で構築したので、従来(C見積もり)と比較して飛躍的に生産性を向上させることはできなかった。今後は、今回開発したクラスライブラリの活用とさらに充実を図ることにより、生産性向上につなげることができると確信する。

7.2 拡張性

製品開発のスパンが非常に短縮されている現在、オブジェクト指向技術の必要性がますます高まってくると思われる。

実際、このシステム開発でも途中での機能追加や仕様変更は製品検査直前まで行っていた。これだけパソコン環境が劇的に変化している現在、計画時の仕様どおりに作るとは、ある意味では時代遅れとなってしまう可能性がある。システムの開発途中でも柔軟に仕様変更・機能追加が容易にできる仕掛けが必要である。

オブジェクト指向技術は、この意味でも今回のシステム開発に力を発揮したと考える。

8 おわりに

このシステムの開発を通してオブジェクト指向技術を適用したスパイラル型開発手順では、地図のような図形処理業務のアプリケーションパッケージ開発の観点からも有効であり、アプリケーションパッケージとしての品質、特に拡張性・保守性に優れた製品を開発することができたと思われる。

これからは、オブジェクト指向技術を駆使できる人材の育成と、クラスライブラリその他システムへの積極的な流用をどのような形で実施していくか、このインフラストラクチャの充実が大切と考える。

参考文献

1) 広瀬, 外: システムの構築を容易にするオブジェクト指向開発環境, 日立評論, 77, 5, 345~348(平7-5)

2) 秋庭, 外: 病院情報システムへのオブジェクト指向技術の適用(1)-開発手順-, 情報処理学会第50回全国大会(1995)

3) 筒井, 外: 病院情報システムへのオブジェクト指向技術の適用(2)-評価-, 情報処理学会第50回全国大会(1995)