地理情報システムのためのフレームワーク

6U－5

上野和彦　伊藤禎康　藤本憲司
ＮＴＴコミニュケーションウェア株式会社

1. はじめに

地理情報システム(GIS)は、地理情報(道路や建物)を地図上に表示したり、ユーザ情報を地図と連携させて、システムに登録・管理するために利用される。ところが、扱う地理情報は、利用目的に応じて異なる。また、その表現方法についても一様ではない。地理情報を解析して得た結果を表示するような機能も利用目的毎に千差万別である。さらに、既存の地理情報に対しての情報種類の追加やそれに伴う機能の追加等、様々な利用者要求も発生する。このような様々な要望に対して柔軟かつ容易に対応するために、地理情報システムにフレームワークを導入する。その結果、様々な表現方法やユーザ情報管理は、コンポーネントの作成・追加によって行うことができる。

本論では、主に施設管理システムや地図上のポイントデータ(店や観光地などの情報)を扱うシステムのために試作したフレームワークについて述べる。

2. 地理情報システムのフレームワーク

GISに必要となる基本的な機能には、地理情報に対して、(1)地図の表示、拡大、縮小、スクロール (2)レイヤ(道路情報、建物情報などそれぞれのデータ)ごとの表示／非表示切り替え (3)建物などの属性の表示 (4)検索 がある。また、地理情報以外を扱う機能として、(5)ユーザ情報の登録、参照、検索 (6)情報の解析と結果表示 がある。こういった基本機能を持ったシステムにおいて、各システムごとに異なったり、変更の可能性があるものとして (a)地図表現方法の相違 (b)ユーザ登録情報の相違 (c)情報の解析方法の相違 (d)レイヤ追加、変更 (e)ユーザインタフェースの追加、変更 (f)情報サーバの相違 が考えられる。したがって、GISのフレームワークでは、上記の(1)～(6)の基本機能を備えつつ、(a)～(f)の変更に対応できることが必要となる。

Framework for Geographic Information System
Kazuhiko UENO, Sadayasu ITO, Kenji FUJIMOTO
NTT Communicationware Corporation

3. 試作システムの概要

3.1 システム構成

本稿のフレームワークは、地図サーバやユーザ情報をバックエンドに持ったGISのクライアント部分を対象としており、Javaで実装した(図1)。また、適用範囲として、主に施設管理システムや地図上のポイントデータを扱うシステムを対象にしている。したがって、前節で述べた基本機能(1)～(4)、および、地図上の(面でなく)点に置かれるユーザ情報の登録、参照、検索を行うシステムとなる。よって、情報の解析などは対象としない。これら機能をもったGISにおいて、地図表現方法やユーザ登録情報やレイヤの追加・変更やサーバの変更(上記a, b, d, f)が、ある程度容易になる。

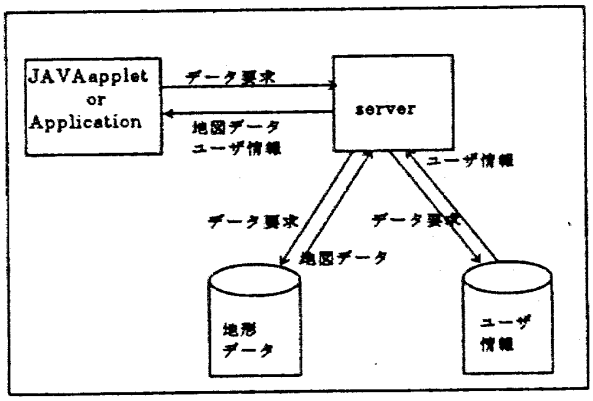

図 1　構成図

3.2 フレームワークの設計

このフレームワークは、大きく(1)表示部、(2)データ管理部、(3)データアクセス部の３つに分かれている(図2)。表示部は地図の表示や、マウスイベントの取得をする。データ管理部は、キャッシュのような役割をもち、要求されたデータがメモリ上に無い場合、データアクセス部を通して、サーバからデータを読み込む。ここで、システムの利用形態に合わせて、実際にロードする範囲を変えたり、先読みを行ったりすることもできる。また、その判断をするクラスを交換可能にしてい

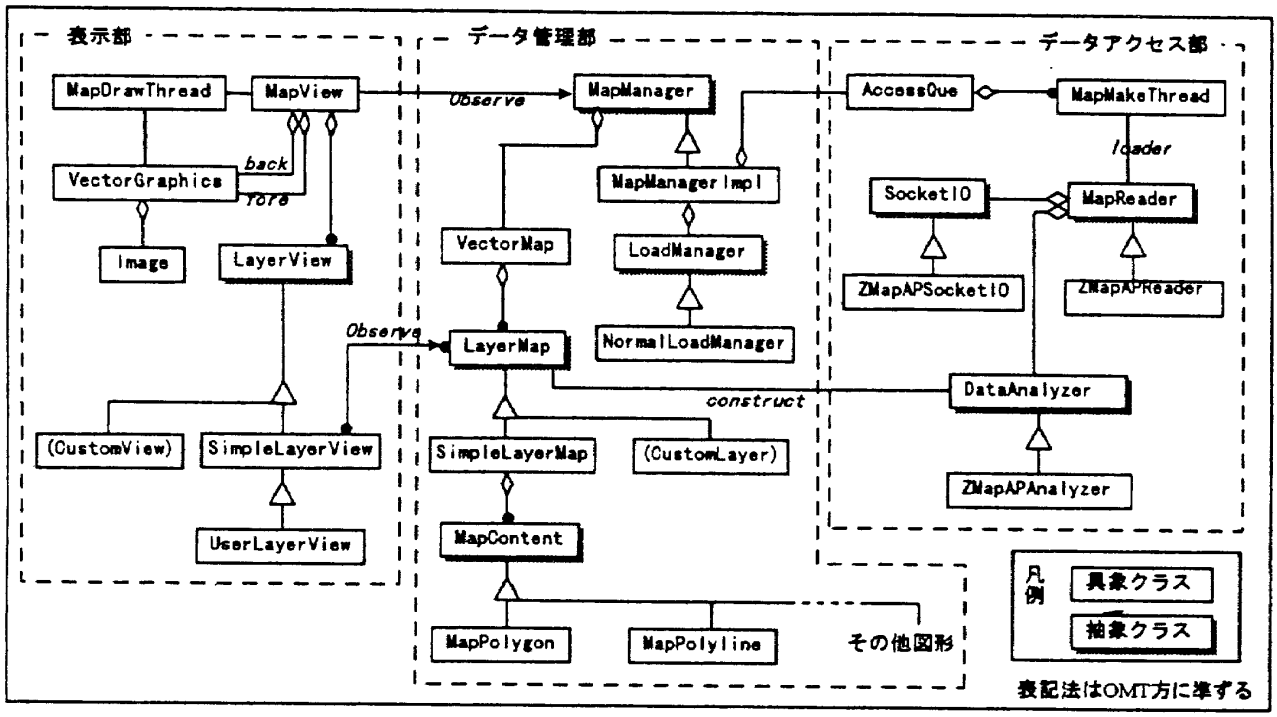

図 2 オブジェクトモデル図(一部)

る。データアクセス部は、要求に応じてサーバへアクセスし、データを読み込んだり、ユーザ情報を読み書きしたりする。

本システムでは、同一情報に対しての地図表現方法を変更可能にするため、表示部のレイヤ表示オブジェクトを抽象化(LayerView)した。ここで、単にベクトルデータをそのまま表示する場合は、LayerViewを継承したSimpleLayerViewを使う。それ以外の表現方法が必要ならば、LayerView(または、SimpleLayerView)を継承した新たなオブジェクト(CustomView)を追加することで、表現の変更が可能となる。

また、ユーザ登録情報の追加やレイヤの追加を可能にするために、レイヤ情報を持つオブジェクトを抽象化(LayerMap)した。地形図のような線や点の組み合わせのベクトルデータのみの場合は、LayerMapを継承したSimpleLayerMapを使うが、その他の構造の情報が必要であれば、LayerMap(または、SimpleLayuerMap)を継承した新たなオブジェクト(CustomLayer)を追加することでレイヤを追加できる。

データアクセス部は、３つの抽象化されたオブジェクト(SocketIO,MapReader,DataAnalyzer)があり、サーバのアクセス方法やデータフォーマットに応じたオブジェクトをこれら３つの抽象クラスから継承して作成する。図２の“ZMapAP～”のオブジェクトがそれに相当する。ここに新しいオブジェクトを追加すれば、サーバやデータフォーマットの相違に対応できる。

新しいオブジェクトの追加や既存オブジェクトからの変更をコードと分離するために、交換可能なオブジェクトを生成するオブジェクト名や使用するレイヤといったようなオブジェクト名を、初期設定ファイルに指定できるようにした。

また、設計にデザインパターン[1]を利用することで、こういった追加、変更時の影響範囲が小さくなるようにしている。

4. おわりに

本稿では、主に施設管理システムや地図上のポイントデータを扱うシステムの基盤となるフレームワークに付いて述べた。今後、対象領域を広げた場合の考察や評価し、いくつかのアプリケーションの作成を通しての有効性評価を行う。

参考文献

[1] E.Gamma, R.Helm., R.Johnson and J.Vlissides, “Design Patterns - Elements of Reusable Object-Oriented Software ”, Addison-Wesley, 1994