

TR-IT-0160

オブジェクト指向による地図情報の表現

深山 修 ローケンキム キュンホ

Osamu FUKAYAMA Kyungho LOKEN_KIM

1996.3.15

最近、オブジェクト指向の考え方が計算機の世界で、大きく話題になってる。
しかし、オブジェクト指向には標準的規定があるわけでもなく、その実態や応用についてはなかなか捉えにくいところがある。そこで本稿ではまず、オブジェクト指向について共通認識されている基本的概念について述べる。つぎにオブジェクト指向の考え方が地図データベースの構成に有効であることを述べ、最後に今回作成した地図データベースのシステム開発の方法について述べる。

ATR音声翻訳通信研究所

ATR Interpreting Telecommunications Research Laboratories

© (株) エイ・ティー・アール音声翻訳通信研究所

© 1996 ATR Interpreting Telecommunications Research Laboratories

目次

1.はじめに	1
2.オブジェクト指向	2
2.1 オブジェクト指向の基礎概念	3
2.2 継承・関連	6
2.2.1 継承	6
2.2.2 関連	6
2.2.3 継承と関連	7
3.オブジェクト指向による地図情報データベースシステム	8
3.1 オブジェクト指向データベース	8
3.2 地図情報データベース	10
3.3 地図データベースの作成	11
3.3.1 分析	12
3.3.2 データの入力	13
4.むすび	15

1.はじめに

近年計算機の世界で、大きく話題になっているものの1つに、オブジェクト指向の考え方がある。この、オブジェクト指向というプログラムにおける新しい考え方は、単にプログラム言語の仕様としてのみならず、プログラミングや設計のための方法論として、また人工知能（AI）分野におけるアプリケーション技術や事象の分析のための技術として大きな注目を集めている。

このように、オブジェクト指向が高い評価を受け、いろいろと話題になっているのは、この考え方が コンピュータサイエンスの世界では最も新しく、そして現在のところ人間にとって最も馴染みやすい、つまりプログラミングの技法というよりも、人間の言語活動と等価な人間の指向表現のための技法、すなわち「思考を表現するためのツール」であるからです。

本報告ではまず、オブジェクト指向の基本概念について簡単に説明し、次にオブジェクト指向の考え方が地図情報データベースシステムに有効であることを述べ、最後に今回作成した地図情報データベースのシステム開発の方法について述べる。

2.オブジェクト指向

過去10年間、オブジェクト指向技術はプログラミング言語、ユーザインターフェース、データベース、OS、エキスパートシステムなどの分野で発展してきた。

しかし、話題になっているにもかかわらず、その実態や応用についてはなかなか捉えにくいところがある。それは、オブジェクト指向がプログラム言語の基礎研究からのみではなく、情報処理として包含される様々な応用分野からいわば同時発生的に提起されたために（図1）、その起源や背景を統一的に把握することが困難であったということが指摘できる。そこで、この章では、オブジェクト指向の基本概念について述べる。

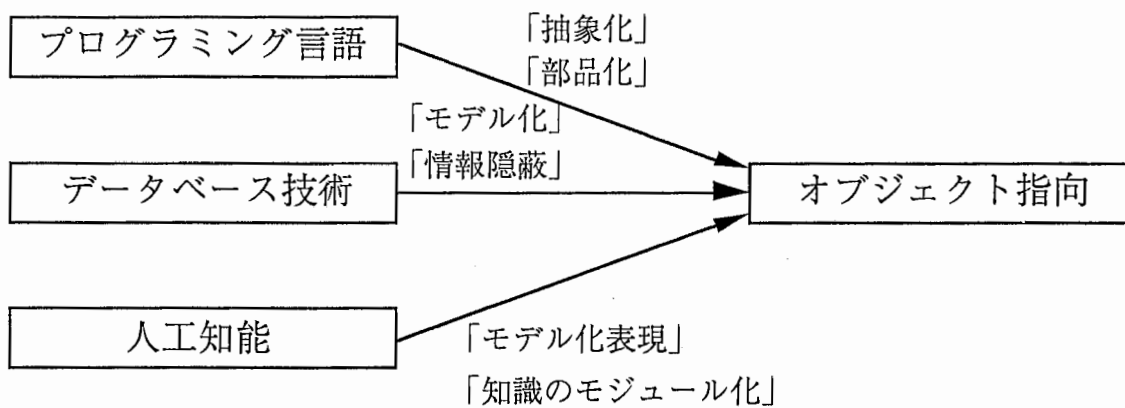


図1 オブジェクト指向への流れ

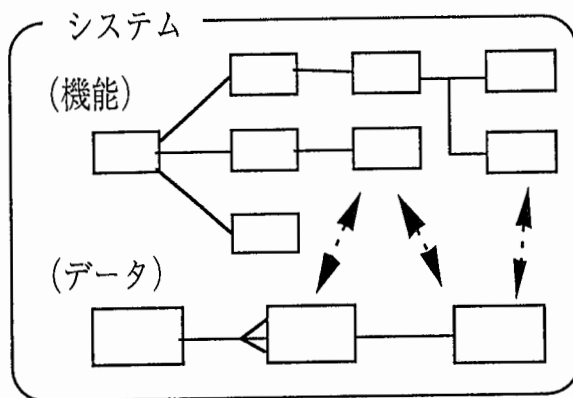
2.1 オブジェクト指向

オブジェクト指向は、システムをデータ構造と振る舞いが一体となったオブジェクト単位に分割し、その複数のオブジェクト同士が相互に協調動作することでシステム全体の役割を果たす。よって、従来の方法が、機能中心かデータ中心あるいは、両者を切り離していたのに対し、オブジェクト指向ではデータと機能が一体となったオブジェクトを中心にシステムを構成する。(図2)

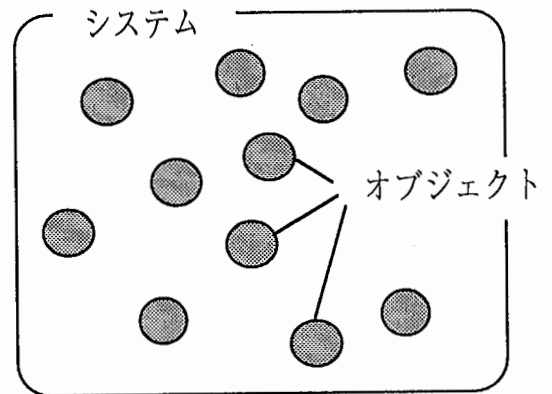
オブジェクト指向によるアプローチには次のようなものがある。

— オブジェクト指向によるアプローチ —

- ・機能中心からオブジェクト中心へ
- ・一貫したオブジェクトモデルの使用
- ・クラスによる強力な抽象化技法
- ・データ隠蔽による変更や保守の容易性
- ・継承による再利用可能性・拡張性の向上



(a) 従来のシステム



(b) オブジェクト指向

図2 システム構成

オブジェクトとは実世界の対象物がある視点から見た「もの」で、すでに述べたように性質を表わす共通なデータ構造（属性）と振る舞い（操作、メソッド）から構成される（図3）。この属性と操作は同じ「もの」でも、それを見る視点により変わってくる。例えば、同じりんごを見ても商品として見たときの属性・操作と食べ物として見た属性・操作は変わってくる。よってシステム開発をする場合、オブジェクトをその目的にあった的確な視点から眺め属性・操作を決めなければならない。

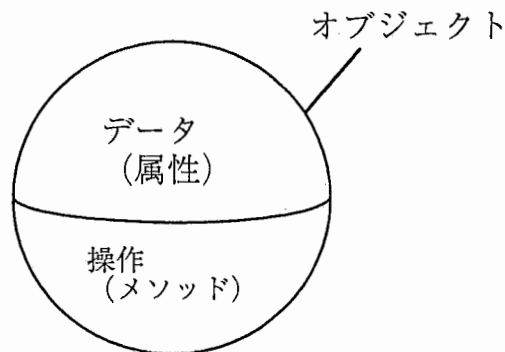


図3. オブジェクト

また、同じ属性とメソッドをもっているすべてのオブジェクトの集まりをクラスという。オブジェクトはこのクラスにより生成される。

(用語)

- ・クラス : オブジェクトの型を定義するもの
- ・インスタンス : クラスにより定義された実体
- ・オブジェクト : インスタンスの総称または個々のインスタンス
- ・メッセージ : オブジェクトの操作を行う手続き

よく「オブジェクトはデータとプログラムを『カプセル化する』」という表現がされる。これはユーザーはオブジェクトの内部を直接操作することはできず、オブジェクトの一部であるプログラムを呼び出すことによるのみ可能ということの意味する。このオブジェクトの操作を行うことを「オブジェクトにメッセージを送る」と表現する。よってオブジェクトはこのメッセージによるのみ操作される。このような性質を「情報隠蔽」といい、オブジェクト指向における重要な性質の1つとなる。

カプセル化の特徴づけ

- ・オブジェクトの内部ソフトウェアすべての有効範囲を定める
 明白な境界
- ・オブジェクトが他のオブジェクトとどのように相互作用するかを記述するインターフェース
- ・ソフトウェア・オブジェクトの機能を詳細する保護された内部表現。その実現の詳細は、そのオブジェクトを定義するクラスの外からはアクセスできない

2.2 継承・関連

2.2.1 継承

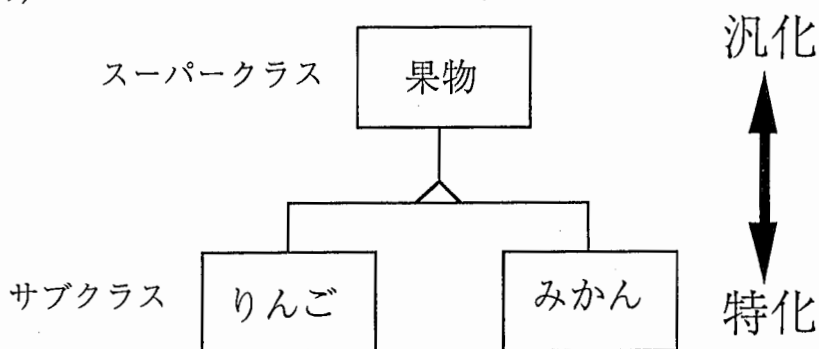
継承は「再利用」とも呼ばれる。継承とは既にあるクラスを拡張して新しいクラスを作ることでできる機能のことである。

継承を行うパターンには以下の2つが挙げられる。

1. 新しく設計しようとするクラスに、似たようなクラスが既にある場合、その属性や操作を追加して新たなクラスを設計する（特化）。
2. 新しく設計しようとする複数のクラスの共通点を抜き出して一つのクラスを作り、そこから継承する（汎化）。

また、1クラス上位のクラスをスーパークラス、1クラス下位のクラスをサブクラスという。サブクラスのオブジェクトはスーパークラスの性質を受け継ぐが、スーパークラスにはない特製を有することができる。すなわち「サブクラスはスーパークラスである」といえる。

例)

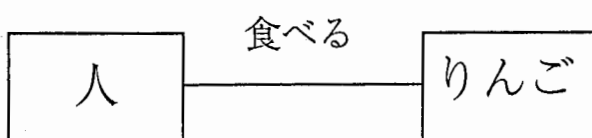


2.2.2 関連

複数のオブジェクトが協調動作するとき、それらのオブジェクトには何らかの関係が成立します。

このクラスとクラスの間を関連といいます。

例)



2.2.3 継承と関連

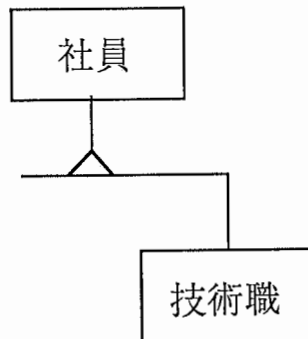
オブジェクト指向によるシステムの分析・設計では、モデルの特徴によって、継承と関連を使い分けることが必要となる。

(関連)



「社員は技術職という職業をもつ」

(継承)



「技術職は社員的一种である」

3.オブジェクト指向による地図データベースの作成

3.1.オブジェクト指向データベース

これまでデータベース管理システムは関係データベース (RDB) が数学的基礎付け、簡明さ等により高い評価を受けてきた。しかしデータベースが「事務処理」分野から技術分野、生産分野など複雑なデータ構造とアルゴリズムを要する応用ではうまくいかない点が見え始めた。それはRDBがデータ構造が単純すぎることで、データ操作が十分強力でないことが大きな原因となっている。そこで、このようなRDBの限界を克服するため演えきデータモデル、意味データモデルなどの新しいデータモデルが考案されているが、その中でも次世代データベースシステムの最有力候補であるのがオブジェクト指向データベース (Object-Oriented DB: OODB) といわれている。論文[1]によればオブジェクト指向データベースが次世代データベースの最有力である理由として以下の2つを挙げている。一つは、他のモデルもオブジェクト指向パラダイムの影響を受け、限定的だが数々のオブジェクト指向概念を取り入れ始めていること。第二に、多数のOODBMSが開発されていることである。このことは OODBが次世代データベースシステムとして世に受け入れられることに特に大きな役割を果たす。

また、論文[1]によるとOODBにはRDBに比べ次のような利点があると考えられている。

(1) 複合オブジェクトの表現が容易

複合オブジェクトとは、複数のオブジェクトから構成されるオブジェクトのことで、これはプログラミング言語でデータタイプの定義に既に定義されたデータタイプを利用できることに類似する。この機能により、データのモデリング時に高い抽象化が可能になる。

(2) データと捜査のこん包

プログラミング言語におけるデータ抽象化とは、データ構造とそれに作用する操作アルゴリズムとを外部から隠蔽し、操作インターフェースのみを見せようという考え方である。このデータと操作とのこん包はOODBではクラス定義により記述される。この考え方はRDBにおける「データ独立」とは異なるアプローチでのデータとプログラムの独立性を保証する。

(3) 汎化階層とプログラム部品の再利用

下位クラスは上位クラスの属性とメソッドとを継承する。似たようなクラスをたくさん定義しなければならない応用では、既存のクラス階層に新しいクラスを追加することによって既存のメソッドを上手に再利用できる可能性がある。メソッドはプログラム部品であるから、プログラムコードの再利用が実現できる。

(4) プログラムの管理

(2) と (3) とにより、データばかりでなく、応用プログラムの管理をもデータベースシステムが可能にしてくれる。この「プログラムの管理」という概念はRDBにはないものである。

以上のようにRDBに比べOODBには利点があるため、可能性を秘めた、次世代データベースとして期待されている。

3.2 地図情報データベース

OODBの応用分野を特徴づけるのは「処理データの複雑性」と応用システムの「動作環境の複雑性」である。現在の計算機環境は（高速）通信回線で結ばれた多数の高機能計算機のネットワークである。このような環境で、複雑な処理を行う応用で高速性を要求するものにはOODBが向いているといわれる。

処理対象のデータ構造が複雑で、処理アルゴリズムも複雑な応用分野には、CAD、画像処理システム、そして地理情報システム等がある。

オブジェクト指向に基づく地図データベースシステムの構成はすでに平成3、4年の2年にわたり、九州工業大学とゼンリン（株）が中心となって開発を行っている。このシステムでは、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、自由な検索を可能とするようにOODBによりデータベース化している[2]。

3.3 地図データベースの作成

オブジェクト指向によるシステム開発は、データと機能を一体化したオブジェクトの構造すなわち、オブジェクトモデルを中心に行う。

分析段階では実世界のシステムを深く理解し、明確にモデルに反映することを目的とする。設計段階では分析段階でのモデルをコンピュータ上に実装するためにモデルの詳細化、最適化を行う。

— オブジェクト指向分析・設計の特徴 —

- ・ 開発サイクルにおいて分析に多くの時間を費やす
- ・ 機能よりもオブジェクト（データ）を中心とする
- ・ 分析から設計まで一貫したオブジェクトモデルを使用する。
- ・ 分析・設計を繰り返し行う

今回、蹴上駅周辺の地図データベースの作成を行った。システムの機能としては、コンピュータ画面上に表示されている地図上のオブジェクトをマウスでクリックすると、その建物や道などの情報が画面に表示されるようする。地図情報は市販されている地図の中から蹴上駅周辺が詳しく記載されているものを選び、その地図をスキャナーによりコンピュータに取り込み、地図に記載されている範囲の情報で入力を行う。

3.3.1 分析

分析段階での主な流れを図6に示す。

(1) オブジェクトの識別

まず、地図に記載されている情報の中からオブジェクトを見付け出す。

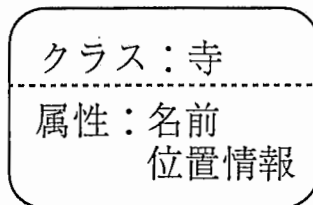
例) 蹴上駅、国際交流会館、鴨川、岡崎動物園、平安神宮、二条通、左京区、
京都ホテル、山王町、丸太町橋、三和銀行、左京税務署、栗田小学校、南禅寺 等

(2) クラスの識別

次にクラスの識別を行う。例えば、本妙寺、要法寺、明照寺、大恩寺 などは
”寺” というクラスにまとめ、属性として寺の名前をもたす。

クラス例) 寺、道、川、小学校、駅、幼稚園銀行、橋、町名、神社、デパート
高校、短期大学、会館、ホテル 等

また、これらのクラスは地図情報としてそれらの存在する位置を識別できなければ
ならないので、属性として位置情報をもたす。



(3) 次にこれらのクラスを継承、関連などを用いてクラス図にする。

今回作成した地図データベースのクラス図を図7～11に示す。

名前や位置情報は各々のクラスでもっている属性なので、Map Objectというクラス
に継承させた。

3.3.2 データの入力

データの入力はまず対象とするオブジェクトのクラスを入力し、次にその指定した位置にあるオブジェクトの名前など、属性の入力を行う。そして位置情報をスキャナーによって取り込んだ画面上の地図をクリックすることにより入力して行う。

3.3.2.1 建物など直線的なオブジェクトの位置情報の入力

建物のような長方形、あるいは正方形の形をしているオブジェクトはそのオブジェクトの左上の頂点座標 (X_{min}, Y_{min}) と右下の頂点座標 (X_{max}, Y_{max}) の2点の座標を入力することにより位置情報の入力を行う(図4)。しかし、この入力方法だと入力は2点を入力するのみで簡単だが、図4に示したように実際の地図上のオブジェクトの位置と入力したオブジェクトの位置にずれが生じてしまう。

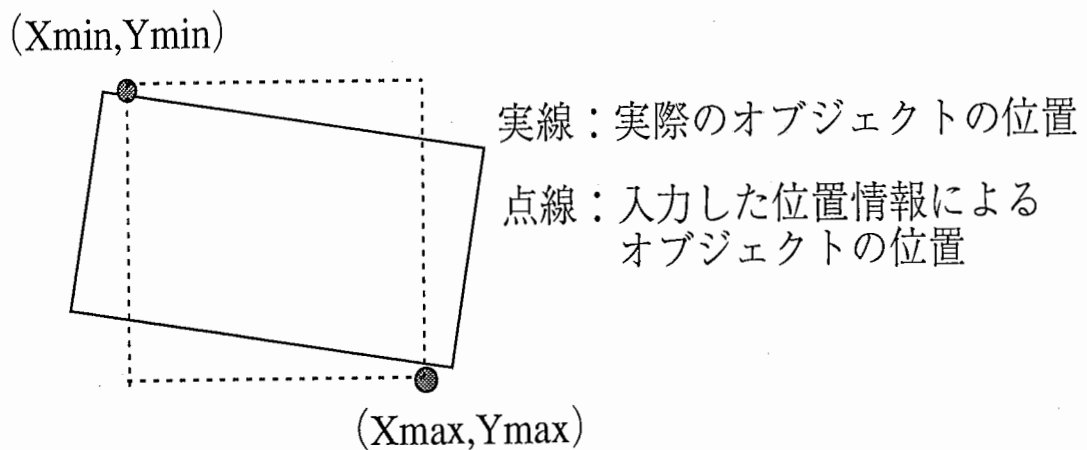


図4 位置情報の入力(1)

3.3.2.2 川、道など非直線的なオブジェクトの位置情報の入力

道、川のような非直線的なオブジェクトの位置情報の入力は比較的直線的になっているところで区切って入力を行う。オブジェクトの中心を通るような直線の両端の座標（ $(Xmin, Ymin)$ 、 $(Xmax, Ymax)$ ）とその中心線からのオブジェクトの幅（Width）を入力する（図5）。

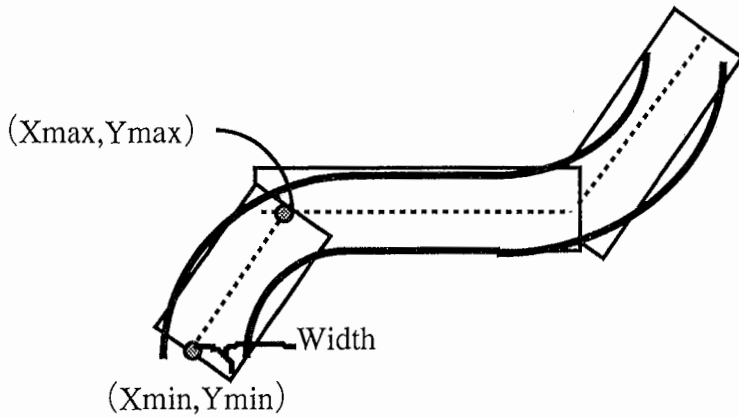


図5 位置情報の入力（2）

4. むすび

今回、オブジェクト指向による地図データベースの作成を行った。研修期間が1ヵ月間しかないにもかかわらず、オブジェクト指向に関する基礎概念を理解することから始めたため、システム開発の過程でローケン・キム キュンホ研究主任や同研究所の皆様の協力を受け、ようやくシステムを完成することができた。今回のシステムの機能は指定したオブジェクトの情報を画面に表示するというものだが、今後ユーザーの指定した情報を満たすオブジェクトを表示したり、目的地までの道案内をしたり、さらに高度なサービスの提供が実現されるのだと思う。

謝辞 今回の実習にあたり貴重な御協力、御教示を頂いたローケン・キム キュンホ研究主任、および同研究所の方々に深く感謝致します。

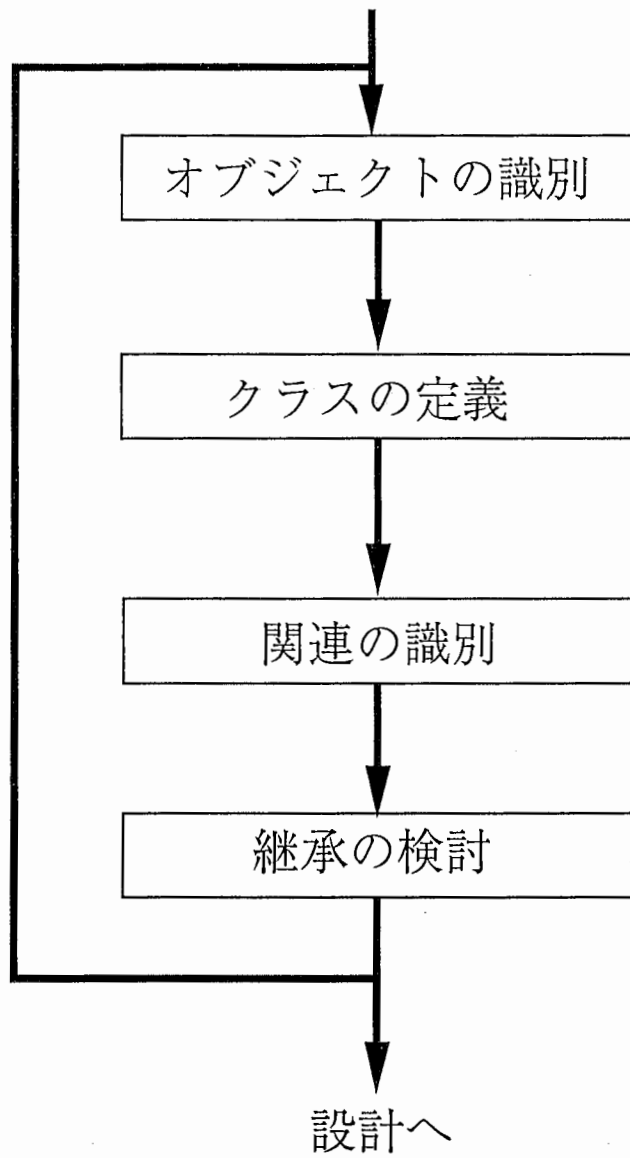


図6 分析の流れ

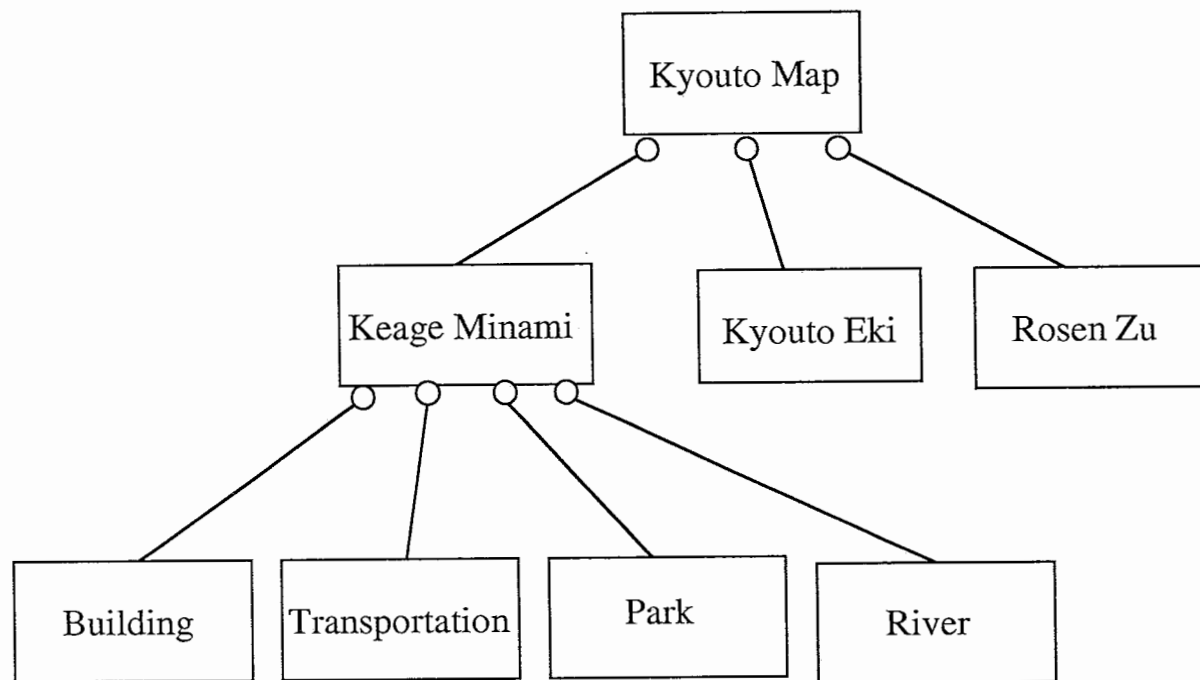


図7 クラスカテゴリー

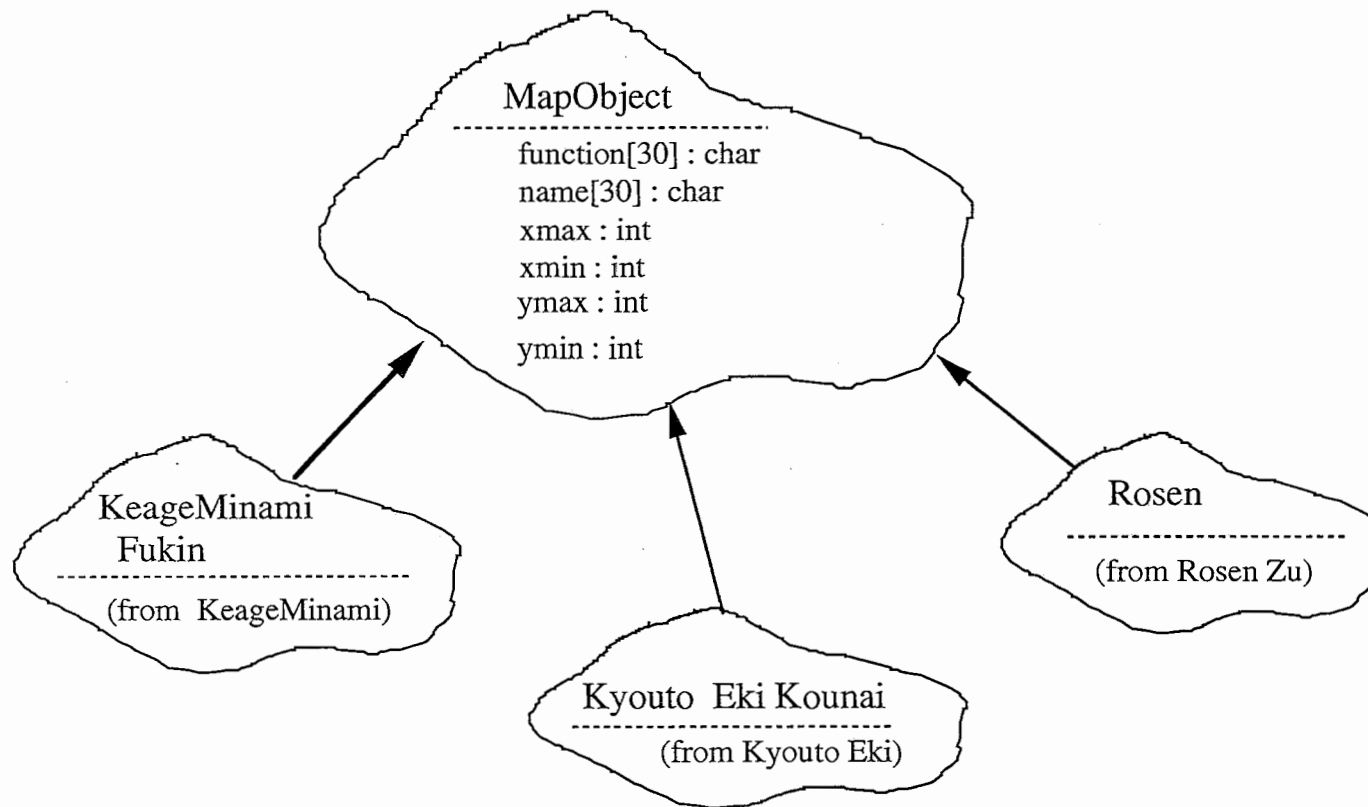


図8 クラス図

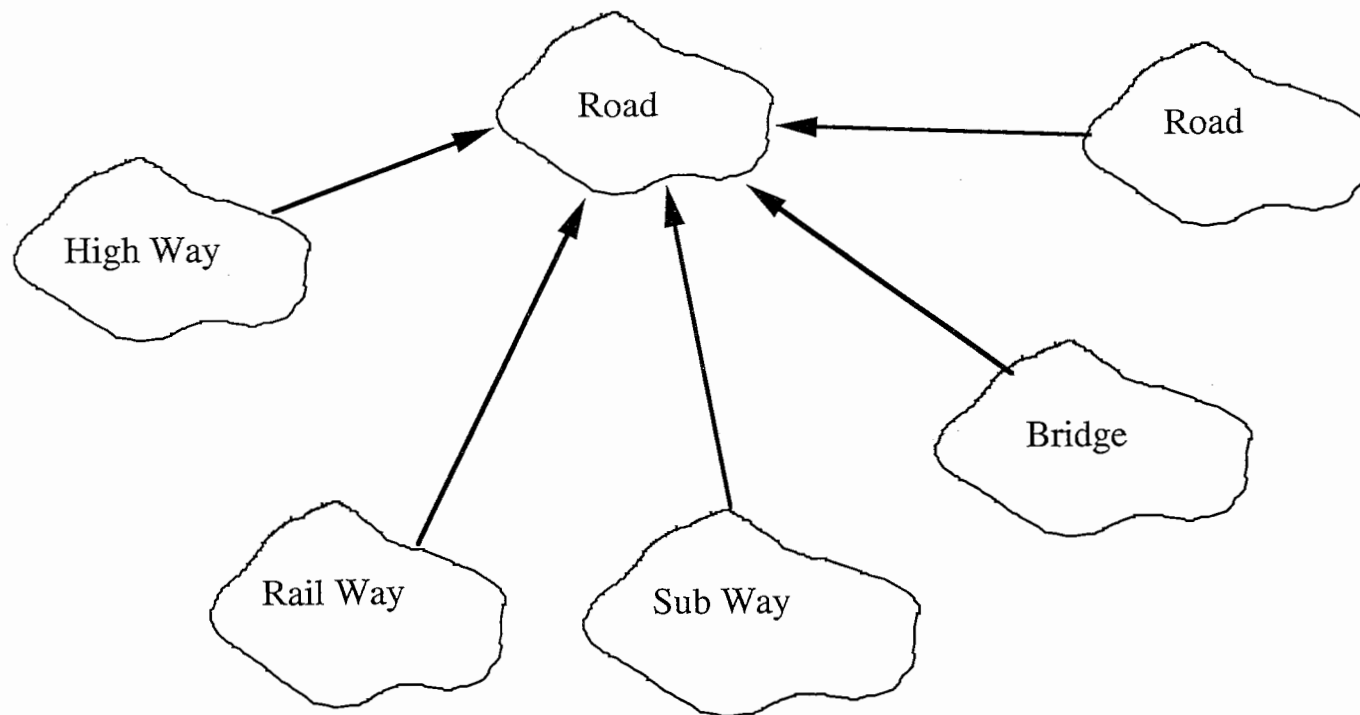


図9 クラス図

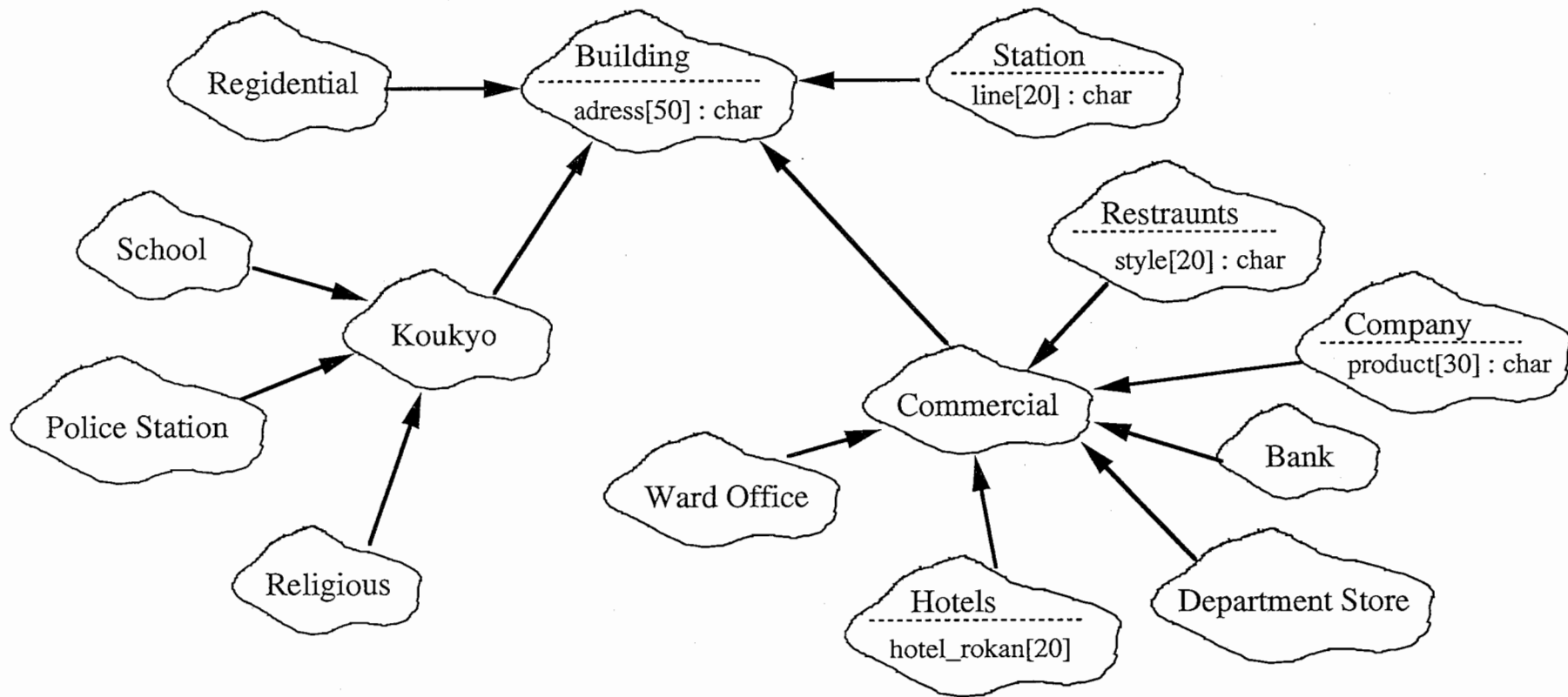


図 10 クラス図

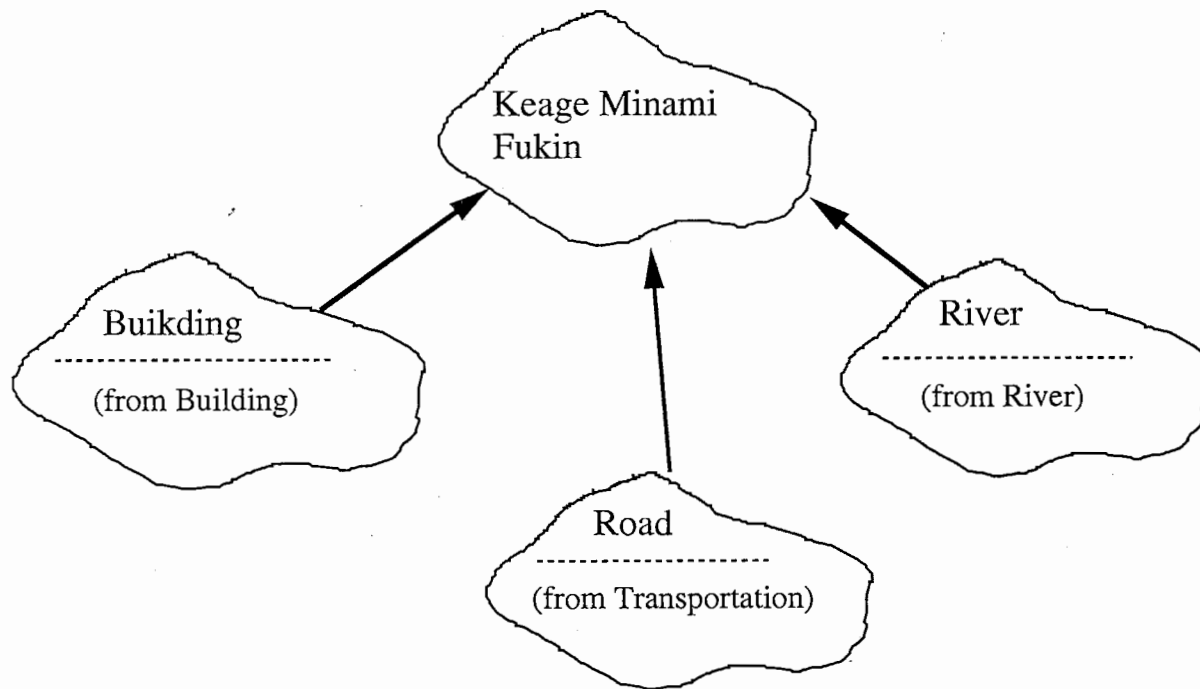


図 1 1 クラス図