

1Q-4

オブジェクト指向意味ネットワークモデル OSMANについて

石川 洋 ・ 宇田川 佳久 ・ 辻 秀一 ・ 西川 正文
(三菱電機㈱ 情報電子研究所)

1. はじめに

近年データベースの適用分野は、従来からの事務処理だけでなく、設計などをも対象とした広範なものとなっており、従来より以上に、多様で複雑なデータを扱わなければならないようになってきている。また、そのデータは、設計作業により、頻繁に変化し詳細化される。

上記のような、複雑なデータを扱うためのものとして、オブジェクト指向アプローチ^[1]が注目されつつある。本文中で述べるOSMAN^[2,3](Object-oriented Semantic Model for handling Nonmonotonic data)はERモデルに対して、オブジェクト指向の概念を導入したものである。

本文では、OSMANのオブジェクト指向の特徴を、マンション・レイアウトを例にして述べる。また、OSMANとSmalltalk80^[4]との違いについて考察する。

2. OSMANの概要

OSMANは、以下に示す3種類のノードをポイントで結合することで実世界を記述する。これらのノードは、オブジェクトであると考えることが出来る。

- (1) カーネル (Kernel) と呼ばれるノードは、属性の値、または他のノードへのポイントを保持している。属性値のみを記憶しているカーネルは、ERモデルにおける実体に対応し、ポイントのみを記憶しているカーネルは、関連に対応する。一般にカーネルは、この中間の状態をとる。カーネルは、オブジェクト指向におけるインスタンスに対応する。
- (2) クラス・カーネル (Class-K) と呼ばれるノードは、ノードの分類を行うために用いられる。つまり、いくつかのノードをまとめて、一つの概念として扱うときに用いられるものである。クラス・カーネルは階層構造を作ることができる。クラス・カーネルは、概念の記述を行うということから、オブジェクト指向におけるクラスに対応する。
- (3) 幾何カーネル (Geometric-K) と呼ばれるノードは、図形・画像といった幾何情報を記憶するため

のものである。幾何カーネルのデータ構造は、幾何モデラに依存する。言い換えれば、幾何カーネルは幾何モデラとの情報の受け渡しをするためのものである。

これら三つのノードはそれぞれ、図1に示す図形で表わされる。ノード間の結合は、ノードを線で結ぶことにより表わされる。

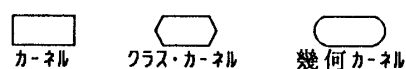


図1. 各ノードのシンボル

3. OSMANにおけるオブジェクト指向

本章では、不動産物件としてのマンションのレイアウトを例として、OSMANにおけるオブジェクト指向の特徴について述べる。ここでは、マンションとは、図2に示すような物件の集合体と考えられる。

モデル化する際の要件として、マンション・レイアウトを、単なる図形情報としてではなく、「部屋」などの空間の集りとして認識し、操作できることを考える。この為には、空間を用途別に分類することと、空間同士の隣接関係を識別できるようにすることが必要となる。この方針に従ってモデル化した例が、図3である。

個々のマンションはクラス・カーネル「マンション」のインスタンスである。また、個々のマンションは物件の集合体となる。各物件はそれぞれ特有の構造を持っており、これを表現するためにインスタンスである物件の下に、更にクラス階層が作られる。

物件の構造は、空間の集りと、壁などの構成要素の集りという2通りの視点から記述されている。クラス・カーネル「空間」は、その下に、より詳細化されたクラス・カーネルを持っている。この様なクラス階層を構成することにより、データの詳細化に対応することができる。

個々の壁材を表わすカーネルと、部屋を表わすカーネルはそれぞれ関係付けられており、部屋同士が共有する壁を認識することで、部屋の隣接関係を知ることができる。また、壁材や柱材は、部屋などの空間と関係なく扱うことも可能である。この様にして、多数のユーザからの、データに対する異なる要求を統合することが可能である。

以上のように、OSMANでは、多くの異なる立場

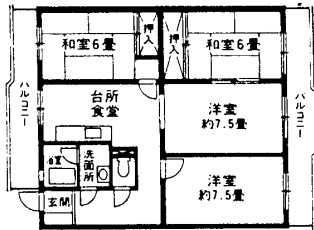


図2. マンション・レイアウトの例

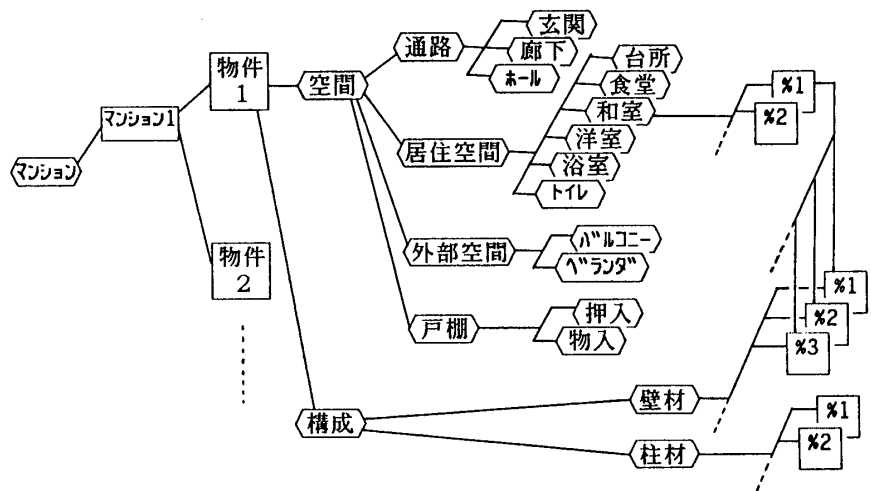
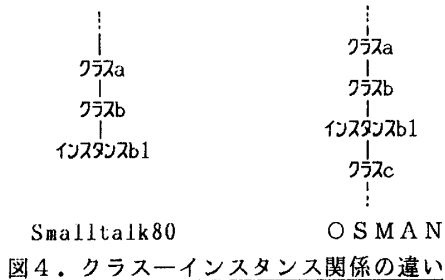


図3. OSMANにおけるレイアウトの表現例

で認識された「もの」をオブジェクトとして定義し、それらの間に関係付けを行ない、さらにクラス階層を定義することで、多様に複雑なデータを表現することができる。また、データに対して、各ユーザごとの認識にそった処理を定義することができる。

4. Smalltalk80とOSMAN

Smalltalk80とOSMANとを比較した場合、もっとも顕著な違いは、クラスとインスタンスの関係にある。前章で示したように、OSMANではインスタンスであるカーネルの下に、さらにクラス・カーネルによるクラス階層を構成することができる。Smalltalk80の場合には、インスタンス変数を用いて、ある程度までは暗にこの様な構造を作ることはできるが、陽にこれを表現することはできない。



Smalltalk80 OSMAN
図4. クラス-インスタンス関係の違い

この様な違いは、両者におけるインスタンスの扱われ方の違いに起因していると言える。Smalltalk80では、クラスは多数のユーザに共用されるが、インスタンスは基本的にアプリケーションの実行中しか存在せず、保存され、共有されることはない。しかし、OSMANの場合には、インスタンスであるカーネルは、半永久的に存在し、多数のユーザに利用され、変更されていく。また、カーネルの総数は膨大なものとなる。さらに、設計分野などへの適用を考えると、カー

ネルは、非常に複雑な構造を持つ実体を表わす必要がある。

よって、OSMANではインスタンスの構造を明確に表現しておく必要がある。構造を表現することにより、そのインスタンスが意味するところがユーザに対して明らかにされる。これにより、データ共用を達成することが容易になる。また、クラス階層によりインスタンス構造を表現することで、あるインスタンスの変更などの影響の範囲を限定することができる。これにより、データ独立を保障できる。

5. おわりに

本文では、OSMANにおけるオブジェクト指向について述べ、またSmalltalk80との比較を行なった。設計分野などへのデータベースの適用を考えた場合、複雑なデータ構造を自然に表現でき、またそれに対する操作をも記述できることから、データベースにオブジェクト指向を取り込むことは非常に有益であると思われる。さらに、データベースとプログラミング言語との差異を考察することで、インスタンス-クラス階層を導入することが有益である、という結論を得た。

参考文献

[1] 田中：“オブジェクト指向データベース・システム—その背景と概念”、bit Vol.20 No.6、共立出版、1988.6
 [2] 石川、宇田川他：“建築CADデータベースのスキーマ定義について”、情報処理学会DB研究会64-2、1988.3
 [3] 宇田川、石川他：“実体一関連モデルのためのデータ定義言語と従属性の扱いについて”、情報処理学会DB研究会65-4、1988.5
 [4] アスキー書籍編集部 編：“Smalltalk 入門”、アスキー出版局、1986