

CAD用データベースにおけるレイアウト図について

岡本 基良 宇田川 佳久

三菱電機(株) 情報電子研究所

okamoto@isl.melco.co.jp udagawa@isl.melco.co.jp

CADで扱う設計情報をデータベースに格納するためには対象となる設計情報に応じたデータモデリングが必要である。また設計情報のモデル化に際しては意味の記述能力が重要となる。そのため対象となるCADデータの性質・特徴を十分に検討する必要がある。

我々は建築CADのための拡張可能DBMSの開発中である。その一環として建築レイアウト図の表現方法を検討した。レイアウト図の構造を柔軟に表現するためのクラス階層の構成と生成、追加、変更、表示を含むメソッドをSmalltalk-80によって記述した。さらにレイアウト図におけるオブジェクト相互の関連を意味の上から検討し、特にインスタンス相互の対等な関連づけの重要性を、例として部屋と部屋の隣接関係の記述を取り上げて示し、考察を行なった。

On an architectural layout in CAD databases

Kiyoshi Okamoto Yoshihisa Udagawa

Computer & Information Systems Laboratory

Mitsubishi Electric Corporation

5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan

To store design information handled by CAD system, data modelling technique for those information is important. In addition, the data model must have ability to describe the semantics of the design information. So we have to examine CAD data structure and characterize that. We are implementing the prototype of extensible DBMS for CAD, dealing with Architectural Layouts. In this paper, we propose a flexible representation of the layouts in the database, describe both class definition and data manipulating methods such as generation, insertion, modification and display in Smalltalk-80. We discuss the relationship among objects in architectural layouts, especially in semantic aspect. We find that equal-level instance-instance relationship plays an important roll in modelling architectural layouts, especially in describing the adjacent relationship among rooms.

1 はじめに

各種 CAD などの実用化に伴い設計情報の格納の点で現在の関係データベースの限界が指摘されてきた。また次世代データモデルの研究が盛んになって久しい。近年ではオブジェクト指向データベース宣言 [2] と第三世代データベースシステム宣言 [5] が相次いで出され議論は盛んである。これらの研究の発端は CAD が扱うデータが平坦ではなく構造を持っていること、それらのデータ型が頻繁に、時には動的に生成されること、データ間で複雑に関連づけが存在し、かつデータ自身と同様に動的な生成、変更などの操作を伴うなどの事実である。このためオブジェクト指向などに基づいた新たな CAD 用データベースが提案されてきた。(例えば [1] [4] [7] [9])

CAD が扱う設計情報は非常に多岐にわたる。それらの設計情報のうち中心的なものは図面データであろう。それゆえ設計図面を扱うレイアウト CAD が重要な役割を果たすことになる。設計図面も多種多様であるが、その中でも建築設計のレイアウト図は、論理回路図や機械設計図と違った性質を有すると考えられる。建築レイアウトは構造を持った実体を表現しており複合オブジェクトとして考えることが出来る。

我々は建築レイアウト図面を扱う CAD のための拡張可能 DBMS について研究試作を行なっている。このデータベースではユーザ定義のデータ型が動的に定義可能であり、さらにデータベースオブジェクト間の関係づけを動的に行うものを目指している。今回、この拡張可能データベースで建築レイアウトを扱うために建築レイアウト図を複合オブジェクトとしてとらえ、その分析を行なった。その結果得られたレイアウト図の性質に基づき、データ構造とデータ操作を定義し、レイアウト図のデータモデリングを試みた。また、建築レイアウト図を扱う CAD に対する要求に基づき、レイアウト要素相互の幾何的な位置関係の記述を上記のモデルを用いて考察し、その実現のための機構について検討した。

本稿では建築レイアウト図として部屋の見取図を取り上げる。部屋の見取図を複合オブジェ

クトとしてとらえ、どのようなデータ構造を与えてデータベースに格納すべきかについての検討をデータ構造とデータ手続きの定義を通じて行なった。

以下、2 では建築レイアウト図の性質を複合オブジェクトとしての観点からまとめる。3 では 2 での考察に基づいて建築レイアウト図のデータ構造を Smalltalk-80 [6] のクラス定義とメソッド記述によって示す。ここではレイアウト図内の構造的な階層を柔軟に格納するための機構について述べる。また表示操作のインタフェースプロトコルについても触れる。4 ではレイアウトオブジェクト相互間のリンクについて考察し、対等な関係のリンクが部屋と部屋の隣接関係を記述できることを指摘し、そのための機構について概要を示す。最後に 5 において議論をまとめ、さらに今後の課題について触れる。

2 建築レイアウト図について

ここでは建築レイアウト図の持つ性質について分析する。建築設計と言っても、その作業にはいくつかの種類が存在する。一般に建築設計は、意匠設計、設備設計、構造設計などに分かれており、それぞれ扱う設計図のデータやその利用の仕方も異なっている。またデータに関する差異は設計作業自身の種類のみならず、設計対象の建築建造物の種類によっても生ずる。

しかし、見取図に属するものに関しては、基本的にデータの構造や利用のされ方が一貫している。例えば、マンションのような高層の集合住居建築物のレイアウトを考える。マンション全体は複数のほぼ同様の構造を持ったフロア毎のレイアウトから構成されている。各フロアのレイアウトは共用スペース（廊下、ダクト、エレベータ、階段など）と各戸毎の内部レイアウトから構成され、一戸分のレイアウトは部屋と空間（廊下など）とそれらを区切る壁、ドアや照明、電源などの付帯設備などから構成されている。一戸建の住居用建築のレイアウトについてもほぼ同様の構造をなしている。

つまり、建築レイアウトは部屋から構成される複合オブジェクトであると言うことができる。

建築レイアウトは複合オブジェクトの中でも、あるレベルのオブジェクトがさらに下位のオブジェクトより構成される。つまり複数レベルにわたる階層性を持っている。その点で建築レイアウト図は分子オブジェクト [3] であるとも言える。階層の各レベルのオブジェクトはそれぞれ実体の物理的な表示のための情報のみではなく、各種の属性情報を保持している。

建築レイアウトを記述する建築レイアウト図は矩形と曲線から構成される図形の集合である。それぞれの部分図形は特定の意味を持っている。その意味は二種類存在する。第一は図形そのものとして本来持っている意味である。図形の形状、全体の中での位置関係、周囲の図形との関わり方などがこれに当てはまる。これを幾何情報と呼ぶ。第二の意味としてはその図形の主に内的な属性に関するもので、図形の名前、種類、用途などが考えられる。これを属性情報と呼ぶ。

本稿では以下、建築レイアウト図として二次元図形の集まりである図面を考えることにする。図面内の図形で表示されるそれぞれの部屋は、さらに下位部分と関連付けられており、それらには幾何情報と共に属性情報が存在するものとする。(図 1) では、名前が B という部屋についての図形情報と属性情報についての関連を示している。

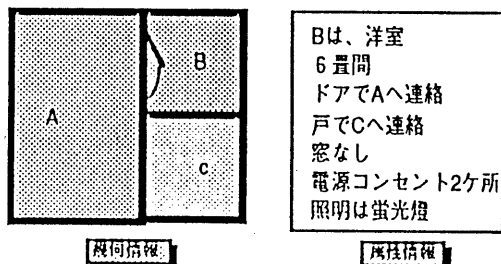


図 1: 図形情報と属性情報の例

建築レイアウト図は部屋や廊下などの空間から構成される。部屋の実体は部屋自身の情報と部屋内設備 (窓、コンセント、照明、etc) の情報を保有する。ゆえに建築レイアウト図中の各オブジェクト (レイアウト自身、部屋、部屋内設備など) はそれぞれ幾何情報と属性情報を

保有していると考えられる。幾何情報は図の表示のための情報でありレイアウト図中の位置関係、形状などがあげられる。一方、属性情報には部屋の種類 (洋室、和室)、面積、窓・照明・電源コンセントなどの部屋内設備に関する情報が考えられる。

また、レイアウト図中の各オブジェクト間の意味的な関係については、二種類の関連づけが存在していることがわかる。つまりレイアウト図内のオブジェクトには部屋がレイアウト図内に存在するという包含関係と、部屋と部屋が隣接しているという隣接関係が存在する。前者はオブジェクト間の従属に類する上下関係である。後者についてはオブジェクト間にそのような上下関係は無く、全く対等な関係である。両者の持つ意味の全く異なるがオブジェクト間に張られたリンクで実現されているという共通性がある。この種類の関連付けのリンクについては 4 で述べる。

3 Smalltalk-80 によるデータ構造の検討

前章で述べたように建築レイアウト図には複合オブジェクトとしてのいくつかの特徴が存在する。これらを踏まえて建築レイアウト図をデータベースに格納する際のデータ構造の検討を行なった。データ構造の検討に当たって、我々は Smalltalk-80 を用いて建築レイアウト図のデータ構造をクラス定義し、データ操作の手続きをメソッド定義した。また建築レイアウト図を画面表示させる手続きの例を通じてレイアウト図のデータ構造の検討も合わせて行なった。

本章ではレイアウト図内の構成要素が、その意味的な内容を検討した結果から三種類のオブジェクトに分類できることを述べる。その分類に基づいて、建築レイアウト図をデータベースに格納する時のデータ構造を Smalltalk-80 のクラス定義によって示し、そのデータに対する操作の手続きをメソッド記述の形で定義する。

3.1 レイアウト図のオブジェクト構成

建築レイアウト図を複合オブジェクトと考えると大きく分けて次に示す三種類のオブジェク

トより構成されていることがわかる。

1. レイアウト図オブジェクト
2. 部屋オブジェクト
3. 部屋内設備オブジェクト

ここでレイアウト図オブジェクトはレイアウト図のオブジェクトの階層構造の中で最上位に位置付けられる単位であり、一つのレイアウト図全体を表す。部屋オブジェクトはレイアウト図内の個々の部屋をモデル化したものである。部屋内設備オブジェクトは各部屋毎に付随した照明設備、水まわりの設備（水道、排水口など）、電源コンセントなどである。

これらのオブジェクトは全体として三層の階層構造を形成している。つまり部屋内設備オブジェクトは部屋オブジェクトに上下関係で関連しており、部屋オブジェクトはレイアウト図オブジェクトにやはり上下関係で関連している。これらの階層性は図形的な意味で言うと包含関係といえる。（図2）この階層構造を特に構造階層と呼ぶことにする。

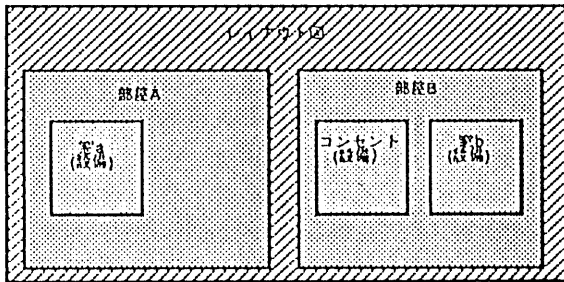


図2: レイアウトオブジェクトの包含関係

3.2 幾何情報と属性情報の取り扱い

ここではレイアウトオブジェクトに関する幾何情報と属性情報の実現について述べる。オブジェクト指向言語として最も「オブジェクト」の概念を徹底させた構文と意味を持つ Smalltalk-80 を用いて建築レイアウトのデータモデリングを行うと、レイアウト図中の各オブジェクトのレベルにおいて幾何情報と属性情報に分離するクラス設計は不自然である。なぜな

ら Smalltalk-80 におけるオブジェクトの概念はデータとデータに関する手続きを統合した存在で、かつオブジェクトに関する情報は全て当該オブジェクト自身が保有することが基本になっているからである。よって建築レイアウトのデータモデリングに際してはレイアウト図オブジェクト、部屋オブジェクト、部屋内設備オブジェクトの三者ともオブジェクト内部に幾何情報と属性情報を合わせて保持させることにする。

これら三種類のレイアウトオブジェクトはレイアウト図面を構成している二次元図形であるという基本原則に関して共通した性質を持っている。このため幾何情報について三者とも共通のデータ構造を持つべきである。よって幾何情報を保持するオブジェクトの構造も三者同一のものとする。

一方、属性情報に関してはレイアウト図、部屋、部屋内設備ではそれぞれ対象とする情報構造が異なる。また建築レイアウト CAD などによる編集作業においてはオブジェクトの属性情報の構造をユーザが自由にカスタマイズすることが予想され、さらに動的な変更や再定義の起こる可能性が高い。そのため属性情報の保持はレイアウト図内の既に述べた三種類のオブジェクトにおいて直接定義せずに、新たに専用のオブジェクトクラスを定義すべきである。レイアウト図内の個々のオブジェクト自身において直接に属性情報の構造を規定するとレイアウト記述能力の柔軟性や拡張性を欠くことになり、クラス設計としては適切ではない。これを実現するためにはパラメタ化されたデータ型を扱うことが必要である。これは、例えば C++ では扱うことができないが [8] Smalltalk-80 では容易に実現できる。

3.3 建築レイアウト図のクラス定義

ここでは建築レイアウト図のデータ構造を Smalltalk-80 のクラス定義によって示す。以下では Smalltalk-80 の習慣に従い、あるクラス SomeClass のインスタンスを aSomeClass とクラス名に不定冠詞をつけて表す。

Smalltalk-80 によるモデリングでは前述の三

種類のオブジェクトを、レイアウト図としてのプリミティブな性質を記述する抽象クラスの直接のサブクラスとして同格に定義し、レイアウト図内の構造の階層性（構造階層）を直接にクラス階層に持ち込まない（図3）。

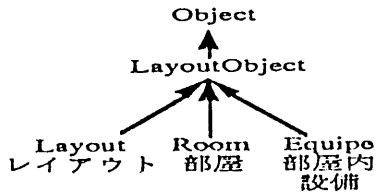


図3: 建築レイアウトのクラス階層

以下にその理由を述べる。既に3.1において、建築レイアウト図内の三種類のオブジェクト間に建築構造上または図形の包含関係上の階層構造が存在することを述べた。このレイアウト図内のオブジェクト間の階層構造を特に構造階層と呼ぶことにする。この構造階層をそのままオブジェクトのクラス階層として定義することは得策ではない。この構造階層においてレイアウト図オブジェクトのすぐ下の階層が部屋オブジェクトのみとは限らないからである。2でも触れたが建築レイアウトは部屋、廊下などの空間から構成され、部屋と同格の概念として例えば廊下オブジェクトがレイアウト図の要素として扱われる可能性がある。

もしクラス階層として3.1で示した包含関係の構造階層を採用すると仮定する。Smalltalk-80は元来、単一継承をサポートする言語である。よってクラス階層はルートを持つ木構造をなす。レイアウト図を編集する際に部屋オブジェクトから廊下オブジェクトへの変更を行うと、これはクラス階層の木の途中のノードを変更することに相当する。

この操作はクラス階層の木においては一つのノードの変更にはとどまらず、変更するノード以下の子孫全てに影響が波及し、変更以前のクラス階層とは全く別のクラス階層を冗長に生成することになり、相当のオーバーヘッドを伴う。

これは動的なオブジェクトの生成、変更が頻繁に起きる建築CADには好ましくない。

よってレイアウト図のクラス定義においては包含関係の階層構造（階層構造）をクラス階層の定義には用いずに、レイアウト図内のオブジェクトをクラス階層上は全て同格のものと位置付ける。包含関係上の階層構造を表現するためにはクラス階層とは別に component-of リンクを用いることにする。これによって構造階層を表す木構造の中間のノードに変更が起きても変更対象ノードの子孫のクラス階層全体を再構築するためにリコンパイルすることなく動的に処理することが容易になる。

さらに別の例を考える。フロア毎のレイアウト図と同じクラス階層を用いてマンションなどの高層建築レイアウトの記述を行なおうとすると、構造階層の木に新たなノードを挿入する必要がある。component-of リンクを用いる方法ではこのノードの挿入に対しても動的な挿入が容易であり、クラス階層を用いるよりも構造記述を柔軟にできる。図4の例では、レイアウト図の編集途中で、構造階層の中間ノードに当たる部屋オブジェクト (aRoom) から廊下オブジェクト (aPassage) に変更する時に二本の component-of リンクを変更するのみで処理される様子が示されている。

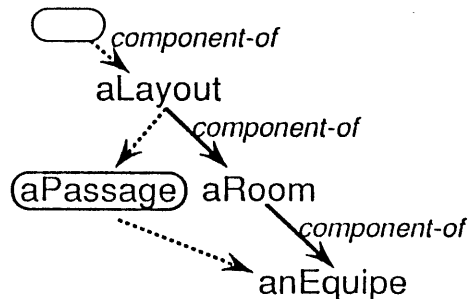


図4: component-of リンクによる階層関係

以下では建築レイアウト図実現のための、実際のクラス定義について述べる。レイアウト図内にある三種類の各オブジェクトはそれぞれ内部に幾何情報と属性情報を持っている。レイアウト図内にある三種類のオブジェクトが幾何情報に関して基本的に同一の構造を持つことは先

に述べた。そこで、それら三種類のオブジェクトの共通な性質を定義する抽象クラスを各レイアウト図内のオブジェクトクラスのスーパークラスとして定義する。レイアウト図内のオブジェクトの抽象クラスとして `LayoutObject` を設定する。`LayoutObject` はインスタンス変数 `figure` にそのオブジェクトの形状を、`position` にそのオブジェクトの座標位置を保持する。この2つのインスタンス変数がレイアウトオブジェクトの幾何情報を保持する。

以下、クラス `LayoutObject` の定義例を示す。

```
Object subclass: #LayoutObject
  instvar: 'figure, position, property'
```

`LayoutObject` は幾何情報の他に `property` という属性情報を保持するインスタンス変数を持っている。この属性情報を規定するオブジェクトとしてクラス `Property` を定義する。ここで、`type` はそのオブジェクトの種類を示すインスタンス変数であり、`components` はそのオブジェクト内に存在するオブジェクトを保持するインスタンス変数であり、`component-of` リンクを実現している。`connectors` にはオブジェクトの隣接関係を示す結合オブジェクトが保持される。この結合オブジェクトについては後述する。

以下、クラス `Property` の定義例を示す。

```
Object subclass: #Property
  instvar: 'type, name, size,
           components, connectors'
```

以上、レイアウト内部のオブジェクトの共通の抽象クラス `LayoutObject` について述べた。`LayoutObject` の構造を図5に示す。

これまで述べてきたようなクラス定義によって実現された `component-of` リンクによる建築レイアウト図のデータ構造を図6に示す。この例ではレイアウトの階層構造が `component-of` リンクによって実現され、属性情報が `property-of` リンクを通じて `aProperty` を参照することにより実現されている様子が示されている。

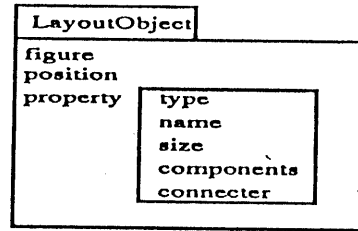


図5: オブジェクトの構造

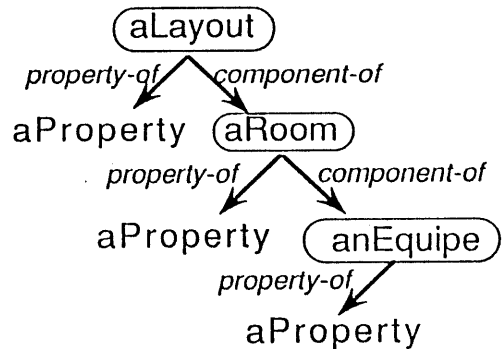


図6: 建築レイアウト図のデータ構造

3.4 データ操作

ここでは建築レイアウト図内のデータに関する生成、追加、検索、更新、表示といったデータ操作手続きを Smalltalk-80 のメソッド記述によって示す。まず、オブジェクトの生成は次のようなメソッドによって実行される。ここに、`aPoint` はレイアウト表示の基準となる点であり、`aFig` はレイアウトの形状を示すオブジェクトである。インスタンス変数 `property`、`components` もそれぞれ初期化される。

```
LayoutObject creatAt: aPoint
  figure: aFig

~ self new position: aPoint
  figure: aFig
  property: Property new
  components: Set new
```

また、追加、検索、更新については次のよう

な構文を用いて書くことができる。ここではレイアウト図に対してある部屋を追加、検索、更新する例を示す。aCondBlock は検索条件を記述するブロック式であり、anUpdateBlock は更新手続きを記述したブロック式である。

```
aLayoutObject append: aRoom

aLayoutObject select: aCondBlock

aLayoutObject update: aCondBlock
    with: anUpdateBlock
```

レイアウトの表示については以下のような Smalltalk-80 のメソッドを定義することによって実現できる。ここでは、抽象クラス LayoutObject に表示手続きに関するインタフェースプロトコルを定義することによってレイアウト図内の各階層のオブジェクトが外部から全く同じメッセージによって表示させられることが可能になっている。aLW はレイアウト図を表示するウインドウを示す。

```
LayoutObject methodsFor: 'displaying'
```

```
display
    self displayAll

displayAll
    self displayOn: aLW.
    self property components
        do: [:obj | obj display]

displayOn: aLW
    self figure displayOn: aLW
        at: position.
    self property displayOn: aLW
```

レイアウト図全体、部屋、部屋内設備の各レベルへの表示操作は全て同一のメッセージ display によって起動される。また、ユーザがある部屋を表示しようとするとその部屋内の設備も同時に表示することを暗黙のうちに求められることが多いが、この際も本稿で提示したシ

ステム内ではメッセージ display がそれぞれに送られる。つまり、レイアウト図上の階層構造の上位のオブジェクトに一度メッセージを送れば、メッセージが内部で再帰的に伝搬されて、全てのレイアウトオブジェクトの表示ができることになる。(図 7)

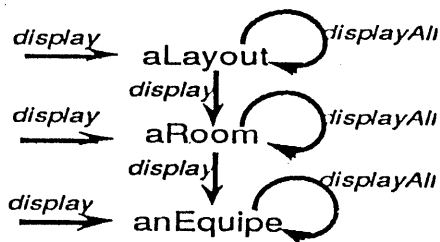


図 7: 表示操作のメッセージ伝搬機構

ここで示した、表示という操作手続きに対して建築レイアウト図全体、部屋、部屋内設備の三者がいずれも同じ振舞いをするのは「インタフェースの統一」というオブジェクト指向パラダイムの原則に基づくものであり実世界の対象をより忠実に表現していることになる。

4 レイアウトオブジェクト間のリンク

本章ではレイアウト図内に存在するオブジェクトの間に存在する関連付けについて考察する。一般にオブジェクト相互間に存在する関連付けとしては、従属関係に代表される上下関係の関連づけと、上下関係や包含関係を持たないオブジェクトとして対等なレベルでの関連付けの二種類が存在する。(図 8)

建築レイアウト図内のオブジェクト間に存在する関連付けのリンクにも、その関連付けの持つ意味的内容の違いより二種類のリンクが存在することは 2 で既に述べた。

建築 CAD において扱うデータではデータベースインスタンス間の関連付けがデータ自身の追加、更新、削除に伴って動的に変更される。さらに建築 CAD においては、これらデータベースインスタンス間に存在する関係の持つ意味がデータそのものと同様に重要な役割を果たす。

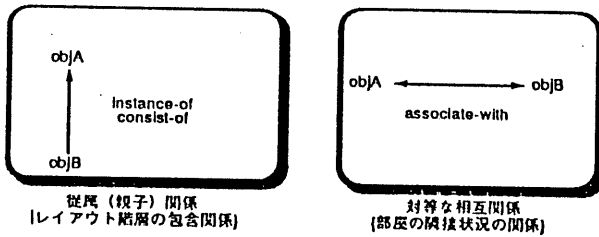


図 8: 二種類のオブジェクト間リンク

従ってこれら二種類のリンクづけを動的に行なえることがレイアウト図のデータ構造を規定する上で非常に重要である。

以下ではインスタンス相互間に存在する二種類の関連付けについてそれぞれ述べる。

4.1 従属関係のリンク

このカテゴリに属するリンクは親子関係のリンクとして広く知られている。オブジェクト指向モデリングにおいては、あるクラスオブジェクトとそこに属するインスタンスオブジェクトとの関係 (instance-of 関係) が代表的である。建築レイアウト図の例では部屋とその内部にある部屋内設備といったレイアウトオブジェクト間の包含関係があげられる。他には is-a 関係や part-of 関係などもこのカテゴリに存在すると考えられる。

4.2 対等関係のリンク

建築レイアウト図では先に述べた従属のリンクだけではなく、インスタンス相互の対等なレベルの関係 (associate-with 関係) のリンクが存在する。この関係ではリンクで結ばれたオブジェクト相互間に上下関係が存在しない。一般のデータモデリングでは関係記述として意味的に継承 (inheritance) や集約化 (aggregation) などの関係を利用することが多いが、この対等なレベルでのオブジェクト間リンクは ad hoc な意味づけではなく、特定の意味に用いられることは少ないように思える。しかしこの対等関係のリンクがレイアウト図のデータモデリング

では非常に大きな役割を果たす。それはレイアウト図上のオブジェクト間の隣接関係の記述においてである。これは部屋同士の隣接関係の記述として建築レイアウトオブジェクトの中で特徴的な関係であると考えられる。

4.3 部屋の隣接関係のモデル化

レイアウトの中で部屋と部屋の隣接関係はドアまたは壁を仲介して記述できる。例えば、ここで三つの部屋からなるレイアウト図 (図 9) を考える。この建築レイアウト図の例では、洋室 8 畳は洋室 6 畳とドアを介して隣接している。また洋室 8 畳と和室 6 畳とは壁を介して隣接している。

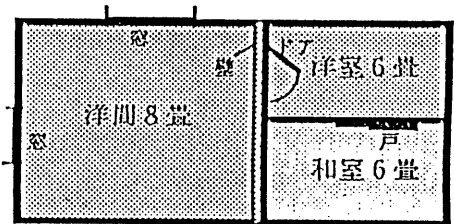


図 9: 建築レイアウト図の例

建築レイアウトを構成するオブジェクトのうち、ドア・壁を部屋と部屋を結合するオブジェクトとして設定し、それらの結合オブジェクト (connector) を通じたリンクによって部屋の隣接関係を記述する (図 10)。connector の機構は基本的には、レイアウトオブジェクト間の相互参照を connector を仲介させて、隣接オブジェクトポインタを保持するリストを connector のインスタンス変数に格納することにより実現する。例えば、クラス Association のインスタンスを各要素として保持するクラス SortedCollection のインスタンスを定義して、connector に aSortedCollection を格納することにより実現できる。この時、anAssociation が connector によって、接続されている双方の aRoom を参照している。ここで特に注目されることは本稿で提案した connector を採用する

と、部屋オブジェクト (aRoom) は壁 (connector) を仲介させない限り隣接している部屋オブジェクト (aRoom) や廊下オブジェクト (aPassage) を参照することも、またはその存在を知ることができないことである。これは現実存在する建築物の隣接関係の意味をより忠実に反映したものになっている。

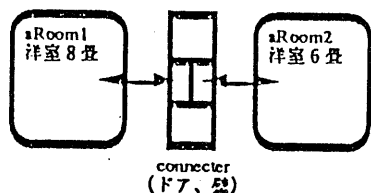


図 10: 部屋の隣接関係

5 おわりに

本稿では、建築レイアウト図を扱う CAD のための拡張可能データベース開発の一環としてレイアウト図を複合オブジェクトとしてとらえ、その構造と性質を検討した。検討に当たっては Smalltalk-80 による建築レイアウト図のクラス定義とデータ操作のためのメソッド記述を行なった。またデータ操作のうち、特にレイアウト表示操作例を通じて建築レイアウト CAD に要求されているデータの格納構造を検討した。

この中で、レイアウト図内に存在する構造的な階層構造を特に構造階層として取り上げ、これをそのままクラス階層に用いるとデータ構造の動的な変更に対して柔軟な処理が困難であることを示し、クラス階層によらないレイアウトオブジェクトの構造階層の表現方法を提示した。

また建築レイアウト図に関するデータベースオブジェクトの中には、オブジェクト間の関係が二種類存在することについて述べ、そのうちで建築レイアウト図においてはそれらの二種類の関係のうちで、特にインスタンス同士の対等な関連づけが大きな役割を果たすことを、部屋と部屋の隣接関係の記述例を提示することにより示した。そこで、建築レイアウトオブジェクトの隣接関係を記述するためのメカニズムの概

要を示した。

今後の課題としてはレイアウトオブジェクトに関するデータ整合性チェックの機構について検討が考えられる。このデータ整合性制約には、レイアウト図の図形的な制約 (部屋を構成する線分は単独では存在しないなど) と、さらに高度な建築設計作業上の種々の制約 (部屋同士は重ならないなど、面積、体積などの物理量との関わり) が想定される。それらのうち、設計上の制約条件は複雑であり課題が多いが設計支援において今後ますます機能の高度化を要求されると考えられる。

6 謝辞

日頃から御指導頂き、本研究の機会を与えて頂いた情報電子研究所武藤副所長、富沢グループマネージャに感謝します。また本研究を進めるにあたり、議論に加わって頂いた (株) 三菱総合研究所の山内正、飯沼聡の両氏に感謝します。さらに情報電子研究所の皆様にも感謝します。

参考文献

- [1] Afsarmanesh, H., Knapp, D., Mcleod, D., and Parker, A. "An Object-Oriented Approach to VLSI/CAD," Proc. of VLDB 1985
- [2] Atkinson, M. et al. "The Object-Oriented Database Manifesto," Proc. of DOOD International Conf., Kyoto, 1989
- [3] Batory, D. S., Buchmann, A. "Molecular objects, abstract data types, and data models: A framework," Proc. of VLDB 1984,
- [4] Batory, D. S., Kim, W. "Modelling Concepts for VLSI CAD Objects", ACM Trans. Database Syst. 10, 3 (Sept. 1985),
- [5] The Committee for Advanced DBMS Function "Third-Generation Data Base System Manifesto", Memo No. UCB/ERL M90/28, UC Berkeley, 1990

- [6] Gordberg, A., Robson, D. "Smalltalk-80: The language and Its Implementation", Addison-Wesley, 1983
- [7] Maier, D., J. Stein, A. Otis, and A. Purdy. "Development of an Object-Oriented DBMS," Proc. of OOPSLA '86 Conference, 1986
- [8] Stroustrup, B., "What is Object-Oriented Programming?", IEEE Software magazine, May 1988
- [9] Kim, W., Banerjee, J. and Chou, H.T. "Composite Object Support in an Object-Oriented Database System," Proc. of OOPSLA '87 Confernece, 1987
- [10] 岡本、宇田川 「複合オブジェクトとしての建築レイアウト図」 情報処理学会第42回全国大会講演論文集 (p.4-123) 2L-8、平成3年3月
- [11] 岡本、宇田川 「複合オブジェクトとしての建築レイアウト図」 電子情報通信学会データベースワークショップ論文集 (pp.67-75)、平成3年2月