データベース・システム 64 － 2
（１９８８. 3. １５）

建築ＣＡＤデータベースのスキーマ定義について

†石川 洋　†宇田川 佳久　†市川 照久
††西川 正文　†††伊藤 光一

†三菱電機㈱情報電子研究所　††三菱電機㈱ｺﾝﾋﾟｭｰﾀ製作所　†††三菱電機東部ｺﾝﾋﾟｭｰﾀｼｽﾃﾑ㈱

今日、ＣＡＤシステムは、建築設計の分野で広く用いられている。これら建築ＣＡＤシステムの多くは、設計作業の全工程を支援する機能を備えている。しかし、これらのシステムでは、各工程で用いられる設計データの統一がほとんど成されていない。このため、多くの場合に、次の設計工程をスムーズに始められない。この問題を解決するためには、設計データを統一的に管理するためのデータベースシステムが必要である。この様なデータベースには、特定の工程に依存しない形でデータを管理する機能が求められる。本文では、三菱建築ＣＡＤデータベースの基本モデル、データ構造およびスキーマの構造について述べる。また、スキーマ構築のための視覚的作業環境に対して求められる機能について考察する。

SCHEMA DEFINITION REQUIREMENTS FOR AN ARCHITECTURAL CAD DATABASE

Hiroshi ISHIKAWA, Yosihisa UDAGAWA, Teruhisa ICHIKAWA,
Masafumi NISHIKAWA, Kouichi ITOH

Information Systems & Electronics Developmant Lab.
Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura City, Kanagawa, 247, Japan

So far, CAD systems have been used in various fields, including architectural design. Most of these architectural CAD systems have functions for supporting all design processes. But in these systems, data integration of design data that are used in each design process is hardly achieved. Therefore, we cannot begin next design process smoothly at many times. For solution of this problem, we need a database system for unified management of architectural design data. Such database must has some functions of data management independent of every specific design processes. In this paper, we discuss the basic model, data structure and scheme structure of MITSUBISHI architectural database. We also consider some functions that would be required for visual environment of scheme construction process.

１．はじめに

従来の建築ＣＡＤシステムは建築設計の全工程を支援することに成功しているとは言えない。建築ＣＡＤシステムが各設計工程を支援するための機能をすべて用意していたとしても、それは支援ツールを各工程別に揃えたに過ぎない場合がほとんどである。また、各ツールが扱う設計データは別々に管理されており、さらに各設計データにおけるデータの形式も様々である。この様な状況で、建築物の設計データ間の一貫性を保つことは非常に困難であるとの指摘がある[1,2]。

建築設計における全設計工程を支援する建築ＣＡＤシステムを実現しようとするならば、各工程で取り扱う設計データを統一的に管理するためのデータベースシステムが必要である。ここで考慮しなければならないのは、設計データは作業の進行とともに次々とデータが詳細化されていくことと、各工程でのデータに対する視点が様々に異なることである。この様なデータベースシステムには、データの大きな変更に対応できる機能および各設計工程における多様な視点からのデータの取り扱いを支援する機能が不可欠である。

本文で述べる三菱建築ＣＡＤデータベース（以下“建築ＣＡＤ－ＤＢ”とする）は[3,4,5,6]、この様な機能を実現するために、図形データや数値データといった形ではなく、建築オブジェクトそのものをデータとして取り扱おうとしている。建築ＣＡＤ－ＤＢにおいて、建築オブジェクトはInformationと呼ばれる基本単位を用いて表わされる。このInformationには不定長のデータを格納できる。またInformationの間をポインタで結合することで建築オブジェクト間の様々な関連を表現できる。またこのポインタは１対１から多対多までの結合を表わすことができる。この様な特徴により、建築ＣＡＤ－ＤＢはデータの詳細化や建築オブジェクトに対する多様な視点に対応しうる。

本文では、まず建築ＣＡＤ－ＤＢの基本的な概念と構造について述べる。次に建築ＣＡＤ－ＤＢを管理するためのスキーマの構造について述べる。スキーマに対しても建築ＣＡＤ－ＤＢ本体と同様に高い自由度が要求される。そして最後にこのスキーマを定義するためのグラフィックユーザインタフェースについて考察する。ここで問題となるのは、いかにして作業者にスキーマ構造を分かりやすく見せるか、ということである。

２．建築ＣＡＤ－ＤＢ

２．１　基本モデル

建築物の設計工程全体に対応した建築ＣＡＤ－ＤＢを構築しようとした場合、まず建築物を表現するためのモデルについて考察する必要がある。このモデルに対する要件としては、

ⅰ．建築オブジェクトそのものを自然に表現できる。
ⅱ．データの詳細化に対応できる。
ⅲ．多様な視点からの建築オブジェクトの姿を記述できる。

などが挙げられる。そこで我々はこれらの条件を満たすものとして、次の様なモデルを考えた。

ａ．まず、ある建築オブジェクトが存在することを表わすものとしてKernel（核）を考える。次に、その建築オブジェクトをある視点から捉えたときに見える属性をProjection（射像）とする。これにより、建築オブジェクトはKernelにいくつかのProjectionが付随したものとして記述される。これを図で表わすと図１の様なイメージになる。Kernelにはいくらでも Projectionが付随するものとする。

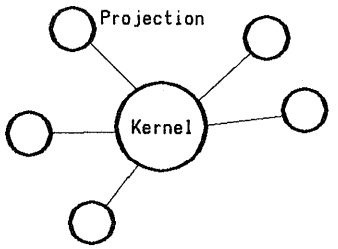

図１．KernelとProjectionの関係

ｂ．２つの建築オブジェクトの間の関連を表現する場合は、２つのKernelがそれぞれもう一方のKernelに付随するという形になる。このとき、一方のKernelからは、他方のKernelは自分に付随するProjectionと見なされる。

このモデルにおいてデータを詳細化することはKernelに付随するProjectionを増やしていくことに相当する。また今までとは異なる視点から見た建築オブジェクトの記述も、Projectionを追加するという形で付加することができる。さらにKernel自体はどの様な視点にも依存せず、建築オブジェクトが存在するということのみを表わしている。これらのことから、本節で示した建築ＣＡＤ

—ＤＢ基本モデルは上記の要件を満たすものであると言える。

２．２　建築ＣＡＤ—ＤＢの構造

建築ＣＡＤ—ＤＢは、上に示した基本モデルをほぼそのままの形で実現しようとしたものである。建築ＣＡＤ—ＤＢは、Informationと呼ばれる基本単位をポインタで結合することで、実世界を表現する。このInformationは、基本モデルにおけるKernelやProjectionを表わしている。Informationは基本的に図２の様な構造をしている。制御部にはＤＢＭＳが必要とする情報が書き込まれている。また、データ部には各種のデータや、他のInformationへのポインタが格納される。データ部の大きさは任意であり、実行中に変更することもできる。個々のInformationにはシステムにより一意に識別コードが割り当てられる。これは制御部に格納される。

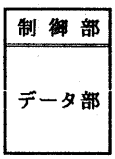

図２．Informationの構造

Informationは、その役割によりいくつかの種類に分けられる。その内で、基本となるのはKer、EKerおよびDPrの３つである。これらの他にシステム内部でのみ用いられるものが数種類存在するが、本文では省略する。

Kerは基本モデルにおけるKernelに対応する。これは建築オブジェクトそのものを表わすものであり、データ部には他のInformationへのポインタを保持する。またここに属性値（例えば建築オブジェクトの名前など）を格納することもできる。

DPrは基本モデルにおけるProjectionに対応する。DPrのデータ部には属性値が格納される。また、他のDPrへのポインタを持つこともできる。これにより、複雑な構造を持った属性を表現することが出来る。これは特に図形データなどを格納するのに有用である。

EKerは、Informationに対する視点を統合／分類するという役割をもつ。これは基本モデルに対して拡張された部分である。通常Informationの間はポインタで直接結ばれるが、EKerを用いることで図３の様に木構造の結合を作ることができる。この木を根の方から見ると、１つの視点が次々と詳細化されていくと考えることができる。また逆に葉の方から見ると、多くの異なるInformationが１つの視点から見た概念に統合されていくと見なせる。これにより、Information間の複雑な結合を記述できる。

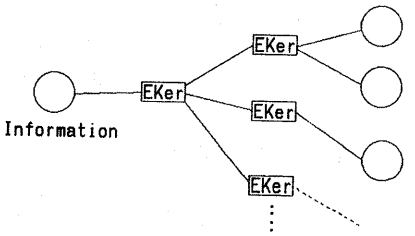

図３．EKerによる木構造

ポインタによるInformation間の結合には、対等型と従属型の２種類がある。また結合のパターンとしては１対１から多対多まで存在し得る。

２つのInformationが対等型で結合している場合、両者のデータ部には共に相手を指すポインタが存在する。これは２つのInformationが互いに独立であることを表す。これは基本モデルにおけるKernel同士の結合に対応する。

従属型で結合している場合には、ポインタは一方のInformationにのみ存在する。この場合、ポインタを持つ方がオーナーとなり他方を所有する形になる。これは基本モデルにおけるKernelとProjectionの結合に対応する。

この結合の型は、結合されるInformationの組合せにより決まる。これを表１に示す。Ker,DPrがそれぞれ基本モデルのKernel,Projectionに対応することと、EKerがポインタの機能の拡張を行なうということを考慮して、この様な形を提案している。

表１．Information間の結合型

Informationの組合せ	結合型	結合パターン
Ｋｅｒ　—Ｋｅｒ	対等型	1:1,1:n,m:1,m:n
Ｋｅｒ　—ＥＫｅｒ	従属型	1:1
ＥＫｅｒ—ＥＫｅｒ	従属型	1:1
ＥＫｅｒ—Ｋｅｒ	対等型	1:1,1:n,m:1,m:n
Ｋｅｒ　—ＤＰｒ	従属型	1:1,1:n,m:1
ＥＫｅｒ—ＤＰｒ	従属型	1:1,1:n,m:1
ＤＰｒ　—ＤＰｒ	従属型	1:1,1:n,m:1

ここまでで述べた建築ＣＡＤ—ＤＢの構造は、実体—関連モデルの諸概念と対応付けることができる。基本的にKerは実体または関連に相当し、DPrは属性に相当する。特にDPrは同一の実体に関する異なるメディアデー

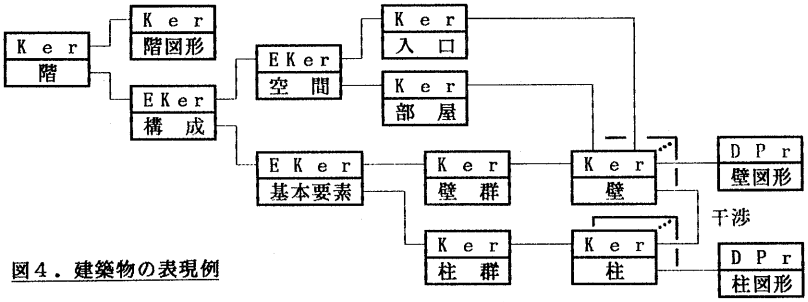

図４．建築物の表現例

タを表わすものであると言うことができる。またEKerは汎化概念(generalization)に対応する。建築ＣＡＤ－ＤＢと実体－関連モデルの対応関係を表２に示す。

表２．建築ＣＡＤ－ＤＢと実体－関連モデル

建築ＣＡＤ－ＤＢ		実体－関連モデル
Ker(属性値のみ)		実体
Ker(ポインタのみ)		関連
関連の内訳	Ker-Ker	集約(aggregation)
	Ker-EKer-Ker	汎化(generalization)
	Ker-DPr	メディアの集約
	Ker-EKer-DPr	メディアの汎化

以上に述べた構造により、建築物の１つの階を表現した例を図４に示す。壁Kerや柱Kerは実際には複数個存在する。

３．スキーマの構造

建築ＣＡＤ－ＤＢは前章で述べたように非常に自由度の高いシステムである。これはまた、データ管理のための強力な機能が必要である、ということを意味していると言える。従って、建築ＣＡＤ－ＤＢでは通常のデータベースにも増してデータディクショナリ[7,8]の役割が重要である。本章ではデータディクショナリ内のスキーマの構造について述べる。スキーマに関してもデータベースと同様、高い自由度が求められるため、スキーマ自身もまた建築ＣＡＤ－ＤＢによって構成される。

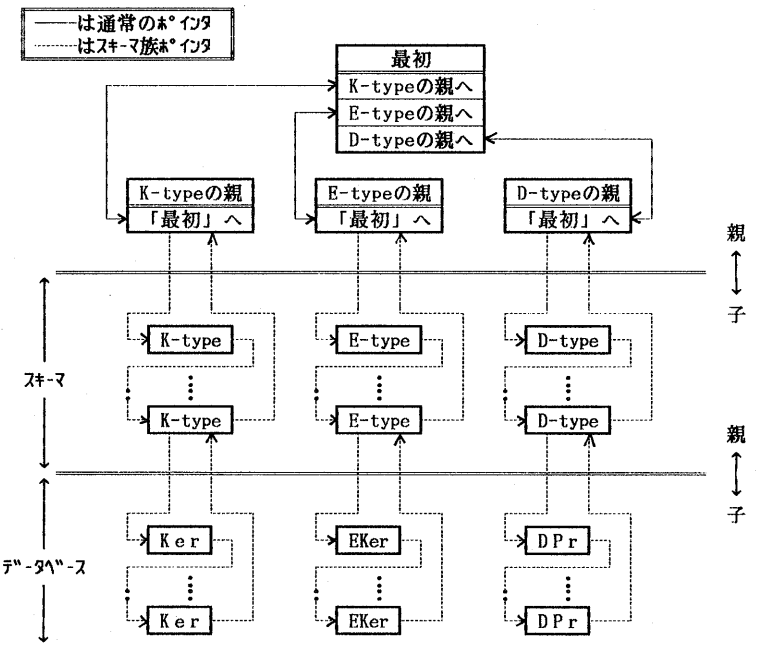

図５．スキーマとデータベースの関係

３．１　全体の構造

Ker,EKer,DPrという３種類のInformationを表現するために、スキーマにもK-type,E-type,D-typeの３種類を用意する。K-type,E-type,D-typeは、いずれもKerを用いて表現される。

データベースとスキーマの関係は図５の様になる。スキーマとそれが定義するデータとの間はスキーマ族ポインタで結ばれる。このポインタはディクショナリが自動的に結合するものであり、そのための領域は各Informationの制御部に確保されている。

「最初」および各タイプの「親」は、Kerで表わされる。これらは、データベースに最初にアクセスする際に参照される。この４つのKerは常に存在する。

３．２　各タイプの構造

K-type,E-type,D-typeはいずれも図６の様な構造を持つ。

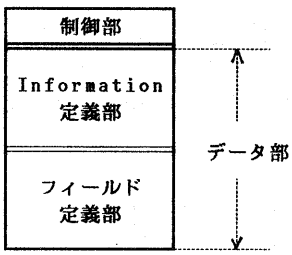

図６．スキーマの各タイプの構造

スキーマの制御部には、通常のInformationの制御部が持つ情報に加えて、K-type,E-typeおよびD-typeを区別するための識別コードが格納される。

スキーマのデータ部は「Information定義部」と「フィールド定義部」に分けられる。「Information定義部」にはスキーマが定義するInformationに関する情報が格納される。また「フィールド定義部」にはInformation のデータ部がどの様な形で使用されるかという情報（データ型の定義）が格納される。

「Information定義部」にはスキーマ名、このスキーマによって定義されるInformationがデータベース中にいくつ存在するか、Informationがもつ属性がいくつ定義されているかなどの情報が格納される。また各タイプごとに必要な情報が格納される。

「フィールド定義部」はいくつかの「フィールド定義」の並びからなる。K-type,E-type,D-typeのフィールド定義部はみな同じである。「フィールド定義」は、スキーマが定義するInformationのデータ部が、どの様な構造のデータとして扱われるかを記述する。フィールド定義では、表３に示すデータ型が定義できる。

表３．定義できるデータ型

データ型	注　釈
整数	１ワード長の整数(1ﾜｰﾄﾞ=4byte)
実数	１ワード長の実数
倍精度実数	２ワード長の実数
ｗｏｒｄ	１ワード長の領域を確保する
ポインタ	他のInformationへのポインタ
文字列	4*n文字の文字列(256文字まで)
列挙型	整数か16文字の文字列
集合型	要素は列挙型
構造体	いわゆるレコード型

現在のところ、スキーマにはデータ定義管理に最小限必要な情報のみを格納しているが、スキーマの各部分は可変長となっており、将来の拡張に容易に対応できるものである。

３．３　Information間の結合の表現

Information間の結合に関する定義情報は、その結合に関与するInformationを定義しているスキーマのフィールド定義部の中のポインタフィールド定義に格納される。ここに含まれる定義情報としては、

・結合の型（対等型or従属型）
・結合のパターン（結合先のInformation数：１orＮ）
・結合先のスキーマに対するポインタ

などがある。図７に「部屋」と「壁」の関係を定義した例を示す。

４．スキーマ構造の図形表記法

建築ＣＡＤ－ＤＢのスキーマの全体構造は、前章で示した様に、いくつものスキーマがポインタで複雑に結合し合っている、という形となっている。これはＥＲモデルと非常に類似している。従って、この様な構造を把握するにはＥＲモデルで用いられているような図形による表記法を使うのが適当であると考えられる[9,10]。そこで本章では建築ＣＡＤ－ＤＢのスキーマ構造を記述するための図形による表記法を示す。

まず、K-type,E-type,D-typeをそれぞれ図８に示すシンボルで表わす。シンボル内にはスキーマ名が記述され

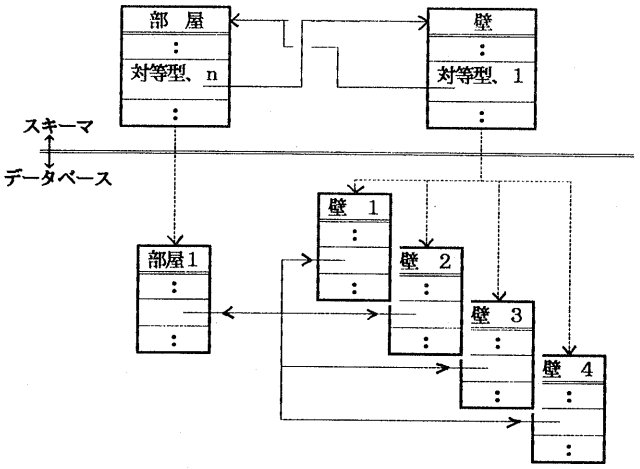

図７．部屋―壁のスキーマ構造

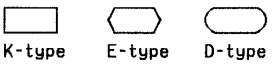

図８．各タイプのシンボル

図９．フィールド名の表記

る。また、スキーマ内に定義されたフィールド名を表記する場合は、図９の様にする。

ポインタにる結合は、図８に示したシンボルの間を結ぶ線によって表わす。これを図１０に示す。対等／従属の区別は、図１０のa),b)の様に、線に矢印をつけて表わす。矢印が両方向を向いていれば、その結合は対等型である。矢印が一方のみを指していればその結合は従属型であり、矢印の向いている側のスキーマがもう一方のスキーマに従属する。ポインタフィールド名を明示する場合は図１０のc)の様にする。

また、ポインタを表わす線には写像基数が付けられる。これは、そのポインタによっていくつのInformationが結合されるかを表わす。写像基数は１またはｎのどちらかである。例えば図１０のc)の場合、一つの柱図形には一つの点が中心点として付随し、またある一つの点はいくつかの柱図形の中心点となる、ということを表わしている。以上の表記法によって図４のスキーマ構造を記述した例を図１１に示す。

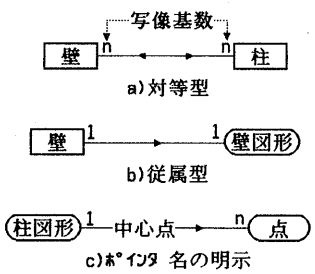

図１０．ポインタの表記法

５．スキーマ構造構築環境

建築ＣＡＤ－ＤＢのスキーマは、スキーマ定義言語を用いることにより構築される。この言語の主なコマンドは次の通りである。

表４．スキーマ定義コマンド一覧

ｺﾏﾝﾄﾞ名	機　能
DEFS	スキーマを生成し、フィールドを定義
ADDF	スキーマにフィールドを追加
CONC	２つのスキーマ間の結合を定義
CUT	スキーマ間の結合を切る
DELS	スキーマを削除
DELF	スキーマ内のフィールドを削除
SNAM	スキーマ名一覧を出力
SINF	スキーマの定義情報を出力
FNAM	スキーマ内のフィールド名一覧を出力
FINF	フィールド定義情報を出力

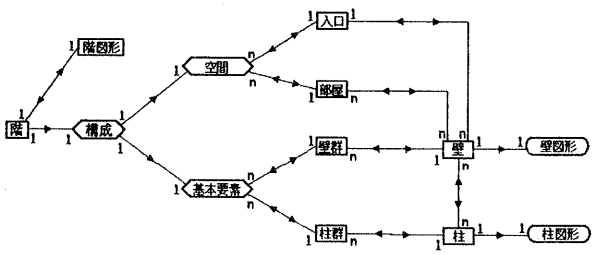

図１１．スキーマ構造の表記例

この言語によるスキーマの定義は、一見するとプログラミング言語のデータ定義文の羅列のようなものである。機能的には必要十分なものであるが、この言語による記述のみでスキーマの構造全体を把握することは非常に困難であると言えるであろう。

ところでこの言語の持つ機能は、前章で述べた図形表記により、スキーマ構造を記述する際の作業と次の様に対応付けることができる。

・スキーマを生成する(DEFS)
　→各タイプのシンボル（図８）を描く
・フィールドを定義する(DEFS,ADDF)
　→スキーマのシンボルにフィールド名を付加する（図９）
・２つのスキーマを結合する(CONC)
　→シンボルを線で結ぶ（図１０）
・結合を切る(CUT)
　→線を消す
・フィールドを削除する(DELF)
　→スキーマのシンボルに付加されたフィールド名を削除する
・スキーマを削除する(DELS)
　→シンボルを消す

また、上記以外のコマンドはスキーマに関する情報を表示するものであるが、これは表記法自体がその機能を果たしている。よって、上に示した様な操作を行うグラフィックインターフェースを用いてスキーマを構築することで、スキーマの全体構造を容易に把握することができると考えられる。

しかし、ある程度以上に大きく複雑なスキーマ構造を単純にディスプレイ上に表示しようとすると、かえってスキーマの全体像を捉えにくくするおそれがある。建築ＣＡＤ－ＤＢが対象とするのは建築物全体の記述であり、これは非常に広範かつ複雑なものである。これをディスプレイ上に一度に表示することは不可能であるし、無理に表示しようとすれば、かえってスキーマ構造の把握の妨げになると思われる。よって、グラフィックインタフェースに以上の問題を解決するための何らかの機能を持たせる必要がある。

さて、ここで建築ＣＡＤ－ＤＢのスキーマの特徴について考えてみると、次のようなことがいえる。

①スキーマ全体の構造はK-typeを主体として構築される

D-typeは属性値の定義でありK-typeで集約される。またE-typeの定義するEKerはポインタの拡張子と見なせるのでこれもK-typeの一部と言える。

②２つのスキーマの間に複数の異なる関連付けが存在し得る

例えば図１１の「階」と「壁」の間には「階－構成－基本要素－壁群－壁」、「階－構成－空間－部屋－壁」など、いくつもの関連付けが存在する。

③他のスキーマからの結合が集中するスキーマが存在する

例えば図１１では「壁」に結合が集中している。また、「点」などは図形を定義しているスキーマのほとんどすべてと結合する。

④２つのスキーマ間の関連付けの間に多くのスキーマが介在することがある

これは例えば図１１の「階」と「壁」や「柱」の間の関連付けの様なものである。実際にはさらに多くのスキーマが介在する場合もある。

上記の様な点を考慮して、スキーマ構築を支援するための機能を考察すると、次のようなものが必要と考えられる。

ｉ．注目するスキーマ（特にK-type）およびそれに直接結合しているスキーマのみを画面上に出力する。例えば、図１１の場合に壁に注目すると、図１２の様な出力が得られる。

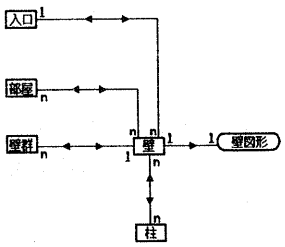

図１２．壁とその周辺

ii．現在行っている作業に不要なスキーマを画面上から一時的に消す。例えば図１１で、壁と柱に注目して作業している場合、階図形、入口、部屋などを画面上に表示しておく必要はない。

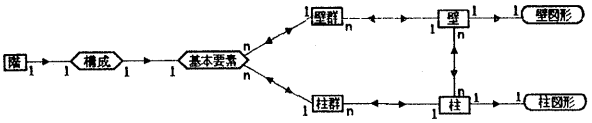

図１３．不要部分の一時的消去

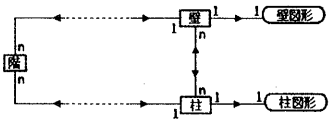

図１５．関連付けの省略記法

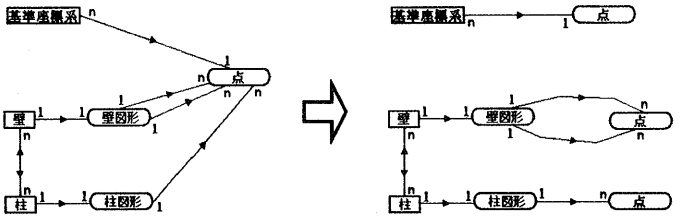

図１４．同一スキーマの複数配置

iii．結合が集中するスキーマについては、画面上に複数個配置できるようにする。これにより結合を示す線が入り乱れるのを防ぐ。これは特に点などの図形データの場合に有用な機能である。

iv．フィールド名やD-typeを表示するかしないかの選択ができるようにする。DPrは属性値を格納するものであるから、それを所有するKerに注目しているとき以外には見える必要がないことが多い。フィールド名も同様に、全体構造の把握にはさほど重要でない場合が多い。

v．注目する２つのスキーマ間に多くのスキーマを介する結合が存在する場合、間に存在するスキーマを省略するための略記法を用意する。これは特にE-typeにより深い木構造が作られた場合に有効である。

この様な機能を用意することで、スキーマ全体の中で現在着目している部分のみを見ながら処理を行なうことが出来るようになり、スキーマの全体構造をより良く把握できるようになると考えられる。

６．おわりに

本文では建築ＣＡＤ－ＤＢについて、その基本モデル、データ構造およびスキーマの構造について述べた。またスキーマ構造を構築するためのグラフィックユーザインタフェースについて、そこで用いられる表記法を示し、必要な機能について考察した。

建築ＣＡＤ－ＤＢの複雑なスキーマ構造をスムーズに構築するためには、作業者の注目している部分のみを表示し、また作業者の観点の変化に容易に対応して表示を切り換える機能が必要であるという結論を得た。

今後は、ここで述べた機能を実現し、その有効性を確認していく予定である。

参考文献

〔1〕 Eastman,C.M. : "System facilities for CAD databases", Proc. 17th ACM/IEEE Design Automation Conference, 1980, pp.50-56.

〔2〕 Katz,R.H. : "Information Management for Engineering Design", (Surveys in Computer Science) Springer-Verlag, 1985.

〔3〕 石川、宇田川他 ："建築ＣＡＤのﾃﾞｰﾀﾃﾞｨｸｼｮﾅﾘ機能"、第35回情報処理学会全国大会、3Cc-6、1987.

〔4〕 宇田川、石川他 ："建築ＣＡＤﾃﾞｰﾀﾍﾞｰｽの簡易言語"、第35回情報処理学会全国大会、4Bb-6、1987.

〔5〕 西川、伊藤他 ："建築ＣＡＤﾃﾞｰﾀﾍﾞｰｽの基本思想" 第35回情報処理学会全国大会、7Bb-3、1987.

〔6〕 "設計合理化の鍵を握るＣＡＤデータベース"、日経ＣＧ12月１日号、pp10-22、1987.

〔7〕 穂鷹 ："データベース要論"、共立出版、1978.

〔8〕 Leong-Hong,B.H. , Plagman,B.K. ; 穂鷹、成田訳 ："データディクショナリ／ディレクトリシステム、オーム社、1986.

〔9〕 Chen,P.P. : "The Entity-Relationship Model Towards A Unified View of Data", ACM Trans. Database syst. Vol.1,No.1,1976.

〔10〕酒井 ："情報資源管理の技法－ＥＲモデルによるデータベース設計－"、オーム社、1987.