

# 図形データベースの特性と管理方式の考察

木村淳美、金居貞三郎、浪岡美予子、中村史朗、大町一彦  
(日立)

## 1. はじめに

データベース(Database、以下DBと略す)の概念は、事務処理の分野で生まれ発展してきた。汎用のデータベース管理システム(Database Management System、以下、DBMSと略す)が登場し、それをういたデータベース・システムの構築が実用の段階に入ってからでも10年以上が経過した。現在、事務処理におけるDB利用技術は成熟段階にある。一方、コンピュータの適用分野の拡大につれDBのアプリケーションも、OA、CAD/CAM/CAEから統計、医療等に多様化している。これらのアプリケーションでは、従来のDBMSが対象としている文字・数値以外に、図形、画像、テキスト、音声等の多様なデータが表われる点に特徴がある。これらのデータを統一的に管理するために、マルチメディアDB管理技術の研究が活発化している<sup>1)</sup>。マルチメディアDBの一種であるエンジニアリングDBの分野では、1970年中頃から図形データの統合化が要求されている<sup>2)</sup>。

筆者らは、技術分野において設計・製造全般に渡るDBサポートを目的としたエンジニアリングDBの研究を行ってきた<sup>3,4)</sup>。この過程で、技術分野への汎用DBMS適用に対する技術課題の一つとして、設計図面に代表される図形データのDB化を取り上げ、汎用DBMSでの図形DB管理機能の実現のための研究開発を進めてきた<sup>5,6)</sup>。本報告では、従来の文字・数値中心のDBと図形DBとの特徴と問題点を考察し、それらを解決するために設定した機能インタフェースについて報告する。

本システムの方針を、以下に示す。

- (1) 図形DB管理機能を、汎用DBMSに対する拡張機能として位置付ける。
- (2) 図形処理における対話処理に耐え得る高速化を実現する。
- (3) 図形データと非図形データの統合管理の可能性を検討する。

以下、2章では、図形データDB化の要請を、3章では、その特徴とDB化における問題点を示す。4章では、その要請を満足する

ために設定した図形DB管理への機能拡張を示し、5章では、処理方式について述べる。

## 2. エンジニアリングDBの要請

エンジニアリングDBとは、設計を中心とするエンジニアリング業務を支援するための中核となるDBである。しかしながら設計業務は必然的に製造工程や生産管理と強い関連を持っており、これらを含めた広義のエンジニアリングDBシステムの理想の姿は、図2.1に示すように、一貫生産支援システムである。各生産活動で必要となるデータが、エンジニアリングDBの中で有機的に形成されることにより、情報の有効活用が可能となる<sup>7)</sup>。エンジニアリングDBに含まれるデータを、DB処理の特徴から以下の3種類に分類することができる。

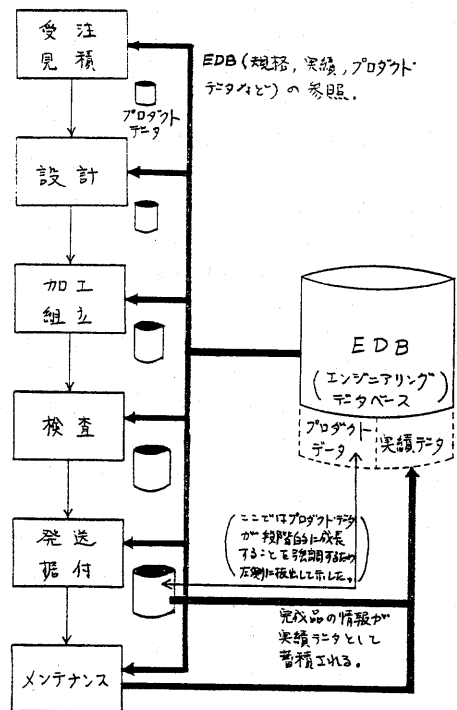


図2.1 エンジニアリングDBの段階的成長とフィードバック

(1) 図形データ

製品の形状(3次元)を表現するデータや、それを2次元平面に投影した図面情報。一般に複雑な関連を持ったデータのある塊を高速にアクセスする必要がある。

(2) エンジニアリング・データ

製品の設計諸元データ、品質管理データ、生産技術データ、規格類のデータなど、生産の各過程で参照されるデータ、および、受注仕様、設計条件や設計値など製作中の製品に関する情報である。非定型的処理要求が多く発生する。

(3) 管理データ

生産管理、原価管理などで必要となるデータである。比較的定型的なアクセスが多い。

上記データの内、エンジニアリング・データ、管理データについては、現行の汎用DBMSでサポートすることが可能である。しかしながら、図形処理は、高速性が要求されるため、専用ファイルとしてサポートされてきた。一方、エンジニアリングDBに関連して、1979年頃から論文発表が盛んになり以下のような提案がなされている。

- ・多様なデータ・タイプのサポート (13/19)
- ・コンプレックス・オブジェクト概念の設定 (12)
- ・長時間トランザクション対策 (12)
- ・エンジニアリング・データ・モデルの提案 (15~17)
- ・各種DBシステム構築の試み (17~24)

筆者らは、先にエンジニアリングDBとしての要求機能について考察した(7)。その中で、エンジニアリングDBを管理するための汎用DBMSとして、使い易さおよび柔軟性の面から、リレーショナルDB(以下、RDBと略す)の適用性が高いことを示した。また、エンジニアリングDBにおける最大の問題は、図形データのサポートであることを述べた。図形データは、次章で述べるように、エンジニアリング・データや管理データとの強い関連を持つ。したがって、エンジニアリングDBの構築は、本質的に図形データの統合を要求する。図形データ統合化の方法には、次の2通りが、考えられる。

- ・図形データ管理は専用パッケージに任せ、DBMSは主に属性データを管理する。
- ・図形データそのものをDBMSが管理する。両者の統合方法は各々特徴があり、相補的に必要と考える。本報告では、RDBを用いた、

図形データのDB管理方式について検討する。

3. 図形データの特徴とDB化における問題

3.1 図形データの特徴

(1) 図形データは、一般に以下のデータから構成されている。

a. 幾何・形状データ

図形処理パッケージの幾何モデルで扱われ、面・線・点等を表現する数値データの集合である。図3.1、図3.2にワイヤフレームモデルによる三角錐の内部表現例を示す。幾何・形状データの構造は、一般に二つのタイプに分けられる。一方は、部品や製品の組合せを表す上位概念で、図のユニット、グループが相当する。他方は、部品の位相・幾何を表し、図のワイヤフレーム、サーフェイスが相当する。幾何・形状データは、次のような特徴を持つ。

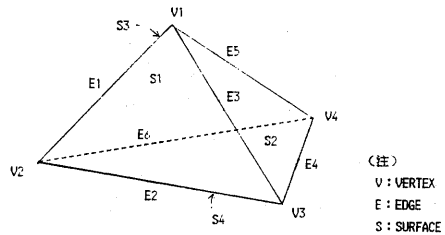


図3.1 対象モデルの例(三角錐)

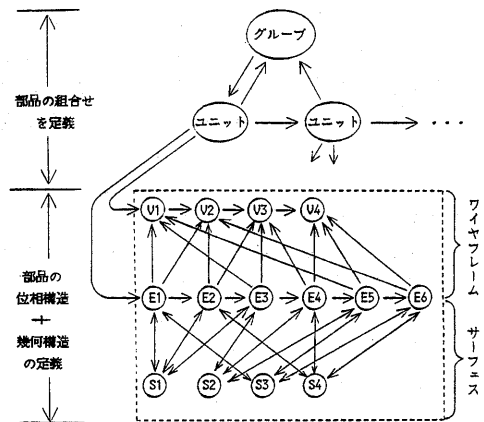


図3.2 対象モデルの内部表現例(三角錐)

i) 1 図形当りのデータ量およびアクセス量が多い。例えば、機械系 CAD システムで扱う図面では、ベクトル形式で表現する場合、A1 サイズで線数 1000 本程度、メモリ所要量数百 KB 程度の量になる。また、川越らは、3 次元図形処理における図形要素アクセス回数の例を挙げている<sup>22,23)</sup>。さらに、自由曲線等の非構造化データをもつ。

ii) 図形要素の種類が多く、データ構造が複雑である。例えば、線素には、直線の他に、円弧、楕円弧、自由曲線等があり、それぞれデータの種類・構成が異なる。

iii) 図形要素相互の関連が強く、各種の完全性制約を要求する。また、データの順序一貫がリーの表現を必要とする。

#### b. 属性データ

図面上では、注記・図面管理・来歴・製作・部品情報等として表現されている。これらの属性データは、図面中に表われるだけでなく先に述べたエンジニアリング・データや管理データとしても必要となるデータである。さらに、属性データは、導入サイト毎に、必要とする属性が異なり、属性データのカスタマイズ機能が要求される。

(2) 図形データは、次のような処理特性を持つ。

a. 大量のまとまりのあるデータ (例えば 1 図面単位) を一度に検索する。

b. 検索結果の図形データに対し、画面を通し試行錯誤的に長時間の会話処理を行う。

c. 会話処理の結果新たに出来上がった図形データを一括してエンジニアリング DB に反映する (DB 更新)。

### 3. 2 データベース化における問題点

上記のような特徴を持つ図形データを忠実に現行 RDB の機能により適用した場合、以下の問題が生じる。

#### (1) 機能上の問題

##### a. テーブルの複雑化

点と線、線と面の関係に示されるように、これらの関係は  $m:n$  関係になり、リレーションシップ・テーブルが必要になる。また、図形要素で述べたように、同じ概念の要素を、データの構成が一部異なるために、別々のテーブルとして定義することが必要になり、テーブルの構成が複雑になる。このため、操作の複雑化や処理時間の増加を招く。

#### b. 非構造化長大データ

自由曲線、自由曲面等は、点列や補間点群の塊で意味をもつ。これらのデータは、可変長の長大データになる。

#### c. 完全性制約

幾何モデルには、各種のセマンティック・インテグリティが必要とされる。これらは図形処理特有のものが多く、DBMS としてサポートすべきものもある。

#### d. 順序の表現

例えば、辺の接続方向によって面の向きが決定される。

#### f. 複合操作

部品単位、グループ単位等、階層的なアクセスが多い。RDB は、基本的にテーブル単位の操作である。

### (2) 性能上の問題

#### a. レスポンス

図形データの処理では、部品単位や図面単位等、大量のまとまりのあるデータを、一度にアクセスする。RDB のレコード単位のアクセス・レベルでは、アクセス時間の総計が会話処理として許容できないレベルに達する。

#### b. スループット

会話処理中、対象データを含むテーブルは、長時間ロックされる。このため、全体のスループットが極度に低下する恐れがある。

#### c. 障害・回復

大量のログ情報やバッファが必要になる。

## 4. 図形データベース管理への機能拡張

### 4. 1 機能拡張の考え方

始めに述べたように、本システムの図形 DB 管理への機能拡張方針を次のように設定した。

(1) 図形 DB 管理機能を汎用リレーショナル DBMS (RDB) の拡張機能と位置付ける。

(2) 図形処理における対話処理に耐え得る高速化を実現する。

(3) 図形データと非図形データの統合管理を可能とする。

上記方針を次の考え方により実現する。

#### (1) 図形データ管理の階層化

図形データの管理を、ある意味を持った塊として扱うフェーズと、塊の中身の構造を認識するフェーズというように、管理フェーズ

の階層化を行う。この方法は、データ・アブストラクション<sup>2)</sup>概念に相当し、認識レベルを分離させることにより、多様な管理を可能にする。

(2) 複合操作を含む多様な操作機能

コンプレックス・オブジェクト<sup>3)</sup>概念に相当する複合操作機能や順序・関連操作機能を持たせる。この機能により、図面・部品単位の処理の簡易化や関連付け操作を容易にする。

(3) 統合操作

ユーザから見たインタフェースを統一する。

(4) 高速化

メモリ階層化概念を適用し、主メモリの活用を計る。主メモリを高速DB領域と考え、図形・部品等の単位で常駐化する。さらに、データ間関連操作の高速化を行う。

#### 4. 2 システム構成

図4・1に、本システムの全体構成と図形DB管理拡張機能の位置付けを示す。本システムは、図に示したように3つのレベルに分かれる。基本層は、汎用リレーショナルDBMS (RDB) である。このレベルでは、材料・重量等の属性データを従来の文字・数値データと同様に、構造を認識して扱う。一方、図形データについては、構造を認識させず、非構造の長大レコードとして扱わせる。

中間層は、図形DB管理拡張機能である。このレベルは、図形データの構造を管理し、構造に従って図形データの実体を操作する。さらに、これらのデータの主メモリ内領域管理、ハッシュ、参照関係操作機能等により高速化を計る。このレベルでは、RDBに対しては、非構造データ操作インタフェースをとり、AP (アプリケーション・プログラム) に対しては、構造認識レベルのインタフェースを提供する。

応用層は、APあるいはエンド・ユーザである。このレベルでは、図形データに対してレコード単位の操作と集合操作が可能である。非図形データに関しては、RDB透過機能を持たせている。すなわち、APの操作要求は図形DB管理機能が受け取る。非図形テーブルに対するコマンドならばRDBに透過する。図形テーブルの場合には、本機能部分が要求に沿った処理を行う。

将来的には、図形テーブルと非図形テーブルの関連操作をシステム・レベルでサポートすべきである。例えば、ある材質を使用している部品を表示するという要求の場合、ひとつのコマンドで表現できると有効である。本システムでは、このような要求は、2つのコマンドに分ける必要がある。

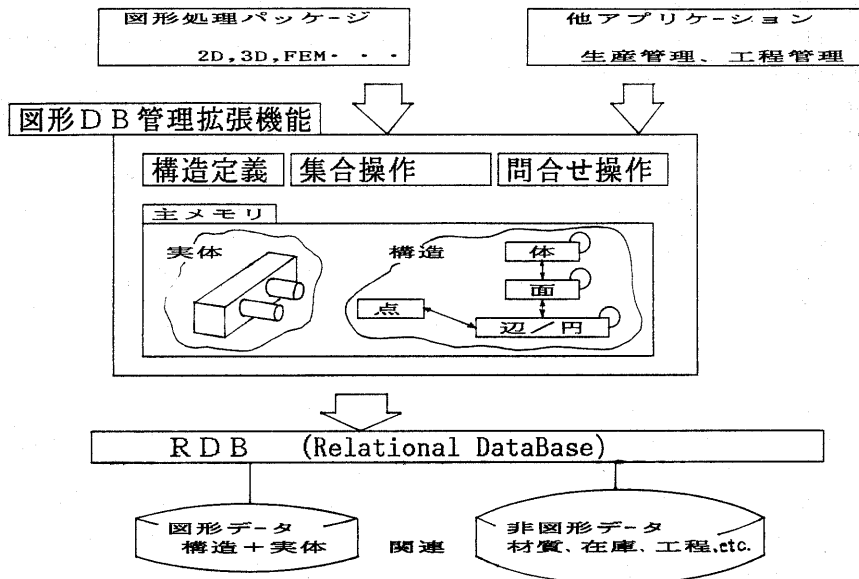


図4・1 図形DB拡張機能の位置付けとシステム構成

この他に、前章で示した問題の内、障害回復および長時間トランザクションに関しては、他の機会に譲る。

#### 4. 3 図形DB管理機能

##### (1) 図形データの構造定義

図形データの構造を定義する機能である。本機能は、図形データを格納するテーブル、テーブルを構成するカラムの属性定義、および、テーブル間の関連を明示的に表現する参照関係定義から成る。以下に図3.1の例を用いて、図形データの構造定義例を示す。

```
create table Group (Gname c 8,.....)
create table Unit (Uname c 8,.....)
create table Edge (Eid(id) i,Ename c 8,
Sp i,Ep i,.....)
create table Vertex (Vid(id) i,Vname c
8,X r,Y r,Z r,....)
create table Sface (Sid i,Sname c 8,E
i,.....)
create refer Vertex from Sp of Edge
create refer Vertex from Ep of Edge
create refer Edge from E of Sface
```

上記の例では、まず、グループ、ユニット、稜線、頂点、面のテーブルを定義した。この時に、参照関係を明示するカラムの内、被参照側のカラム（これをidカラムと呼ぶ）には、その旨の宣言を行う。“Eid”、“Vid”は、この例である。次に、テーブル間の明示的参照関係をcreate referコマンドを用いて定義した。これは、参照側のテーブルおよびカラムと、被参照側のテーブルを指示することにより行われる。これらを、各々、参照テーブル、参照カラム、被参照テーブル、被参照カラム（idカラム）と呼ぶ。実際の関連付けは、参照カラムのカラム値をidカラムのカラム値とすることで表現する。

##### (2) 集合操作

本機能は、図面単位・部品単位等の集合操作を行う以下の機能から成る。

- アーカイブ：指定単位のデータを二次記憶のDB空間から主メモリ空間に展開する。
- リストア：指定単位のデータを主メモリ空間から二次記憶のDB空間に格納する。
- コピー：指定された単位のデータを二次記憶のDB空間間で複写する。

d. スクラッチ：指定された単位のデータを二次記憶のDB空間から削除する。

本機能によりAP操作の単純化を計る。

##### (3) レコード単位操作

集合操作が複合レコード単位の操作であるのに対し、本機能は、条件指示、カーソル指示による単一レコード操作が基本である。

###### a. 条件指示によるレコード操作

比較演算、論理演算による条件を指示し、条件に合致したレコード群を対象に、検索・更新・挿入・削除を実行する。

###### b. カーソル指示によるレコード操作

カレントポジションを示すカーソルに基づき、検索・更新・挿入・削除を実行する。明示的参照関係を操作するために、次の3種類のカーソルを準備している。

(i) 主カーソル：操作を行うテーブルに設定するカーソルである。以下のカーソル使用に先立ち、必ず設定する必要がある。

(ii) 順参照カーソル：主カーソルが設定されているテーブルから被参照テーブルに対して設定する。このカーソルにより、参照テーブルから被参照テーブルへの関連操作を可能にする。

(iii) 逆参照カーソル：主カーソルが設定されているテーブルから参照テーブルに対して設定する。本カーソルは、被参照テーブルから参照テーブルへの関連操作を可能にする。

上記の操作をAPから使用するためのインタフェースとして、コマンド・セットアップ、オープン、フェッチ、クローズのオペレーションを準備している。以下に、これらの機能を用いた操作例を示す。

例1 辺“E1”の頂点座標を検索する。

```
setup cursor E select Sp,Ep from
edge where Ename="E1"
setup cursor V with E select X,Y,Z
from Vertex
open E
open V
do until record not found
fetch V
.....
end
close V
close E
```

例2 辺"E1"の属する面を検索する。

```
setup cursor E select Eid from Edge
  where Ename="E1"
setup cursor S with E select Sname
  from Sface
open E
open S
do until record not found
  fetch S
  .....
end
close S
close E
```

例3 辺"E1"の名称を変更する。

```
update Edge set Ename="E2" where
  Ename="E1"
あるいは、
update with E set Ename="E2"
```

## 5. 図形DB管理機能における処理方式

本章では、図形データの構造管理と高速処理の解決のために、本システムで採用した処理方式上の工夫について述べる。

### 5.1 テーブル管理

先に述べたように、図形データ処理の特徴は、大量データの一括アクセスと更新にある。製図の場合を考えると、最終図面が出来上がるまでは、更新の連続である。したがって、処理対象データの高速アクセスおよび更新オーバーヘッドの削減が必要である。

本システムは、このため、対象図形データベースを主メモリで管理する。この方法により、二次記憶装置とのアクセス・オーバーヘッドを削減する。さらに、主メモリ内の論理ページ管理および論理ページ内レコードのロット管理により、Null値や更新によるフラグメンテーションの削減および、更新によるレコード・アドレス変更の影響を排除する。この方法では、主メモリを、あるサイズを単位とする論理ページに分割する。論理ページは、格納するデータの特性によって、図形データそのものを格納する実レコード論理ページ、および後述する仮想レコード論理ページ、ハッシュ表論理ページにグループ化して管理する。ページ内の各レコードは、ページに設けられたロットを介してポイントされるスロ

ット方式により管理する。

この主メモリ・テーブル管理方式により、データを論理的なアドレスで管理することが可能である。また、データ更新によるフラグメンテーションおよびアドレス変更の影響を排除できる。

これら論理ページ群は、RDBの長可変長データ型を用いて格納し、主メモリ展開・格納時のアクセス回数削減を計っている。

### 5.2 明示的参照関係管理

図形テーブル間の関連操作手段には、カラムの値による結合方法と、ここで述べる、ポインタを用いた明示的参照関係による方法がある。図5.1に、4.2節の定義に基づく、テーブルの実現値の例を示す。参照関係の操作とは、頂点"V2"を始点とする稜線の検索("E1","E4","E5","E7"が相当する)、あるいは、稜線"E9"の始点座標の検索("V4"が相当)等の関連に基づく操作を示す。図形データの場合、データ相互の関連が強かつ複雑であるため、カラム値による結合だけでは、サーチの占める時間が無視できなくなる。このサーチ時間を削減するため、明示的参照関係を定義したテーブルには、ポインタによる関連付けとハッシングを導入する。

#### (1) ポインタ

図5.2に示すように、被参照レコードから参照レコードに対する被参照ポインタ、および、関連する参照カラム間を関係付ける兄弟ポインタから構成する。このポインタにより、被参照テーブルと参照テーブル間の高速アクセスを可能にする。

#### (2) ハッシング

明示的参照関係を定義したテーブルでは、被参照テーブルから参照テーブルを参照するケースが多い。したがって、被参照テーブルのサーチを高速化することが必要になる。このため、被参照テーブルのidカラム値とレコードidをもとにしたハッシュ表を設け、高速化を計る。このハッシュ表により、参照カラムから被参照テーブルに対する逆参照も高速化することが可能になる。

#### (3) 仮想レコード

テーブル間の明示的参照関係管理により、図形データ操作を効率良く行なうことができる。しかし、明示的参照関係の導入により完全性制約、特にリファレンシャル・インテグ

レティを満たさない関係は、存在できなくなる。例えば、ある稜線が参照する頂点のデータが存在しない等、参照関係に不整合がある場合、図5.2に示すような参照関係のポインタ・チェーンを形成することができない。しかしながら、図形データ処理の途中段階では、このような不整合状態が生じる。したがって、このような状態にある参照関係の取扱いが問題となる。

本システムでは、この問題を解決するため、仮想レコードの概念を導入する。仮想レコードは、参照レコードを管理するための被参照ポインタのみを持つレコードである。仮想レコードは、以下のような場合に生成される。

- ・参照レコードを持つ被参照レコードが削除された場合。
  - ・参照カラムを持つレコードが挿入されたが、被参照レコードが存在しない場合。
- また次のような場合、仮想ローは消滅する。
- ・参照レコードが全て削除された場合。
  - ・被参照レコードが挿入された場合。
- 仮想レコードの導入により、完全性制約条件の管理が可能になる。また被参照レコードの挿入、削除の時のポインタ・チェーンの修正は、仮想レコードと被参照レコードの間の被参照ポインタの移動のみですむ等の利点が得られる。

被参照側

頂点テーブル					
頂点ID	頂点名	X座標	Y座標	Z座標	...
1	V1	25.0	25.0	55.0	
2	V2	0.0	50.0	0.0	
3	V3	0.0	0.0	0.0	
4	V4	50.0	0.0	0.0	
5	V5	25.0	25.0	-55.0	

参照側

稜線テーブル				
稜線ID	稜線名	始点	終点	...
1	E1	2	1	
2	E2	3	1	
3	E3	4	1	
4	E4	2	4	
5	E5	2	3	
6	E6	3	4	
7	E7	2	5	
8	E8	5	3	
9	E9	5	4	

操作(検索)

- (1) 1テーブル内の検索
- (2) 被参照側テーブルを元に参照テーブルを検索 (頂点V2を始点とする稜線)
- (3) 参照側テーブルを元に被参照テーブルを検索 (稜線E9の始点)

図5.1 テーブル実現値と参照関係の例

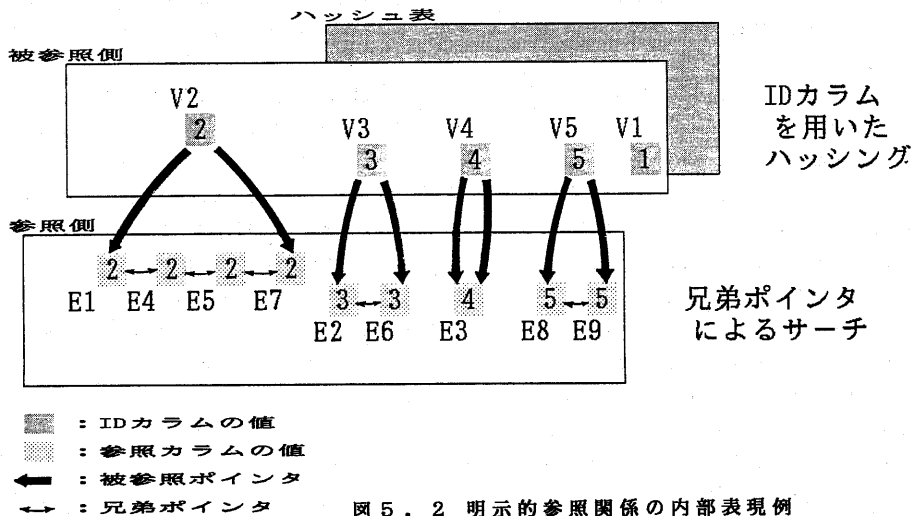


図5.2 明示的参照関係の内部表現例

## 6. おわりに

エンジニアリングDBにおいては、図形データのDB化が、統合管理のための重要な課題である。本報告では、図形データの構造・処理上の特性を考察し、筆者らが開発した図形DB管理拡張機能について述べた。現在、本システムの図形処理パッケージへの適用実験を進めている。本報告で示した方式は、図形データと同様な特徴を持つデータ、例えば文書データ等のDB化にも有効と考えらる。

### — 参考文献 —

- 1) データベース・システムに関する調査—マルチメディアデータベースの調査—、日本電子工業振興協会 60-C-508(Mar. 1985).
- 2) エンジニアリング・データベース・システム調査報告書、情報処理振興事業協会(Mar. 1975, 1976, 1977).
- 3) Nakamura, F. et al.: Design Document Generation from Engineering Databases, Proc. of Inter Graphics '83 (1983).
- 4) 中村、高西、新井: エンジニアリング・データベースの動向、日立評論、Vol.65, No.3 (Mar. 1983).
- 5) 金居他: 図形データベース管理システムの開発(1)—システムの概要—、情報処理学会第31回全国大会講演論文集(Sep. 1985).
- 6) 浪岡他: 図形データベース管理システムの開発(2)—処理方式—、情報処理学会第31回全国大会講演論文集(Sep. 1985).
- 7) 中村: エンジニアリング・データベース、情報処理、Vol.25, No.4(Apr. 1984).
- 8) Computer-Aided Design, Vol.11, No.3(CADデータベース特集号)(May 1979).
- 9) Encaracao, J. and Krause, F.L.(ed.): File Structures and Data Bases for CAD(Proc. of IFIP WG 5.2 Working Conf. 1981), North-Holland Pub. Co. (1982).
- 10) Proc. of Engineering Design Applications in Database Week Conf., IEEE Computer Society Press (May 1983).
- 11) Eastman, C.M.: Database Facilities for Engineering Design, Proc. of the IEEE, Vol. 69, No. 10 (Oct. 1981).
- 12) Haskin, R.L. and Lorie, R.A.: On Extending the Functions of a Relational Database System, Proc. of SIGMOD Conf. (1982).
- 13) Lorie, R. and Plouffe, W.: Complex Objects and Their Use in Design Transactions(参考文献10)に収録).
- 14) Johnson, H.R. et al.: A DBMS Facility for Handling Structured Engineering Entities(参考文献10)に収録).
- 15) Managaki, M.: Multi-Layered Database Architecture for CAD/CAM Systems(参考文献9)に収録).
- 16) 大保他: 柔構造関係データベース、情報処理学会データベース管理システム研究会資料30(May 1982).
- 17) Meier, A. and Lorie, R.A.: A Surrogate Concept for Engineering Database, Proc. of VLDB Conf.(1983).
- 18) Fischer, W.E.: PHIDAS-A Database Management System for CAD/CAM Application Software (参考文献8)に収録).
- 19) 宇田川、溝口: アドバンスト・データベースADAMのデータ定義言語について—エンジニアリング・データ管理へのアプローチ—、情報処理学会データベース・システム研究会資料35 (May 1983).
- 20) Ulfsby, S. et al.: TORNADO: A DBMS for CAD/CAM Systems(参考文献9)に収録).
- 21) Becerril, J.L. et al.: GSYSR: A Relational Database Interface for Graphics
- 22) 川越他: CAD用データベース管理システム、情報処理学会データベース・システム研究会資料42(July, 1984).
- 23) 川越他: 3次元形状処理システムの構想と実現、情報処理学会グラフィックスとCAD研究会資料46(June, 1984).
- 24) 宇田川、溝口: 先進的データベースADAMの実現、情報処理学会データベース・システム研究会資料46(Mar. 1985).
- 25) Smith, J.M. and Smith, D.C.P.: Database Abstraction: Aggregation and Generalization, ACM TODS, Vol. 2, No. 2 (June 1977).