

# マルチメディア・データベース に向けたデータモデルについて

村田達彦 寺中勝美 鈴木健司 石垣昭一郎 岸本義一

(日本電信電話公社 横須賀電気通信研究所)

## 1. はじめに

近年のデータベースの発展に伴い、データベース化するデータも多様化しており、従来の数値・文字データに加えて図形・画像データも統一的に管理したいという要求が高まっている。マルチメディア・データベース管理システムはこのような要求に応えることを目的としたデータベース管理システム(以後DBMSと略す)であり、図面とその設備管理を行なう計算機支援によるCAE, 図形・画像情報を含む情報検索, 文書処理を主体とするOA, コンピュータ・マッピング等の分野への適用をねらいとしている。

マルチメディア・データベース(以後MDBと略す)では従来のデータベースに比べ、データベースで扱う対象が大きく、複雑な関係が存在するという特徴がある。このような対象を従来のDBMSで扱うには対象の表現能力が十分ではなく、利用者の操作性およびアクセス時の性能が問題となる。

本稿では上記特徴を有する対象を表現する上で必要となる基本的な概念をデータモデルに導入し、MDBに対する利用者の操作性の向上, 処理の効率化を図ることをねらいとする新しいデータモデルCERA(Complex Entity with Relationship and Attribute)を提案する。

2章では、既存のデータモデルの問題点およびデータモデルへの新たな要求条件を、3章では、CERAモデルの概要とCERAモデルの最大の特徴である複合対象の表現方法について述べる。また4章と5章では、新しい概念を中心にCERAモデルのデータ定義機能, データ操作機能についてそれぞれ述べる。

## 2. 既存データモデルの問題点とデータモデルへの新たな要求条件

MDBでは複雑な対象を扱うため、データベース機能に対する要求も多様化, 高度化している。以下ではMDBの一分野であるエンジニアリング・データベースを例に新たな要求条件とそれに対する既存データモデルの問題点を述べる。

### (1) 複合対象の表現と操作

エンジニアリング・データベースにおける図面処理等では、1つの図面が様々な図形要素の集合として構成されており(図2-1参照), 設計業務では図面内の個々の要素を単位とする操作に加えて、例えば図面全体の検索, 移動, 削除のような、図面という単位で全体をまとめて操作, 管理する機能が必要である[1]。

現実世界の事物に対応するモデル化の要素を対象(エンティティと呼ぶ。既存の階層モデル, ネットワーク・モデル, リレーショナル・モデルでは、対象をレコード/リレーションとして表現している。しかしこれらのデータ

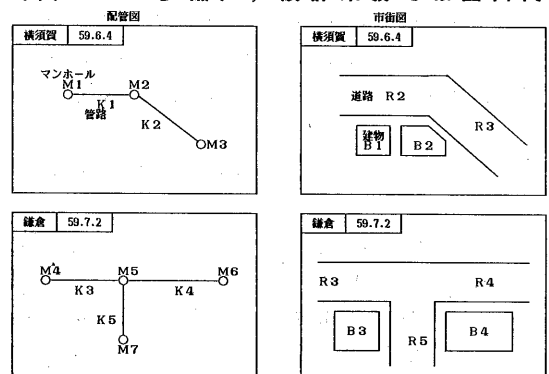


図2-1 地下配管図の例

モデルでは図面のような複数の異なる対象の集合として構成される対象 --- これを複合対象 (Complex Entity) と呼ぶ --- を表現する概念が存在しない。このため複合対象を利用者が扱う場合には、複合対象とそれを構成する対象間の関係を利用者が自分で管理する必要があり、以下のような操作、制御上の問題がある。

① 複合対象全体を操作する場合でも、利用者は単一の対象に対する操作命令を繰り返し発行しなければならず、操作性が非常に悪い。

② 複合対象を扱う上で最適なアクセス制御、格納制御、排他制御等の実現が不可能であり、次のような問題がある。

(i) ある複合対象内のデータだけを操作する場合でもアクセスする度に全体のデータの中からアクセス対象を探す必要があるため性能が悪い。

(ii) 複合対象を構成するデータ集合を近接配置することができず、その結果2次記憶媒体へのI/O回数が増加する。

(iii) 複合対象を単位とする排他制御ができないため、不必要なデータをロックすることによるコンカレンシの低下や、あるいは必要データをその都度ロックすることによるオーバーヘッドが生ずる。

## (2) 対象間の複雑な関係の表現

図面を考えた場合、そこに表現されている図形要素間には様々な複雑な関係が存在する。これらの関係を操作、管理するためには、複数の対象間の関係 (n 項関係) および2つの対象間の様々な関係 (n : m 関係) をデータモデルとして表現できる必要がある。例えば図2-1の例では、マンホールと管路との間に”接続”という n : m の関係がある。

しかし従来のデータモデルでは、関係を陽に定義できないリレーショナル・モデルは当然として、ネットワーク・モデルでも高々2項関係かつ 1 : n 関係までしか表現できず、それ以上の複雑な関係は利用者の管理に委ねられている。

## (3) 同一対象の多面的表現

従来の事務処理等の分野では、データベース化の対象は比較的単純で、データベース内でも1通りの方法で表現されれば十分であった。しかしエンジニアリング・データベースでは1つの対象をさらに詳細に分類し、分類されたそれぞれの対象も管理する必要がある。例えば建物の例では、図面設計業務では建物という対象で一括して管理するだけでよいが、管理業務においては建物をさらに分類して住宅、商店、会社という対象で管理する必要がある。

従来のデータモデルでは、レコードやリレーションで表現される対象はそれぞれ全く別な対象としてDBMSが管理している。従って同一対象であるにもかかわらず同一対象としてDBMSが管理できないために更新の同期等を利用者が行う必要がある、という問題がある。

上記(1)～(3)をまとめると以下に示す新たな要求条件を満たすデータモデルが必要である。

(A) 複合対象をデータモデル上で陽に定義可能とし、かつ複合対象を単位とする操作、制御、管理を可能とする機能を提供すること。

(B) 対象間の複雑な関係の表現を可能とすること。

(C) 現実世界での1つの対象に対して、複数の観点からの多面的な表現を可能とすること。

### 3. CERAモデル

#### 3.1 概要

CERAモデルの概要を以下に示す(表3-1参照)。

①対象および複合対象をそれぞれテーブルおよびグループという概念で表現する。

②対象間の複雑な関係を表現するために、実体-関連-属性モデル(E-R-Aモデル)[2]の概念を導入する。

③同一の対象に対する多面的表現を可能とするため、汎化階層[3]の概念を導入する。

また、

④上記概念に対して、高水準な操作機能を提供する。


図3-1は、図2-1の地下配管図面をCERAモデルのデータ構造によって表現したものである。図中で  の記号がグループを表現している。オカレンスの例を図3-2に示す。

表3-1 CERAモデルの概要

CERAモデル		ネットワーク/ 階層モデル	リレーショナル モデル
対象	テーブル	対象の最小管理単位の表現	レコード
	グループ	複数の異なる対象(テーブル)より大きな対象の表現	なし
	スーパータイプ	類似する複数のテーブルの共通部分を抽出して1つにまとめたテーブル(汎化)	なし
	サブタイプ	あるテーブルをさらに分類してより詳細な情報を追加したテーブル(専化)	なし
関連	対象間の関係(n項関連)の表現	セット(2項)	なし
属性	イメージ	画像の2次元配列表現	なし
	テキスト	長大可変長文字列表現	なし
	図形	図形要素の表現	なし
	数値	数値の表現	数値
	文字	文字の表現	文字

データ構造図

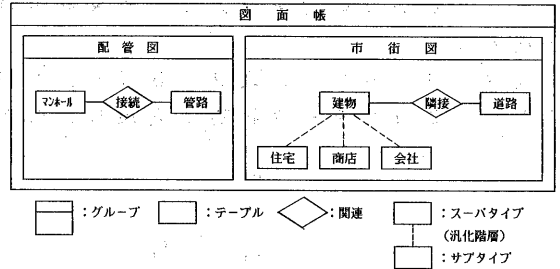


図3-1 地下配管図データベースの構成例

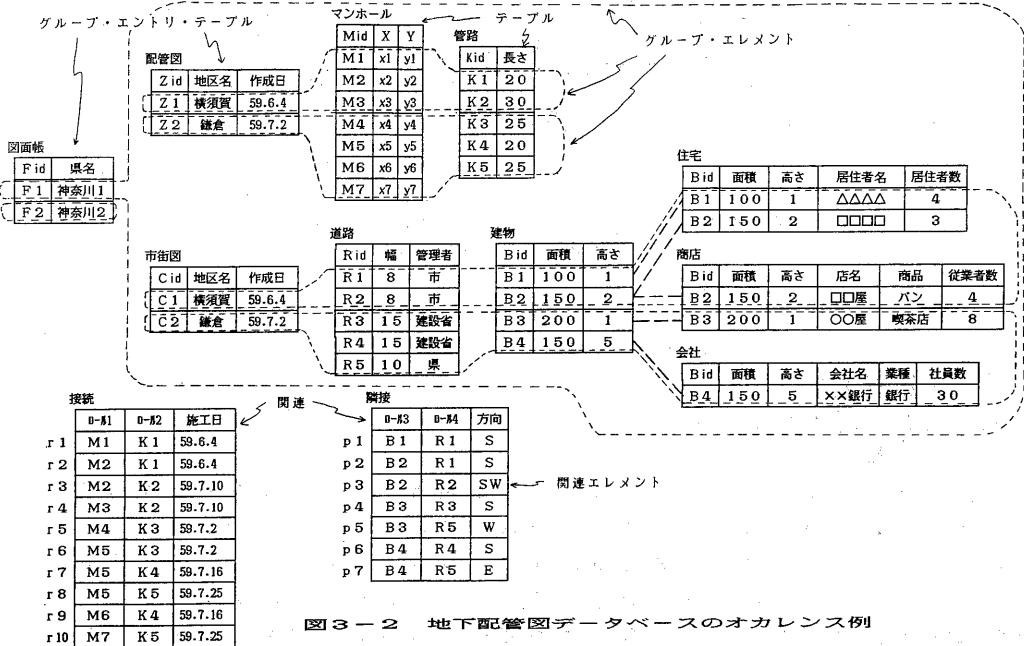


図3-2 地下配管図データベースのオカレンス例

### 3.2 複合対象の表現

CERAモデルの特徴的な概念の1つである複合対象について、データモデル上の表現方法を述べる。

現実世界は一般に、対象とそれら対象から構成される複合対象、および複数の複合対象から構成されるより大きい複合対象で表現できる。これら対象と複合対象および複合対象とより大きい複合対象の間には、上位の対象が下位の対象を”包含している”という自然な関係が存在する。この自然な関係を包含関係と呼ぶ。

例えば、図3-3(a)は、図2-1に示した地下配管図面を包含関係(⇒)で示す)に基づいて表現したものである。この例では図面帳という複合対象は配管図と市街図という複合対象を包含しており、さらに配管図はマンホールと管路という対象を包含し、市街図は建物と道路という対象を包含している。

CERAモデルでは上記の包含関係に基づいて複合対象を表現するために”グループ”という概念を導入し、対象を表現する概念である”テーブル”と明確に区別する。

図3-3(b)は、図3-3(a)と対応するCERAモデルでの概念を示しており、これをデータ構造で表現したのが図3-1である。

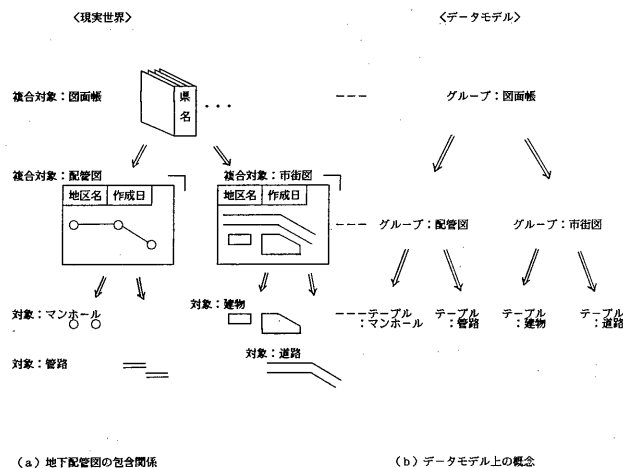


図3-3 包含関係による複合対象の表現

## 4. データ定義機能

### 4.1 グループとテーブル

#### 4.1.1 概念

グループは包含関係に基づいて複合対象を表現する概念である。従ってグループは、より小さい複数のグループあるいは複数のテーブルを包含する。

グループ、テーブルの個々の実現値(オカレンス)を表すものを、それぞれグループ・エレメント、テーブルの行と呼ぶ。また、個々のグループ・エレメントの識別等のために、グループ自身が持つ属性情報等から構成されるテーブルを、グループ・エントリ・テーブルと呼び、その実現値をグループ・エントリ・テーブルの行と呼ぶ。

#### 4. 1. 2 定義機能

図4-1は図3-1のデータ構造を定義した例である。本図において、①～④のステートメントでテーブルの定義を、⑤～⑩でグループの定義を行なっている。グループの定義においてはCONTAIN句によってそのグループが包含するテーブルまたはグループを定義する。このCONTAIN句によりグループ間およびグループとテーブル間の包含関係の定義が可能になる。例えば⑥は配管図グループがマンホール・テーブルと管路テーブルを包含していることを表しており、⑩は図面帳グループが配管図グループと市街図グループを包含していることを表している。

#### 4. 2 関連

##### 4. 2. 1 概念

関連は対象間の関係を複数の対象の組として表現したものであり、対象のn個組情報および関連自身の属性情報により構成される。関係づけられた個々の関連を関連エレメント、関連の属性情報によって構成されるテーブルを関連テーブルと呼ぶ。関連を構成するいくつかの概念について以下に述べる。

- (1) ロール : 関連に関与するテーブルがその関連の中で果している役割を明確に区別するための識別子をロールと呼ぶ。特に関連がループを形成している場合には、ロール名によってテーブルの関連への関与のしかたを識別することが必須である。
- (2) マッピング制約 : 一般に任意の2つの対象間に定義された関連は、1対1, 1対n, n対mの3つのパターンに分けることができる。CERAモデルではこれをn項関係に拡張して関連の任意の2つのロールの組に対して、1対1, 1対n, n対mの対応関係があることを明示的に記述可能とする。

##### 4. 2. 2 定義機能

図4-1の⑪～⑬が関連の定義である。ON句に続く( )内がロールの定義である。またMAPPING句でロール間の対応関係を定義する。

#### 4. 3 汎化階層

##### 4. 3. 1 概念

同一の対象物に対する多面的な表現を可能とする概念である。複数の対象を抽象化して統合した1つの対象としてとらえることを汎化、逆にある対象を細分化

(テーブルの定義)

```
①CREATE TABLE マンホール  
(Mid CHAR(2), X INTEGER, Y INTEGER)  
IDENTIFIED BY Mid
```

```
②CREATE TABLE 管路  
(Kid CHAR(2), 長さ INTEGER)  
IDENTIFIED BY Kid
```

```
③CREATE TABLE 建物  
(Bid CHAR(2), 面積 INTEGER, 高さ INTEGER)  
IDENTIFIED BY Bid
```

```
④CREATE TABLE 道路  
(Rid CHAR(2), 幅 INTEGER, 管理者 NCHAR(8))  
IDENTIFIED BY Rid
```

(汎化階層の定義)

```
⑤CREATE TABLE 住宅 SUBTYPE OF 建物  
(居住者氏名 NCHAR(10), 居住者数 INTEGER)
```

```
⑥CREATE TABLE 商店 SUBTYPE OF 建物  
(店名 NCHAR(10), 商品 NCHAR(10), 従業員数 INTEGER)
```

```
⑦CREATE TABLE 会社 SUBTYPE OF 建物  
(会社名 NCHAR(19), 業種 NCHAR(10), 従業員数 INTEGER)
```

(グループの定義)

```
⑧CREATE GROUP 配管図  
(Zid CHAR(2), 地区名 NCHAR(5), 作成日 CHAR(8))  
IDENTIFIED BY Zid  
CONTAIN TABLE マンホール, 管路
```

```
⑨CREATE GROUP 市街図  
(Cid CHAR(2), 地区名 NCHAR(5), 作成日 CHAR(8))  
IDENTIFIED BY Cid  
CONTAIN TABLE 建物, 道路
```

```
⑩CREATE GROUP 図面帳  
(Fid CHAR(2), 図名 NCHAR(5))  
IDENTIFIED BY Fid  
CONTAIN GROUP 配管図, 市街図
```

(関連の定義)

```
⑪CREATE RELATIONSHIP 接続 (施工日 CHAR(8))  
ON (ロール1 WITH マンホール),  
(ロール2 WITH 管路)  
MAPPING (ロール1, ロール2 N:M)
```

```
⑫CREATE RELATIONSHIP 隣接 (方向 NCHAR(2))  
ON (ロール3 WITH 建物),  
(ロール4 WITH 道路)  
MAPPING (ロール3, ロール4 N:M)
```

図4-1 地下配管図データベースの定義例

して別の表現でとらえることを専化と呼び、これらの関係を汎化階層と呼ぶ。また汎化されたものを表すテーブルをスーパータイプ、専化されたものを表すテーブルをサブタイプと呼ぶ（図4-2参照）。

サブタイプのある行が存在するとき、これに対応するスーパータイプの行も存在する。しかし、スーパータイプの行が存在しても、これに対応するサブタイプの行は存在しないこともある。つまり、住宅、商店、会社といった分類がまだ決っていない建物の存在を表現可能とする。サブタイプはスーパータイプを詳細化したものであるから、スーパータイプの性質は全てのサブタイプに遺伝する。このことから、サブタイプの属性情報の中には、スーパータイプの属性情報が全て含まれる。これを共通属性と呼ぶ。同一の対象に対するスーパータイプとサブタイプの間では、共通属性の値は一致している。

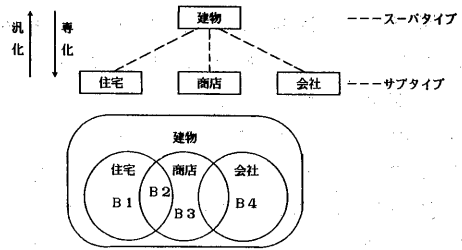


図4-2 汎化階層の概念

#### 4.3.2 定義機能

図4-1の⑤～⑦がサブタイプのテーブルの定義である。スーパータイプとなるテーブルのテーブル名（建物）およびそれぞれのサブタイプ固有の属性情報を定義する。スーパータイプのテーブルの定義（③）は通常のテーブルの定義と同様であるが、スーパータイプとサブタイプが同一の対象であることを識別するためのキーの指定が必須である。

### 5. データ操作機能

#### 5.1 グループ操作

グループに対する操作には以下の2つがある。

- (1) グループ・エレメントに包含されているテーブルの操作
- (2) グループ・エレメントの操作

##### 5.1.1 グループ・エレメントに包含されているテーブルの操作

図面の設計処理では大量の図面の中から1つの図面を選びだし、その図面内のデータを繰り返し操作する。そこであるグループ・エレメントを特定したとき、その特定されたグループ・エレメント内に閉じたテーブル操作が可能となる機能を導入する。

<例1> 横須賀の配管図の中のマンホールと管路を求める。

```

START GROUP  図面帳  WITHIN Fid="F1"
START GROUP  配管図  WITHIN Zid="Z1"
  SELECT Mid FROM マンホール      --- M1,M2,M3が求まる
  SELECT Zid FROM 管路          --- K1,K2が求まる
END GROUP
END GROUP
  
```

START GROUP ステートメントによって以降に発行する操作ステートメントでアクセスする対象の範囲が、指定した特定のグループ・エレメント内に限定される。この範囲はEND GROUP ステートメントまで有効である。このようにグループ・エレメントをあらかじめ特定することによって、個々の操作ステートメントでグループ・エレメントを特定する必要がなくなるため操作性が向上するとともに、アクセス制御上も指定されたグループ・エレメント内だけをサーチの対象とすることができ、処理効率の向上が可能となる。

### 5. 1. 2 グループ・エレメントの操作

グループ・エレメントに包含されるデータを一括して操作する機能である。グループ・エレメント全体を操作する場合に、包含される個々のデータの操作を繰り返す必要がなくなり、操作性の向上を図るものである。

#### (1) グループ・エレメントの作成／削除

グループ・エレメントの作成／削除は、そのグループ・エレメントの識別情報、即ちグループ・エントリ・テーブルの行を格納／削除することにより行なう。

〈例2〉 逗子の配管図を作成し、マンホールM8, M9, 管路K6を格納する(図5-1参照)。

INSERT GROUP INTO 配管図 (Zid, 地区名, 作成日)

VALUES ("Z3", "逗子", "59.8.4")

INSERT INTO マンホール (Mid, ...) VALUES ("M8", ...) WITHIN Zid="Z3"

INSERT INTO マンホール (Mid, ...) VALUES ("M9", ...) WITHIN Zid="Z3"

INSERT INTO 管路 (Kid, ...) VALUES ("K6", ...) WITHIN Zid="Z3"

〈例3〉 例2で作成した逗子の配管図を削除する。この結果はもとの図3-2になる。

DELETE GROUP FROM 配管図 WHERE Zid="Z3"

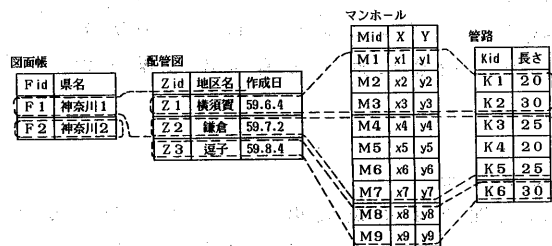


図5-1 グループ・エレメントの作成の例(例2)

#### (2) グループの包含関係の操作

グループの包含関係の操作には、あるグループ・エレメントと上位のグループ・エレメントとの包含関係を確立／解除／組み換える機能がある。

〈例4〉 返子の配管図を神奈川1の図面帳に入れて管理する。

```
INCLUDE GROUP 配管図 WHERE Zid="Z3"
TO 図面帳 WITHIN Fid="F1"
```

〈例5〉 返子の配管図を神奈川2の図面帳で管理するように変更する。

```
CHANGE GROUP 配管図 WHERE Zid="Z3"
FROM 図面帳 WITHIN Fid="F1"
TO 図面帳 WITHIN Fid="F2"
```

〈例6〉 返子の配管図を神奈川2の管理からはずす。

```
EXCLUDE GROUP 配管図 WHERE Zid="Z3"
FROM 図面帳 WITHIN Fid="F2"
```

以上の例を図5-2に示す。例6の結果は図5-2の(a)にもどる。

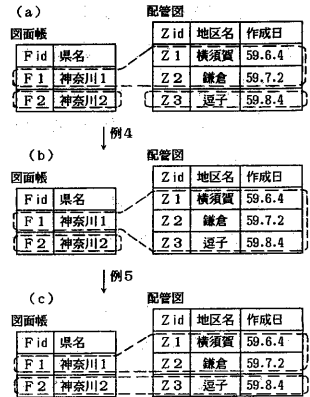


図5-2 グループ・エレメントの包含関係の確立/組み換え/解除の例(例4~例6)

## 5.2 テーブル操作

テーブルの操作機能のうち関連によるテーブルのアクセス方法について述べる。

### 5.2.1 関連関数

n項関係, n:m関係を扱うために, 関連関数と呼ぶ概念を導入する。関連関数の例を以下に示す。

RELPRJ (接続, マンホール, 管路)

このように関連関数は, 関連付けられたテーブルを関連により結合して一括した大きなテーブルとして扱う機能であり, 関連関数で作成されたテーブルに対して制限, 射影演算を可能とすることにより, 関連に対する高水準な操作機能が提供できる。

RELPRJ (接続, マンホール, 管路)						
	Mid	X	Y	Kid	長さ	施工日
r1	M1	x1	y1	K1	20	59.6.4
r2	M2	x2	y2	K1	20	59.6.4
r3	M2	x2	y2	K2	30	59.7.10
r4	M3	x3	y3	K2	30	59.7.10
r5	M4	x4	y4	K3	25	59.7.2
r6	M5	x5	y5	K3	25	59.7.2
r7	M5	x5	y5	K4	20	59.7.16
r8	M5	x5	y5	K5	25	59.7.25
r9	M6	x6	y6	K4	20	59.7.16
r10	M7	x7	y7	K5	25	59.7.25

図5-3 関連関数の例

### 5.2.2 関連によるアクセス

関連関数によって関連に関与する複数のテーブルが結合した1つのテーブルが作成されるので, これを用いて関連によるアクセスを行なうことができる。

〈例7〉 マンホールM2に接続されている管路を求める。

```
SELECT Kid FROM RELPRJ (接続, マンホール, 管路)
WHERE Mid="M2"          --- K1, K2が求まる
```



### 5.3 関連操作

関連操作は、関連エレメントの作成、関連エレメントのロールの切り離し、ロールの組み換えを行なう機能である。

〈例8〉 逗子の配管図に、マンホールM8、M9が格納されている。これに管路K6を格納し、M8とM9と接続する（図5-4参照）。

```
INSERT INTO 管路 (Kid, ...) VALUES ("K6", ...)
      WITHIN Zid = "Z3"
      RINSERT 接続 (施工日) RVALUES ("59.9.10")
      TABLE マンホール WHERE Mid = "M8"
CONNECT 接続 (施工日) RVALUES ("59.9.10")
      TABLE 管路 WHERE Kid = "K6"
      マンホール WHERE Mid = "M9"
```

RINSERT句により、行の格納と同時に関連エレメントが作成される。

接続			
ロール1	ロール2	施工日	
r 1	M1	K 1	59.6.4
r 2	M2	K 1	59.6.4
r 3	M2	K 2	59.7.10
r 4	M3	K 2	59.7.10
r 5	M4	K 3	59.7.2
r 6	M5	K 3	59.7.2
r 7	M5	K 4	59.7.16
r 8	M5	K 5	59.7.25
r 9	M6	K 4	59.7.16
r 10	M7	K 5	59.7.25
r 11	M8	K 6	59.8.4
r 12	M9	K 6	59.8.4

図5-4 関連操作の例(例8)

〈例9〉 逗子の配管図の管路K6をマンホールM9から切り離す。

```
DISCONNECT 接続
      FROM RELPRJ (接続, マンホール, 管路)
      WHERE Kid = "K6" AND Mid = "M9" ROLE 0-#2
```

〈例10〉 横須賀の配管図のマンホールM2-M3間の管路K2を、M1-M3間に変更する。

(図5-5参照)

```
RECONNECT 接続
      FROM RELPRJ (接続, マンホール, 管路)
      WHERE Mid = "M2" AND Kid = "K2" ROLE 0-#2
      TO TABLE マンホール WHERE Mid = "M1"
```

接続			
ロール1	ロール2	施工日	
r 1	M1	K 1	59.6.4
r 2	M2	K 1	59.6.4
r 4	M3	K 2	59.7.10
r 5	M4	K 3	59.7.2
r 6	M5	K 3	59.7.2
r 7	M5	K 4	59.7.16
r 8	M5	K 5	59.7.25
r 9	M6	K 4	59.7.16
r 10	M7	K 5	59.7.25
r 13	M1	K 2	59.8.10

図5-5 関連操作の例(例10)

### 5.4 汎化階層操作

汎化階層の操作機能は4.3.1項で述べたように以下の条件を満たす必要がある。汎化階層の操作機能を表5-1に示す。

- ① サブタイプの行が存在すれば、必ずスーパータイプの行が存在する。
- ② スーパータイプの行が存在しなければ、サブタイプの行は存在しない。
- ③ 対応するサブタイプの行が存在しないスーパータイプの行も有り得る。
- ④ サブタイプにはスーパータイプの全ての属性情報が存在し、しかも値が一致している。

表5-1 汎化階層の操作機能

	スーパータイプのテーブルを対象にした場合	サブタイプのテーブルを対象にした場合
参照 (SELECT)	共通属性を参照する	サブタイプの属性情報(共通属性を含む)を参照する
挿入 (INSERT)	スーパータイプの行を作成する	サブタイプの行を作成する。対応するスーパータイプの行がなければ、スーパータイプの行も作成する
削除 (DELETE)	スーパータイプの行および対応するサブタイプの行を削除する	サブタイプの行を削除する。また指定により対応するスーパータイプの行が孤立した時、同時に削除する
更新 (UPDATE)	共通属性を更新し、対応するサブタイプの共通属性を更新する	サブタイプの属性情報を更新する。共通属性の更新時は、対応するスーパータイプの共通属性も更新する

〈例 1 1〉 商店 B 3 を休業する。  
 ( 図 5 - 6 参照 )

```
DELETE FROM 商店
WHERE Bid = "B3"
```

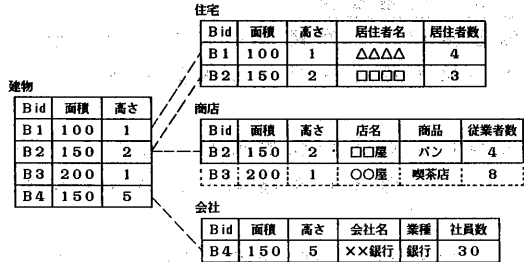


図 5 - 6 汎化階層操作の例 (例 1 1)

## 6. おわりに

本論文では、複雑な対象を表現しかつ高水準な操作機能を備えた M D B 向けのデータモデル、C E R A モデルについて述べた。

C E R A モデルでは、複数の異なる構成要素の集合として構成される対象 (複合対象) を、包含関係に基づいて "グループ" という概念でデータモデル上で表現可能とし、グループを単位とする各種の操作機能を導入した。また対象間の複雑な関係を表し意味の一貫性を保つため、関連および汎化階層の概念をデータモデルにとりこみ、これらに対する高水準な操作機能を新たに導入した。

グループの概念の導入により、5章で述べたグループ操作に加えてさらに以下に示すようなグループを単位とする制御、管理機能を D B M S が実現することが可能になり、複合対象に対する処理を一層効率よく実現することが可能となる。

### ( 1 ) 配置制御

グループ・エレメント単位にデータを近接配置することにより、アクセス効率の向上をはじめ、以降に示す制御機能、管理機能を容易に実現可能とすることができる。

### ( 2 ) データ転送

大容量のデータベースの中からワークステーション等の作業で必要となるデータの部分をグループ・エレメントとし、この単位でのホストとワークステーション間のデータ転送が可能となる。

### ( 3 ) 排他制御

グループ・エレメント毎の配置制御を行なうことにより、グループ・エレメントという論理的な単位での排他制御が容易に実現可能となる。

### ( 4 ) 版管理

グループ・エレメントを単位とする版管理機能が実現可能となる。

## [ 文献 ]

- [1]Haskin,R.L.and Lorie,R.A.:On Extending the Functions of a Relational Database System,SIGMOD,pp207-212,1982
- [2]Chen,P.:The Entity-Relationship Model-Toward a Unified View of Data, ACM TODS Vol.1, No1, January, pp9-36,1976
- [3]Smith,J.M.and Smith,D.C.:Database Abstractions:Aggregation and Generalization,ACM TODS Vol2, No2, pp105-113,1977