データベース・システム 35－1
（1983. 5. 19）

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

宇田川 佳久・溝口 徹夫
（三菱電機株式会社）

ABSTRACT

In this paper, we discuss an extended relational data model for managing data in engineering applications. Most of conventional general purpose data models are designed primarily for efficient storage and retrieval of alphanumeric data. So these data models are difficult to be generally applied for managing engineering data, which are constructed by alpha-numeric and pictorial information.

Our approach is summarized below.

(1) Picture decralation part is furnished in DDL.

(2) A facility called abstraction mechanism is introduced for dealing with a group of objects as a unit.

(3) Occurrences with arguments are introduced for repeating appearance of similar objects.

1. はじめに

工業製品の開発期間の短縮，品質の向上，低コスト化を実現するために，エンジニアリング・データのデータベース化が要求されている。本来，エンジニアリング・データは工業活動に係るデータの総称であるが，本研究では電気回路図，電力配線図，システム構成図，ソフトウェア仕様書，並びにこれらの図面に関する属性情報，事務情報に着目した。これらのエンジニアリング・データは，文字数値データと図形データが混在しており，文字数値データを管理するために開発された従来の汎用データベースでこれらのデータを管理することは難しい。[5]

本論文は，エンジニアリング・データを組織的に管理することを目的としたADAM（Advanced Database system with Abstraction Mechanism）と称する先進的データベースのデータ定義言語について論じたものである。ADAMは概念的にはリレーショナル・データベースに

(1) 引数付きの実現値を導入し，
(2) 実現値に関連する図形表現を定義する項目を加え，
(3) 複数のリレーションを1個の実現値として扱う機能を備え，

たものを多重に用いたデータベースである。ADAMの特徴はリレーショナル・データベースの特徴に加え

(1) 図形情報が扱える，
(2) 関連する一群のデータをひとまとめにして扱うことができる（抽象化機能），
(3) 類似した実現値を抽象化された実現値から導出でき，これによって繰返し出現する実現値を簡潔に記述できる，
(4) データベースの構造を容易に変化させることができる，

(5)　対象物の抽象化とデータモデルの抽象化が一致している，
(6)　対象物を徐々にモデル化できる，
(7)　図面が複雑になっても記述量が急激に増大することがない，
(8)　データの変更が即座に表示図形に反映する．

本論文第2章ではエンジニアリング・データと事務データとの違いを述べ，エンジニアリング・データの特徴を明らかにする。第3章では簡単な例を用いてADAMデータモデルの概要を述べるとともに，他のアプローチとの比較を行なう。第4章ではADAMデータ定義言語について述べ，電力配線図を記述した例を示す．第5章では結論としてADAMデータモデルの特徴を要約するとともに，今後の研究課題について述べる。

2. エンジニアリング・データの特徴

図1は発電機からモータへの電力配線状態を表わした図であり，表1はこの図で用いられている記号とその意味を表わしたものである。エンジニアリングにおける対象物は構成要素の結合状態を表わす図面と，構成要素の属性情報の結合せによって表現される．従って，エンジニアリング・データの第1の特徴は，

(1)　エンジニアリング・データは，文字数値で表現される情報に加え，図形情報が付随する。例えば，図1の電力配線図のスイッチならば，最大定格電流，製造年月日の他に配線図における図形表現 ○ 又は ◎ が付随する．

エンジニアリングで扱う対象物を1枚の図面で表現することは稀である。通常，図面の中で1個の記号として表現されている要素そのものが詳細に記述するならば複数の要素から構成されており，これが再び図面によって表現される．すなわち，エンジニアリング・データの第2の特徴は，

(2)　エンジニアリング・データの構造は複雑である。

このために，エンジニアリング・データベース・システムは幾つかの要素をまとめて1つの要素とみなし，対象物の構造をデータモデルに反映させることができることが望ましい．構造化のためには注目している情報を陽に記述し，他の情報を外部からアクセスできないようにする必要がある（抽象化）。

今までに地図情報を汎用データベースによって管理する手法が研究されてきた[1][2]。地図情報は図形情報とその属性情報と

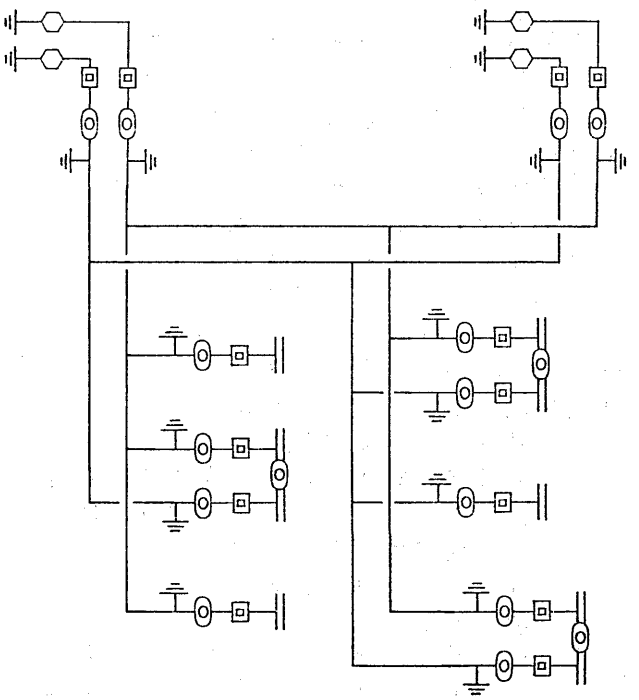

図1．　電力配線図の例。

PART	NOTATION	PART	NOTATION
POWER SUPPLIER		EARTH	
SWITCH (OFF)		MOTOR	
SWITCH (ON)		DOUBLE MOTORS	
BREAKER (OFF)			
BREAKER (ON)		TERMINAL	

表1．　図1の要素とその表現。

の組合せによって表現されるものであり，この点においては(1)，(2)の特徴を有している。しかし，図1を注意深く観察するとエンジニアリング・データに現われる図形情報は，地図で用いられる図形情報と次の点で異なることがわかる，
(3)　類似した実現値が多数出現する。
これはエンジニアリングで用いられる部品が規格化され，モジュール化されていることに起因する。例えば，図1にはスイッチやブレーカーが多数出現する。
最後に，エンジニアリングにおける対象物は，初めから完成された形で記述されるものではなく，企画，設計，生産，販売に到る段階で徐々に詳細化されるものである。すなわち，エンジニアリング・データの第4の特徴は，
(4)　記述対象を一度に詳細に分析してデータベース化することは困難である。
エンジニアリング・データの(2)と(4)の特徴から，このデータ管理においては対象物を徐々にデータベース化することが重要であることが理解できる。そのために，エンジニアリング・データベースは，データベースの構造を容易に変化させることができるものであることが望ましい。

3．ADAMによる記述例と他の方法との比較

この章では簡単な例を用いてADAMデータ定義言語の概観を示すとともに，ADAMのアプローチと他のアプローチとの比較を行う。リレーショナル・データベースによる図形データ管理の試みにN.S.ChangとK.S.FuによるREDI[3]がある。彼らは文献3で，図形を異なった抽象レベルの階層構造に展開する方法を論じ，これをリレーショナル・データモデルによって表現するアルゴリズムを与えている。例えば図2に示した図形は図3に示した階層構造によって表現され，これは図4に示したリレーションによって表現される。図3から図4への変換アルゴリズムの概要を次に示す。
(1)　エッジに付いたラベル名 ⟷ リレーション名。

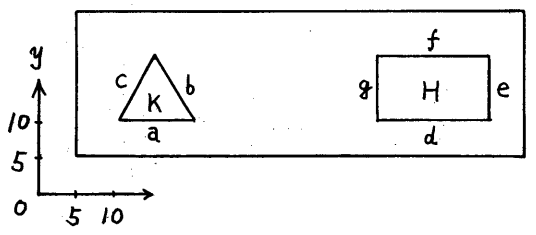

図2．　図形データの例。

(2) 同じラベルが付いているエッジの集合（ノードのペア）⟷リレーションのタプル。

(3) 各ノード，エッジに付随する属性を適宜付加する。

しかし，この方法は次に述べる問題がある。

(1) 複数の抽象化レベルによって表現された図形を1レベルのリレーションによって表現しているために，対象物の構造とデータモデルの構造とが一致しない。

(2) 各抽象レベルが異なる属性集合によって記述される場合，文献3]で論じられている方法を適用することは難しい。

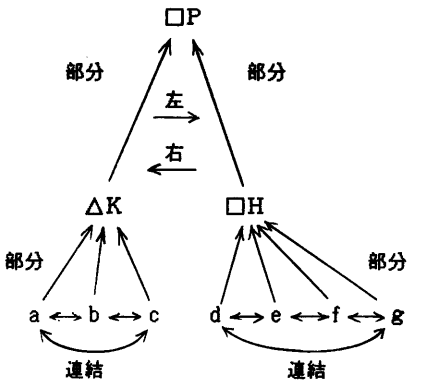

図3. 図2の構成要素の階層構造。

例えば△Kに対しては面積，線分αに対しては長さの属性が考えられる。図4のリレーション"部分"では△Kとαとを同じ属性"要素1"として扱っているが，このリレーションに面積及び長さの属性を同時に，かつNull値を用いずに導入することは難しい。

(3) 各図形構成要素を個別に記述しているために，類似の図形が繰返し出現するような対象物に対しては記述が繁雑になる。

図1に示した図形をADAMの定義言語によって記述したものの概要を図5に示した。図4に比べて図5の方が記述が長くなっているが，これは図4では図2に示した図形の"名前"の関係を示しているだけであり"名前"によって示された図形がどのような形状や属性を持っているかを指定していない。これに対し，図5では図形の形状及びその属性（面積，長さ）まで指定しているからである。いわば記述の詳しさが違うことが原因であるが，ADAMのデータ定義言語では任意の図形の形を指定することができることが特徴である。

ADAMデータ定義言語を用いれば，図2に示した図形を図3に示した階層構造に従って表現することができる。例えば，図5の□Pの定義のリレーション"部

部分	要素1	要素2
	△K	□P
	□H	□P
	a	△K
	b	△K
	c	△K
	d	□H
	e	□H
	f	□H
	g	□H

連結	要素1	要素2
	a	b
	b	c
	c	a
	d	e
	e	f
	f	g
	g	d

左	要素1	要素2
	△K	□H

右	要素1	要素2
	□H	△K

図4. リレーショナル・データベースによる記述例。

分"で図形要素△Kと□Hが用いられているが，これは図3の"□Pは△Kと□Hから構成されている"という階層構造を表わしている。この階層表現によって図形要素に関連する属性を容易に付け加えることができる。例えば□Pの構成要素である△Kと□Hに関する属性"面積"は□Pの定義部で記述することができる。また，△Kと□Hの構成要素である線分a, b, …, gの属性"長さ"は△K及び□Hの定義部で記述することができる。

ADAMデータモデルでは引数付きの実現値を用いている。引数に適切な値を代入することによって繰返し出現する実現値を容易に表現し管理することができる。例えば，図2に□Hを追加したいとする（図6）。ADAMでこれを行なうためには，図5のリレーション"部分"にタプル＜□H(X+20, Y+5), 150.0＞を，リレーション"左"にタプル＜□H(X+20, Y+5), □H(X+40, Y+5)＞を，リレーション"右"にタプル＜□H(X+40, Y+5), □H(X+20, Y+5)＞をそう入すれば良い。これに対し，Changらの方法では11箇所にわたって変更を行なう必要がある。引数付きの実現値を導入することによって記述が簡潔になるだけでなく，更新処理を含むデータ操作に対しても良い

```
//  図形  □P  の定義  //
変数     ( x, y )
名前      □P ( x, y )
概形     長方形 ( x, y, 60, 20 )
関係
```

部分	要素	面積
	△K (x+5 , y+5)	43. 3
	□H (x+40 , y+5)	150. 0

左	要素1	要素2
	△K (x+5 , y+5)	□H (x+40 , y+5)

右	要素1	要素2
	□H (x+40 , y+5)	△K (x+5 , y+5)

```
詳細図形
        表示 ( 部分 [ 要素 ] )

//  正三角形  △K  の定義  //
変数     ( x, y )
名前      △K ( x, y )
概形     正三角形 ( x, y, 10 )
関係
```

部分	要素	長さ
	a (x, y, x+10, y)	10
	b (x+10, y, x+5, y+8. 7)	10
	c (x+5, y+8. 7, x, y)	10

```
詳細図形
        表示 ( 部分 [ 要素 ] )

//  長方形  □H  の定義  //
変数     ( x, y )
名前      □H ( x, y )
概形     長方形 ( x, y, 20, 30 )
関係
```

部分	要素	長さ
	d (x, y, x+15, y)	15
	e (x+15, y, x+15, y+10)	10
	f (x+15, y+10, x, y+10)	15
	g (x, y+10, x, y)	10

```
詳細図形
        表示 ( 部分 [ 要素 ] )

//  線分  a  の定義  //
変数     ( x, y, v, w )
名前      a ( x, y, v, w )
概形     線分 ( x, y, v, w )
          。
          。
          。
//  線分  g  の定義  //
変数     ( x, y, v, w )
名前      g ( x, y, v, w )
概形     線分 ( x, y, v, w )
```

図5.　ADAMデータ定義言語の概要。

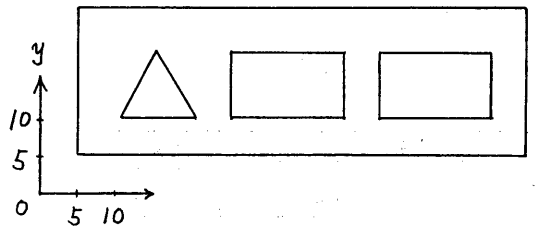

図6. 図形データの例。

結果をもたらすものと考えられる。

4. ADAMデータ定義言語と記述例

4.1. ADAMデータ定義言語の概観

ADAMデータ定義言語は以下の7項目から構成されている。

(1) 変数宣言。
(2) 抽象化された実現値の宣言。
(3) 実現値の抽象化された図形の記述。
(4) 属性の宣言。
(5) リレーションの記述。
(6) 一貫性条件の記述。
(7) 実現値の詳細な図形の記述。

これらの項目を記述することによって，1つの抽象化された実現値が定義される。ADAMデータベースは，上記のデータ定義言語によって定義された実現値の集まりである。図7はADAMデータベースをBNF構文で表現したものである。上記の項目のうち(1)，(2)，(3)，(7) はADAM特有の項目である。以下それらの機能と運用法について論じよう。

```
< ADAM database > ::= < ADAM database element >
                    | < ADAM database > ; < ADAM database element >
< ADAM database element > ::=
                      [ < variable declaration > ; ]
                      [ < abasract instance declaration > ; ]
                      [ < general figure > ; ]
                      [ < attribute declaration > ; ]
                      [ < list of relations > ; ]
                      [ < list of consistency constraints > ; ]
                      [ < detailed figure > ; ]
```

図7. ADAMの構文。

4.1.1. 変数宣言

ADAMデータモデルでは，引数付きの実現値を導入している。変数宣言項目の役割は引数で用いられる変数の名前と定義域を宣言することである。図8は変数宣言の1例である。この例では変数X，Yが0.0から300.0の間の不動小数を値

```
VARIABLE-DECLARATION
<    X    OVER    SUBRANGE( 0.0 .. 300.0 ) ;
     Y    OVER    SUBRANGE( 0.0 .. 300.0 ) ;
     ON   OVER    BOOLEAN
 > ;
```

図8. 変数宣言の例。

とし，変数 ON が T（真）又は F（偽）を値とすることを宣言している．

4.1.2. 抽象化された実現値の宣言

ADAM データモデルでは抽象化された実現値を記述する場合，その実現値の名前と引数を宣言しなければならない．図9に示した例は，変数 X，Y，ON を引数とし，S と名付けられた実現値を宣言している．

```
ABSTRACT-INSTANCE    S  ( X, Y, ON )    ;
```

図9. 抽象化された実現値の宣言。

4.1.3. 実現値の抽象化された図形の記述

ADAM において実現値はより高い抽象レベルと，より低い抽象レベルとから参照される．図10にその参照形態を図示した．"実現値の抽象化された図形の記述" 項目では，より高いレベルから参照される図形表現（実現値の抽象化された図形表現）を記述する．概念的には図10の抽象レベル I における太線で示した四辺形に相当する図形を記述する．図11は抽象化された図形の記述例である．同図で関数 ELIPS（2.0，3.0，X，Y）は X，Y 軸方向が 2.0，3.0 であり，中心が X，Y である楕円を表示する．もし，ON の値が T（真）であるならば ELIPS（1.0，1.5，X，Y）が実行される．従って ON が T なら ◎ が表示され，F ならば ○ が表示される．

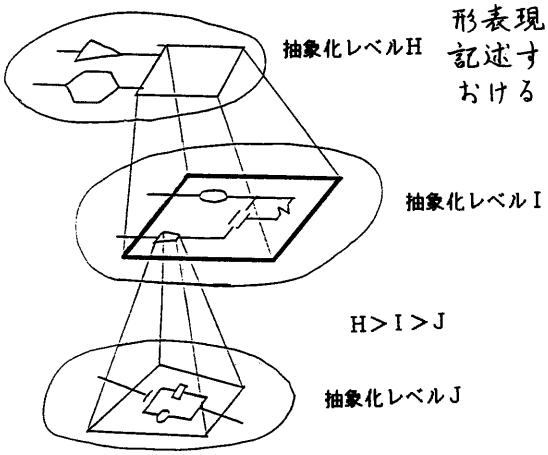

図10. 実現値とその参照パターン。

```
GENERAL-FIGURE
 <    ELIPS ( 2.0, 3.0, X, Y ) ;
    IF    ON==T    THEN
       <  ELIPS ( 1.0, 1.5, X, Y ) > ;
```

図11. 抽象化された図形の記述例。

4.1.4. 実現値の詳細図形の記述

この項目では対象物の詳細な図形をより低い抽象レベルの実現値を用いて記述する．概念的には図10の太線の枠内に書かれている図形を記述する．詳細図形は対象物の構成要素の結合状態を記述するものであり，その状態は頻繁に変更される．ADAM では対象物の構造の変化が即座に表示図形に反映するように，この図形をリレーションを引数とする図形表示関数を用いて記述している．図12は詳細図形の記述例である．

```
DETAILED-FIGURE
<   GRAPHICS ( FIG | ELEM | ) ;
    X ( H1, V1, DTNAME, DINAME ) :=
            ( CONN | G ; STNAME = TNAME ,
                         SINAME = INAME | TERM )
            | HCOOR, VCOOR, DTNAME, DINAME | ;
    RLINE ( ( X | G ; DTNAME = TNAME ,
                      DINAME = INAME | TERM )
            | H1, V1, HCOOR, VCOOR | )
>
```

図12.　詳細図形の記述例。

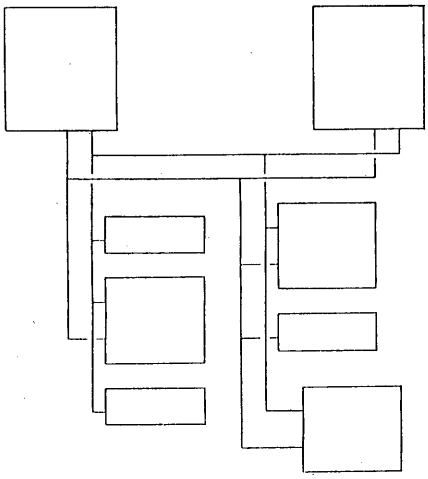

図13.　図1を抽象化した図。

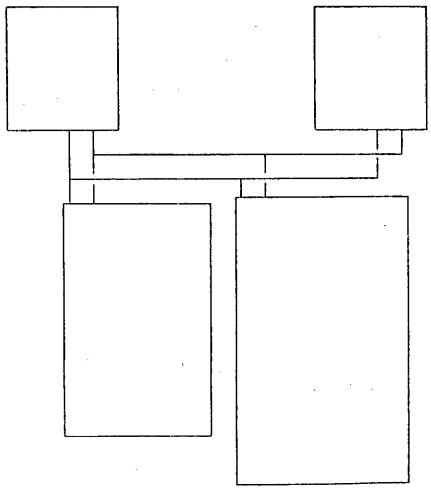

図14.　図13を抽象化した図。

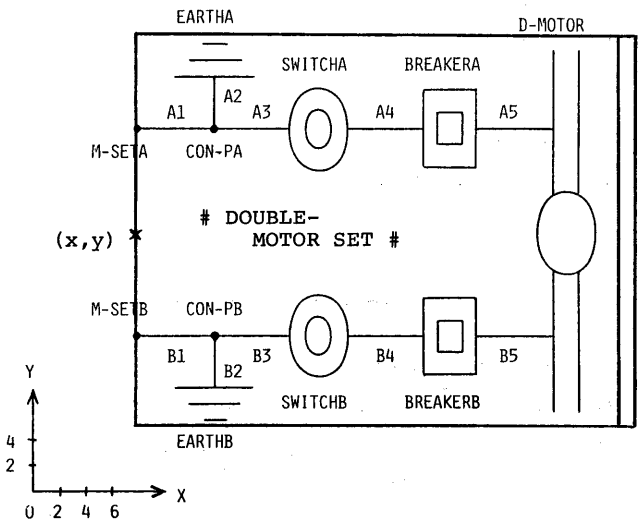

図15.　動力装置の例。

4.2. ADAMデータ定義言語の記述例

この節ではADAMデータ定義言語の特徴を明らかにするために記述例を示す。図1は電力の配電状況を詳細に記述したものであるが，電力の配電系がどのようなモジュールから構成されているのかを表現するためには発電装置と動力装置をひとまとめにし，これを単位として記述するのが有効である。図13は，図1を抽象化を行ない，詳細な構造を隠した電力配線図である。このような抽象化を必要な回数だけ繰返えすことにより，ユーザが希望するように対象物を構造化することができる。図14は図13の動力装置をひとまとめにし，電力の供給源と需要先との関係に注目できるように構造化したものである。図14では電力の供給源と需要先とが2重の配電系によって結合されていることが明確に表現されている。

図15に示した動力装置をADAMデータ定義言語によって記述したものを図16に示した。図16の9～12行目は変数X，Yを宣言している。これらの変数は図15に示した基準座標として用いられている。すなわち，変数X，Yに適切な値を代入することによって，この動力装置を任意の位置に定義することができる。図16の13行目は，この動力装置をDMSET（X，Y）と呼ぶことを宣言している。14～22行目は，この動力装置の概形図形を定義している。このうち，15～19行目は32×36の長方形 を定義しており，20～21行目は #DOUBLE-MOTOR SET# なる文字列を定義している。23～36行目は，リレーションの定義で用いる属性とその領域とを定義している。

37～50行目はリレーションFIG（INAME，POWER，ELEM）を定義

```
 1.000 /*****************************************/
 2.000 /*                                       */
 3.000 /*          DEFINITION OF                */
 4.000 /*      ABSTRACT INSTANCE                */
 5.000 /*              DOUBLE MOTOR SET         */
 6.000 /*                                       */
 7.000 /*****************************************/
 8.000 /**/
 9.000 VARIABLE-DECLARATION
10.000 <   X   OVER   SUBRANGE( 0.0 .. 300.0 ) ;
11.000     Y   OVER   SUBRANGE( 0.0 .. 300.0 )
12.000 > ;
13.000 ABSTRACT-INSTANCE   DMSET ( X, Y )  ;
14.000 GENERAL-FIGURE
15.000 <  LINE( X      , Y+16.0, X      , Y-16.0 ;
16.000          X      , Y-16.0, X+36.0, Y-16.0 ;
17.000          X+36.0, Y-16.0, X+36.0, Y+16.0 ;
18.000          X+36.0, Y+16.0, X      , Y+16.0 ;
19.000          X+34.0, Y+16.0, X+34.0, Y-16.0 ;
20.000    SYMBOL( X+3.0, Y+2.0,  2, "# DOUBLE-" ) ;
21.000    SYMBOL( X+7.0, Y-1.0,  2, "MOTOR SET #" )
22.000 > ;
23.000 ATTRIBUTE-DECLARATION
24.000 <   INAME    ON  CHAR  ;
25.000     POWER    ON  SUBRANGE( 0 .. 200 ) ;
26.000     ELEM     ON  ABSTRACT-INSTANCE ;
27.000     TNAME    ON  CHAR  ;
28.000     INAME    ON  CHAR  ;
29.000     HCOOR    ON  SUBRANGE( 0.0 .. 300.0 )  ;
30.000     VCOOR    ON  SUBRANGE( 0.0 .. 300.0 )  ;
31.000     LINE-ID  ON  CHAR(4)  ;
32.000     STNAME   ON  CHAR(4)  ;
33.000     SINAME   ON  CHAR  ;
34.000     DTNAME   ON  CHAR(4)  ;
35.000     DINAME   ON  CHAR
36.000 >  ;
37.000 RELATION-DEFINITION
38.000    FIG ( INAME, POWER, ELEM ) ;
39.000 <  EARTHA  , 30, E1( X+6.0, Y+12.0, 3 ) ;
40.000    SWITCHA , 30, S ( X+14.0, Y+8.0, TRUE ) ;
41.000    BREAKERA, 20, B ( X+24.0, Y+8.0, TRUE ) ;
42.000    D-MOTOR , 20, MM( X+32.0, Y ) ;
43.000    M-SETA  , 20, P ( X, Y+8.0 ) ;
44.000    CON-PA  , 50, P ( X+6.0, Y+8 ) ;
45.000    EARTHB  , 30, E3( X+6.0, Y-12.0,  3 ) ;
46.000    SWITCHB , 30, S ( X+14.0, Y-8.0, TRUE ) ;
47.000    BREAKERB, 20, B ( X+24.0, Y-8.0, TRUE ) ;
48.000    M-SETB  , 20, P ( X, Y-8.0 ) ;
49.000    CON-PB  , 50, P ( X+6.0, Y-8.0 )
50.000 > ;
51.000 /*  RELATIONSHIP BETWEEN A TERMINAL AND
52.000     AN INSTRUMENT  */
53.000 RELATION-DEFINITION
54.000    TERM ( TNAME, INAME, HCOOR, VCOOR, POWER ) ;
55.000 <  A, M-SETA  , X      , Y+8.0, 20 ;
56.000    C, CON-PA  , X+6.0 , Y+8.0, 50 ;
57.000    E, EARTHA  , X+6.0 , Y+12.0, 30 ;
58.000    I, SWITCHA , X+12.0, Y+8.0, 30 ;
59.000    O, SWITCHA , X+16.0, Y+8.0,  0 ;
60.000    I, BREAKERA, X+22.0, Y+8.0, 20 ;
61.000    O, BREAKERA, X+26.0, Y+8.0,  0 ;
62.000   IA, D-MOTOR , X+32.0, Y+8.0, 20 ;
63.000    A, M-SETB  , X      , Y-8.0, 20 ;
64.000    C, CON-PB  , X+6.0 , Y-8.0, 50 ;
65.000    E, EARTHB  , X+6.0 , Y-12.0,30 ;
66.000    I, SWITCHB , X+12.0, Y-8.0 , 30 ;
67.000    O, SWITCHB , X+16.0, Y-8.0,  0 ;
68.000    I, BREAKERB, X+22.0, Y-8.0, 20 ;
69.000    O, BREAKERB, X+26.0, Y-8.0,  0 ;
70.000   IB, D-MOTOR , X+32.0, Y-8.0, 20
71.000 > ;
72.000 /*  RELATIONSHIP BETWEEN TERMINAL POINTS
73.000     AND CONNECTIONS  */
74.000 RELATION-DEFINITION
75.000    CONN ( LINE-ID, STNAME, SINAME, DTNAME, DINAME )
76.000 < A1,  A, M-SETA,  C, CON-PA ;
77.000   A2,  C, CON-PA,  E, EARTHA ;
78.000   A3,  C, CON-PA,  I, SWITCHA ;
79.000   A4,  O, SWITCHA, I, BREAKERA ;
80.000   A5,  O, BREAKERA,IA, D-MOTOR ;
81.000   B1,  A, M-SETB,  C, CON-PB ;
82.000   B2,  C, CON-PB,  E, EARTHB ;
83.000   B3,  C, CON-PB,  I, SWITCHB ;
84.000   B4,  O, SWITCHB, I, BREAKERB ;
85.000   B5,  O, BREAKERB, IB, D-MOTOR
86.000 > ;
87.000 DETAILED-FIGURE
88.000 <  GRAPHICS ( FIG | ELEM | ) ;
89.000    X ( H1, V1, DTNAME, DINAME ) :=
90.000              ( CONN | G ; STNAME = TNAME ,
91.000                           SINAME = INAME | TERM )
92.000            | HCOOR, VCOOR, DTNAME, DINAME | ;
93.000    RLINE ( ( X | G ; DTNAME = TNAME ,
94.000                      DINAME = INAME | TERM )
95.000            | H1, V1, HCOOR, VCOOR | )
96.000 >
97.000 END
```

図16.　ADAMデータ定義言語の記述例。

している．このリレーションは属性 ELEM で示された抽象的実現値の電力（POWER）とその識別子（INAME）との関係を定義している．53～71 目はリレーション TERM（TNAME，INAME，HCOOR，VCOOR，POWER）を定義している．このリレーションは INAME で示された要素の端点（TNAME）の座標（HCOOR，VCOOR）とその端点における電力（POWER）との関係を示している．75～86行目はリレーション CONN（LINE-ID，STNAME，SINAME，DTNAME，DINAME）を定義している．このリレーションは端点 STNAME，SINAME と端点 DTNAME，DINAME が線分 LINE-ID によって結合していることを定義している．88行目は動力装置の構成要素を表示することを指定している．89～95行目はリレーション CONN で定義されている結線状態を表示することを指定している．

5．おわりに

本研究の中心課題は図面をいかに表現し，その意味をいかに処理するかにある．図面は図形情報，属性情報を含む複雑な内部構造を有するために，これを管理するためには一群の要素をまとめて1つの要素とみなす抽象化が必要となる．抽象化することによって実体の主要な特性に注目することができ，結果として現実的な複雑さを有する図面を管理できることと考えられる．

今後の研究課題として，データ操作言語の設計，データ一貫性の制御方式の確立などが残されている[6,7]．

謝辞：日頃御指導いただく三菱電機情報電子研究所情報処理部長・首藤 勝博士，同ソフトウェアグループ・和田 雄次氏，三石 彰純氏，並びに関連諸氏に感謝します．

REFERENCES

1] Kobayashi, I. Cartographic Databases. In "Pictorial Information Systems," Lecture Notes in Computer Science, Vol.80, Springer-Verlag (1980) 322-350.

2] Matsuka, H. and Uno, S. Canonical Geometric Modeling for Computer Aided Degign. In "Data Base Techniques for Pictorial Applications" Lecture Notes in Computer Science, Vol.81, Springer-Verlag (1980) 233-252.

3] Chang, N.S. and Fu, K.S. A relational database system for images. In "Pictorial Information Systems," Lecture Notes in Computer Science, Vol.80. Springer-Verlag (1980) 288-321.

4] Herot, C.F. Spatial Management of Data. ACM Trans. on Database Systems, Vol.5, No.4 (1980) 493-514.

5] Haskin, R.L. and Lorie, R.A. On extending the functions of a relational database system. Proc. ACM-SIGMOD Int. Conf. Management of data (Orlando, June 2-4) 1982, p207-212.

6] 宇田川・溝口，拡張関係代数の先進的データベースＡＤＡＭへの応用，Logic Programming Conference'83 #4.3 (1983)。

7] 宇田川・溝口，エンジニアリング・データ管理のための関係データベースの拡張，昭和５８年電子通信学会総合全国大会，#1368(1983)。