

エンジニアリング・データ管理への接近

宇田川 佳久・溝口 徹夫

(三菱電機株式会社)

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求められている。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，電力系統図，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とが混在しており，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。

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 REDI¹⁾ がある。彼らは文献 1) 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。しかし、この方法は次に示す問題がある。

- (1) 複数の抽象レベルによって階層構造に表現した図形を 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 [1] で論じられている方法を適用することは難しい。
- (3) 各図形構成要素を個別に記述しているために、類似の図形が繰返えし出現するような対象物に対しては記述が繁雑になる。

リレーショナル・データベースと図形表示機能を組合せた研究に C.F. Herot の SDMS がある²⁾。SDMS はデータベース内の指定された情報をデータ・ディクショナリに定義された表示図形に反映することができるデータベースである。この表示図形によって、ユーザはデータベースの内容を視覚的に把握することができ、データベースの構造を知らなくても所望のデータを検索することができる。C.F. Herot の研究は、文字数値情報を表示図形に反映させることによって文字数値情報を管理するデータベースの使い易さを向上させ

ることを意図したものである。これに対し、本研究は図形情報を本質的に有するデータを管理しようとするものであり、C. F. Herotの方針とは基本的に異なるものである。

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・システムに関するものである。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、これにより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・データとビジネス・データとの違いを要約する。第3章では ADAM データモデルとリレーショナル・データモデルとの関連について論ずる。第4章では ADAM データ定義言語の概観を述べ、電力系統図を記述した例を示す。第5章では結論として ADAM データベース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電状況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(発電機から最大発電電力など)の組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

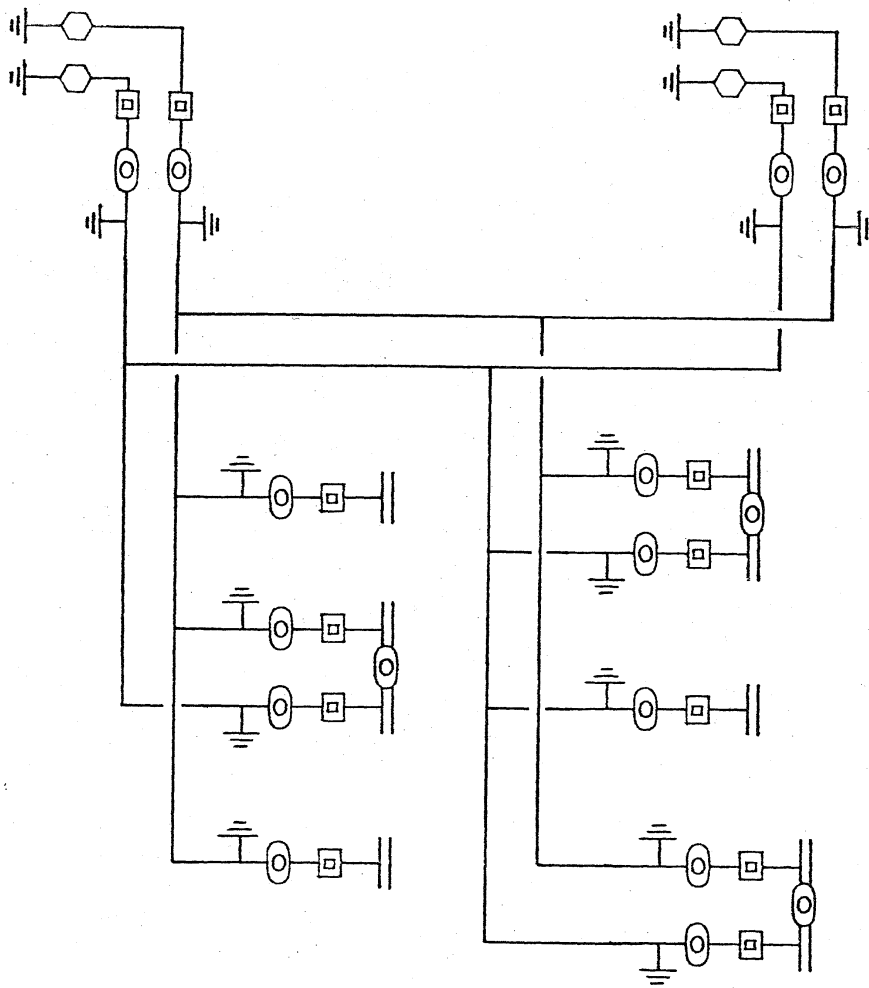


図 1. 電力配線図の例

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

表 1. 図 1 の要素とその表現

情報に加え，図形情報を本質的に有する。図1の電力配線図のスイッチを例にすれば，最大定格電流，製造年月日の他に配電図における図形表現 \bigcirc 又は \odot が付随する。

エンジニアリングで扱われる対象物を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つ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・システムはいく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ものでなければならぬ。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³⁾⁴⁾。地図情報は図形情報と図形の属性情報と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、販売に至る段階で徐々に詳細化されるものである。

(4) 対象物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なければならない。

3. ADAM データモデルとリレーショナル・データモデルとの関連

Codd のリレーショナル・データモデルでは2種類の要素、すなわち

- (1) 文字列または数値で構成されたタプル(レコード)
 (2) ホモジニアスなタプルの集合であるリレーション
 を用いて対象物を表現していた。その結果, リレーショナル
 ・データベースは記述しようとする対象物の構造が比較的単
 純である場合に有効である(例えば事務情報管理)。図2は

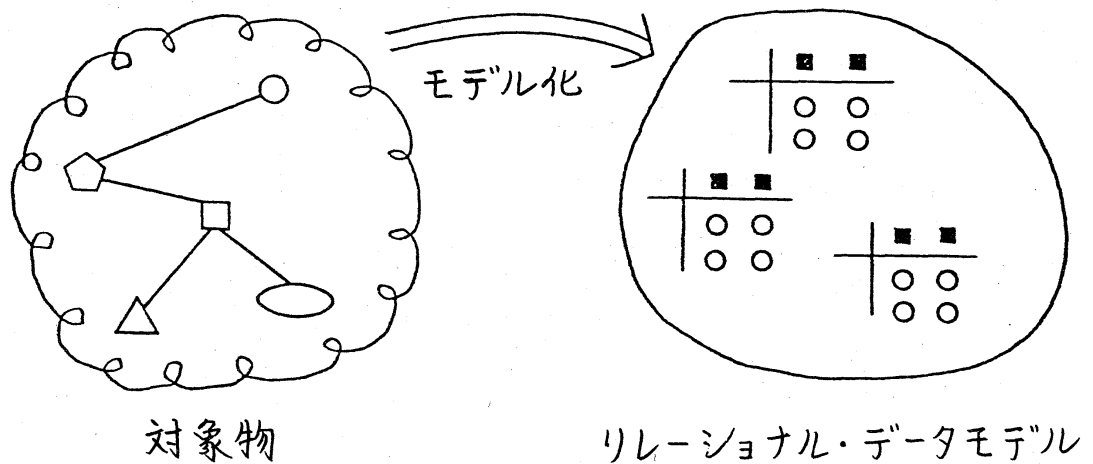


図2. 対象物とリレーショナル・データモデルの関係

対象物とリレーショナル・データモデルとの概念的な関係を示している。

これに対し, エンジニアリングで扱われる対象物は構造が複雑であるために単純なデータモデルでは表現しにくいものが多い。さらに, これらの対象物は物理的な形状をもち, 図面上で対象物を識別するための慣用的な表示図形をもっている。技術者は, この表示図形を通して対象物の構造や特性を

把握することができる。表示図形は新しい発想を生み出す助けをするものであり、エンジニアリングにおいては本質的に重要な役割を演じている。さらに、エンジニアリングで用いられる対象物は、大局的には1個の实体であるけれども詳細に見るならば複数の部品から構成されたものである。従って、これらの対象物を管理するためには、対象物を大局的に記述したり、逆に詳細に記述できるものでなければならぬ。ADAMでは、この機能を抽象化機能 (Abstraction Mechanism) と呼んでいる。

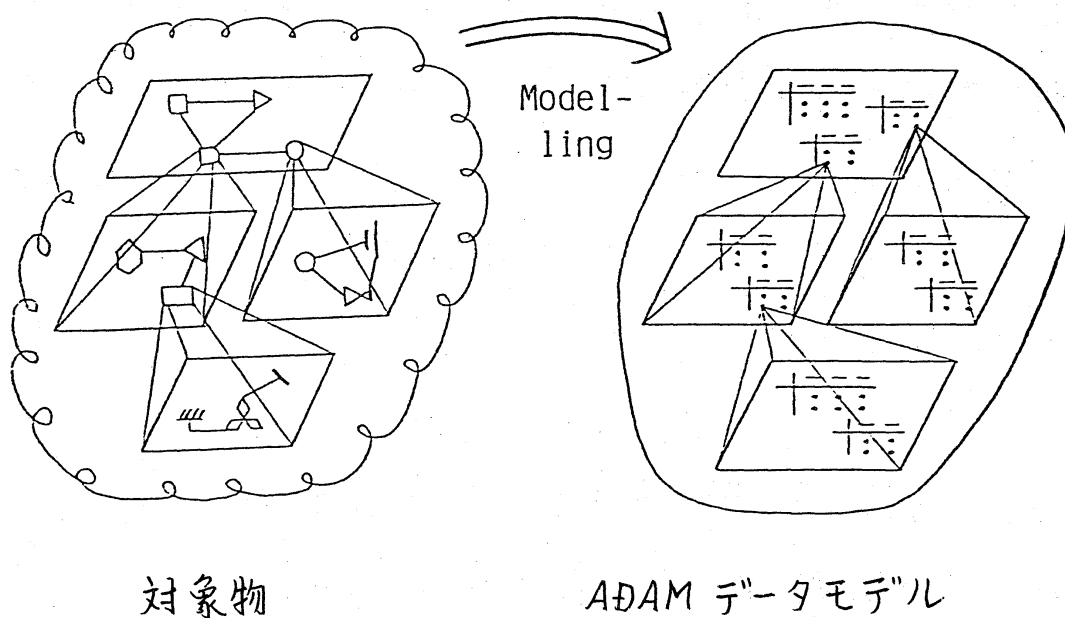


図3. 対象物とADAMデータモデルとの関係

図3は、対象物とADAMデータモデルとの概念的な関係を表わしている。ADAMデータモデルでは、個々の抽象化レベルの実体をリレーショナル・データモデルと同様な方法で記述する。ADAMは、こうして記述された実体の集まりに名称とその表示図形を宣言し、それを代表する特性（例えば対象物の位置や内部状態など）を引数として宣言する機能を備えている。このようにして宣言された対象物の集まりは、さらに高い抽象化レベルでは1つの対象物として扱われる。最も下位の抽象化レベルでは、システム作り付けの対象物、すなわち文字列、数値、図形表現をもつ基本的な関数によって記述される。言い換えれば、ADAMデータベースにおいて文字列と数値のみを実現値として用い、かつ1つの抽象化レベルしか用いない場合がリレーショナル・データベースに対応する。この意味において、ADAMはリレーショナル・データベースを包含している。

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体が定義される。ADAM データベースは、上記のデータ定義言語によって記述された実体の集まりである。図4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特有の項目であるので、以下それぞれの機能と運用法について論じよう。

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5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

```

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

```

図4. ADAMの構文

```

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

```

図5. 変数宣言の例

```

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

```

図6. 抽象化された実現値の宣言例

6に示した例は、変数 X , Y , ON を引数とし、 S と命名された実現値を宣言している。

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7に実現値とその参照パターンを図示した。"実現値の抽象化された図形の記述"項目では、より高い抽象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7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8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 である橢円を表示する、もし ON の値が T (真) であるならば $ELIPS(1.0, 1.5, X, Y)$ が表

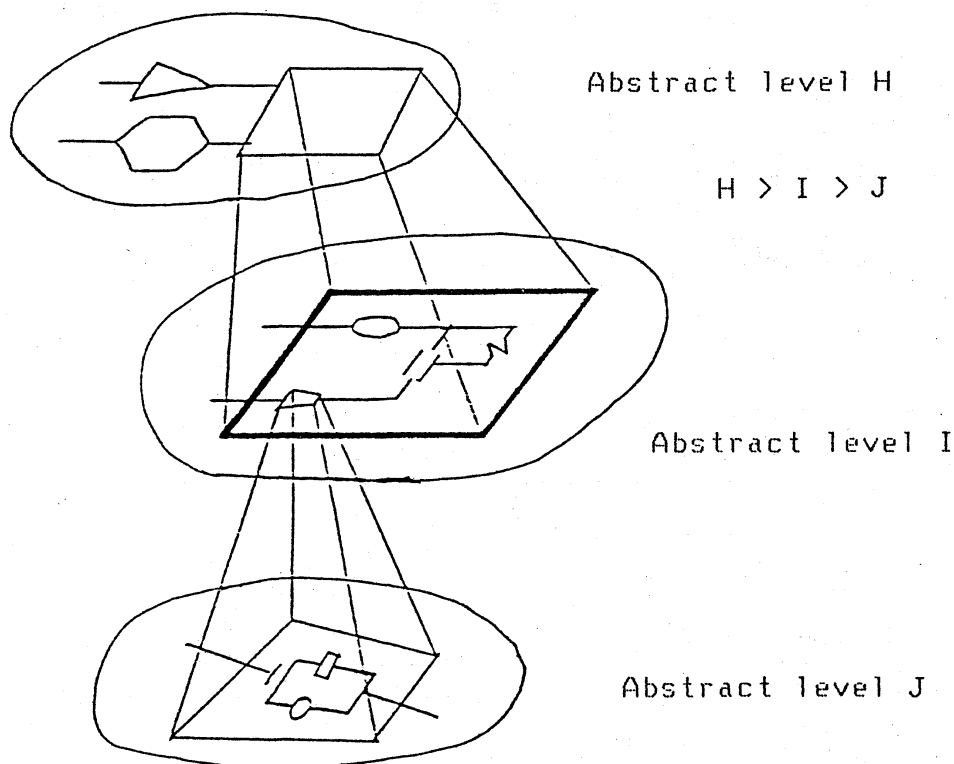


図7. 実現値とその参照パターン

示される。従って ON が T なら \odot が表示され、F なら \bigcirc が表示される。

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

図8. 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7の太線の枠内に書かれた図形を記述する。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

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

図9. 詳細図形の記述例

を用いて記述している。図9は詳細図形の記述例である。

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図1に示した電力配線図を記述してみる。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10は、図1を抽象化し^細詳細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すことにより、ユーザが希望するように対象物

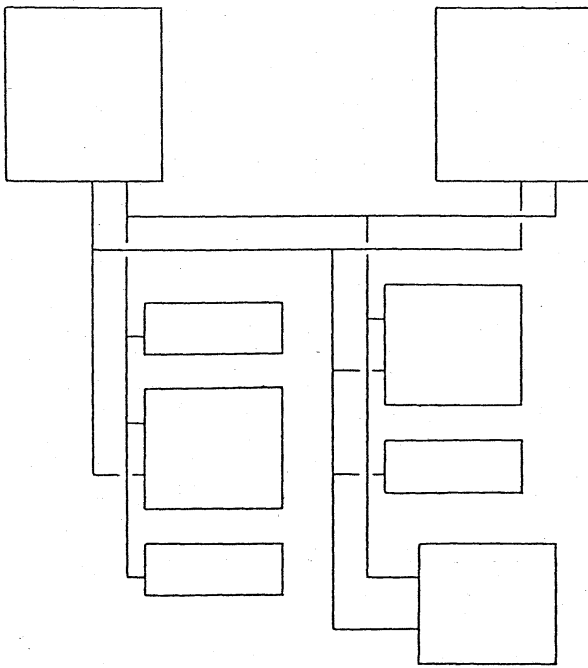


図10. 図1を抽象化した図

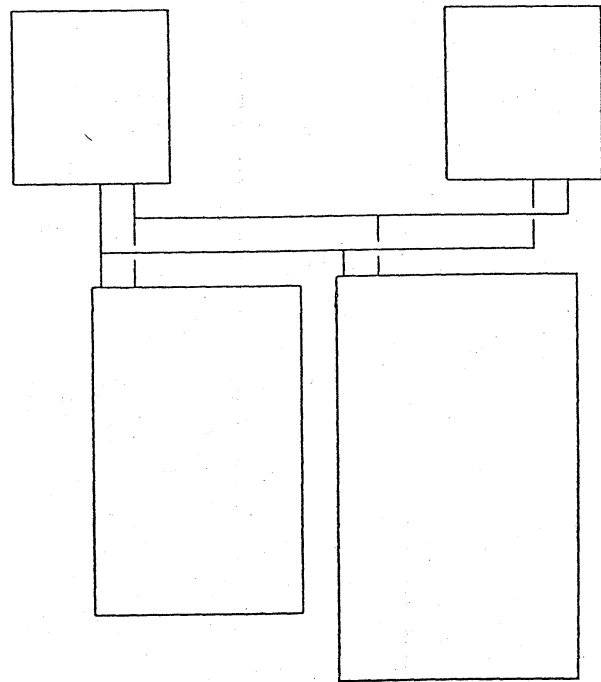


図11. 図10を抽象化した図

を構造化することができる。図11は図10の配電団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。この図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12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13に示した。図13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12に示した基準座標とし

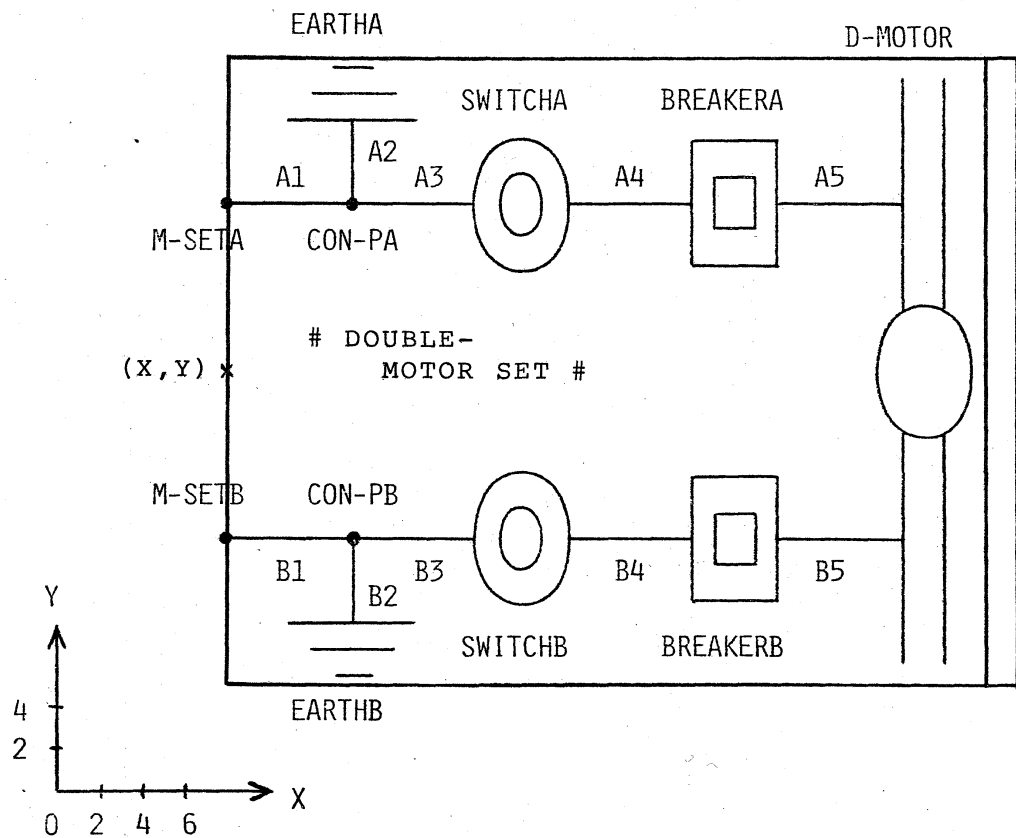


図12. 動力装置の例


```

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     EARTH B , 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 > ;

```

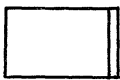
図13. ADAM テータ定義言語の記述例 (続く)

```

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, EARTH B , 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, EARTH B ;
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 I ELEM I ) ;
89.000 X ( H1, V1, DTNAME, DINAME ) :=
90.000 ( CONN I G ; STNAME = TNAME ,
91.000 SINAME = INAME I TERM )
92.000 I HCOOR, VCOOR, DTNAME, DINAME I ;
93.000 RLINE ( ( X I G ; DTNAME = TNAME ,
94.000 DINAME = INAME I TERM )
95.000 I H1, V1, HCOOR, VCOOR I )
96.000 >
97.000 END

```

図13. (続き)

て用いられる。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13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 $FIG(INAME, POWER, ELEM)$ を定義している。このリレーションは属性 $ELEM$ で示された抽象的实现値の電力 ($POWER$) とその識別名 ($INAME$) との関係性を定義している。53~71行目はリレーション $TERM(TNAME, INAME, HCOORD, VCOORD, POWER)$ 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 ($HCOORD, VCOORD$) とその端点における電力 ($POWER$) との関係性を示している。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、電力配電図、システム構成図などの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、これを管理するためには一群の要素をまとめて一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思われる。

本報告は上記の要求に対処した先進的データベース・システム ADAM のデータ定義言語について述べ、その有効性を電力配線図を例にして論じ、エンジニアリング・データ管理の一つのアプローチを明示した。

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],8]}。

謝辞： 今回の発表の機会を与えて下さいました東京大学大須賀節雄教授に感謝します。日頃御指導いただき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、並びに関連諸氏に感謝します。

REFERENCES

- 1] 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.
- 2] Herot, C.F. Spatial Management of Data. ACM Trans. on Database Systems, Vol. 5, No. 4 (1980) 493-514.
- 3] Kobayashi, I. Cartographic Databases. In "Pictorial Information Systems," Lecture Notes in Computer Science, Vol. 80, Springer-Verlag (1980) 322-350.
- 4] Matsuka, H. and Uno, S. Canonical Geometric Modeling for Computer Aided Design. In "Data Base Techniques for Pictorial Applications" Lecture Notes in Computer Science, Vol. 81, Springer-Verlag (1980) 233-252.
- 5] Codd, E.F. A Relational Model of Data for Large Shared Data Banks. Commun. ACM, 13, 6 (1970) 377-387.
- 6] 宇田川, 溝口 : 関係データベースの拡張とエンジニアリング・データ管理への応用, 情報処理学会第26回全国大会予稿集(1983) p733~734,
- 7] 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, ロジックプログラミング・コンファレンス '83 (1983) #4.3.
- 8] 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).