

## EAR モデルに基づく情報構造記述を用いた プログラム仕様記述法 PSDM†

橋 本 正 明\*\*

プログラム再利用の容易化をねらいとしたプログラム仕様記述法 PSDM を提案する。PSDM においては、プログラム入出力データの性質に着目して、プログラム仕様を記述する。入出力データの性質は以下の3側面から捕えることができる。①入出力データが表す対象世界の情報に関して、その情報の枠組を定めた情報構造。②入出力データ内のデータ項目の並び方、および、データ項目内の記号の並び方等を定めたデータ表現方法。③入出力データの格納ファイルに関して、編成法、および、レコード長等を定めたデータ・アクセス方法。PSDM の特徴は、入出力データの性質に従って、以下の3階層に仕様を分けて記述すること、および、対象世界の情報構造を比較的自然的に記述できる EAR モデルを適用することである。①情報構造を記述する情報層。②データ表現方法を記述するデータ層。③データ・アクセス方法を記述するアクセス層。本論文では、固定長レコード順編成ファイルを入出力するプログラムを対象とした PSDM について述べる。PSDM がプログラム仕様を表すことを示すために、PSDM を限定して適用した仕様記述言語 PSDL、および、実行効率を考慮していない COBOL プログラムを生成するジェネレータ PSD-CG を作成した。

### 1. はじめに

プログラムの機能を表すためのプログラム仕様記述法は、その性質として、理解性、拡張性、記述性、形式性、適用性および最小性<sup>1)</sup>を備えることが望まれる。その全性質を備えた仕様記述法はまだない。実用的な仕様記述法の作成には、仕様記述法の用途に合った性質重視が現実的である。本論文では、プログラム再利用向きの仕様記述法を作成し、プログラム仕様自体の再利用を図ることをねらいとした。そのため、再利用対象仕様の理解性、仕様修正・拡張時の拡張性、仕様作成時の記述性、仕様から COBOL プログラム等を生成できる形式性を特に重視した。

プログラム仕様記述法は、プログラム入出力データの性質に着目したもの、または、プログラム手続きの性質に着目したものに分かれる。前者はさらに、抽象モデルを用いたもの、状態機械モデルを用いたもの、または、代数的なもの等に分かれる。本論文では、理解性、拡張性、記述性および形式性を比較的備えている抽象モデルを用いる仕様記述法<sup>1)</sup>をとった。

プログラム入出力データの性質は、以下の三つの側面から捕えることができる。

①入出力データが表す対象世界の情報に関して、対象世界に存在する事物、および、事物相互の関係等の

観点から、情報の枠組を定めた情報構造。

②入出力データ内の項目の並び方、および、項目内の記号の並び方等を定めたデータ表現方法。

③入出力データの格納ファイルに関して、編成法、レコード長、レコード・キー、呼出し方式、および、入出力の区別等を定めたデータ・アクセス方法。

そして、以上の性質を順次表す情報構造仕様、データ表現仕様およびデータ・アクセス仕様の三つの仕様で構成するプログラム仕様を考えることができる。

情報構造仕様は対象世界に基づいて決まる。データ表現仕様は、データ項目の順序がプログラム処理時間等に影響するので、情報構造仕様のほか、データ処理効率も考慮して決まる。データ・アクセス仕様はデータ表現仕様のほか、ファイルの性質も考慮して決まる。このように、上記の三つの仕様は異なる要因に基づいて階層的に決まるので、プログラム仕様を3階層に分けて記述するのがよい。従来、入出力データの性質に着目した仕様記述言語はある<sup>2)</sup>が、プログラム仕様を3階層に分けて記述し、さらに、対象世界の情報構造を比較的自然的に記述できる EAR (Entity Attribute Relationship) モデル<sup>3),4)</sup>を適用したものは見当たらない。

筆者は、プログラム仕様を以下の3階層に分けて記述すること、および、EAR モデルを適用することを特徴としたプログラム仕様記述法 PSDM (Program Specification Description Method) を考案した。

①情報構造仕様を記述する情報層。

②データ表現仕様を記述するデータ層。

† A Program Specification Description Method PSDM Using Information Structure Description Based on EAR Model by MASAOKI HASHIMOTO (NTT Electrical Communications Laboratories, Nippon Telegraph and Telephone Corporation).  
\*\* 日本電信電話(株) NTT 電気通信研究所

### ③データ・アクセス仕様を記述するアクセス層.

従来は、情報構造仕様とデータ表現仕様が明確には分化していないので、プログラム全体をまとめて理解、拡張または記述する傾向が大きかったが、PSDMでは、情報層からデータ層へ段階的に理解、拡張または記述できるので、理解性、拡張性および記述性は比較的良好なものと考えている。なお、入力データが表す情報から、出力データが表す情報を得る方法を示すための計算の仕様を、情報層に記述できるように、EARモデルを拡張した。PSDMがプログラム仕様を表すことを示すために、PSDMを制限して適用した仕様記述言語 **PSDL** (Program Specification Description Language)、および、実行効率は考慮していないCOBOLプログラムを生成するジェネレータ **PSDCG** (Program Specification Description-COBOL Program Generator) を作成した。

本論文の目的はPSDMを提案することである。本論文では、これまでに研究した、固定長レコード順編成ファイルを入出力するプログラムを対象としたPSDMについて述べる。以下、第2章にプログラム入出力データの性質、第3章にPSDM、PSDLおよびPSDCG、第4章に従来との比較について述べる。

## 2. プログラム入出力データの性質

簡単な請求書作成プログラムを例にとり、その入出力データの性質を図1を用いて、プログラム入出力データの性質について述べる。図1中の“02k”等の3文字のうち、末尾の“k”等のアルファベットは、以下のa)～r)のうち、たとえば、k)で説明する箇所を示す。また、先頭2文字の“02”等の数値は、後述する図3のPSDLプログラムにおいて、たとえば、02行目の文で記述する箇所を示す。

### (1) データ・アクセス方法

図1下部に示すように、a) 売上ファイル、顧客ファイル、請求ファイルのデータ集合があり、そのアクセス単位長、つまり、レコード長はそれぞれ、34バイト(B)、26B、36Bである。b) 売上ファイルと顧客ファイルは入力用、請求ファイルは出力用である。c) 売上ファイル、顧客ファイル、請求ファイルはそれぞれ、売上データ、顧客データ、請求データと対応する。

### (2) データ表現方法

図1中段に示すように、d) 売上データは売上番号や売先等を表す基本項目のデータからなる。e) 売上

レコードには基本項目のデータが順次並んでいる。

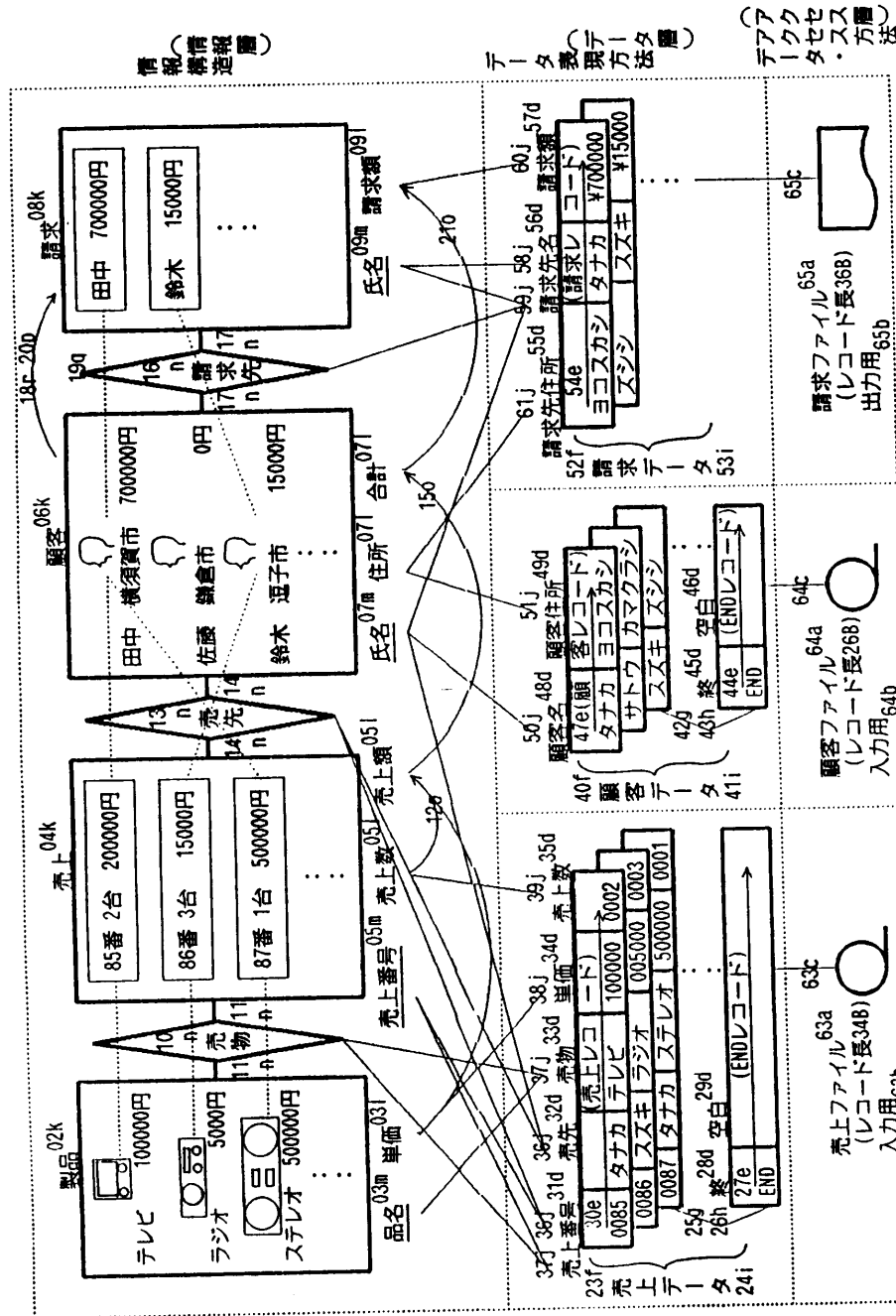
f) 売上レコードは売上ごとに繰り返す。g) 売上レコードのほかにENDレコードもあるので、レコード種別を選択する。h) レコードの最初のデータが“END”であればENDレコードを選択し、その他は売上レコードを選択する。i) ENDレコードが来れば、売上レコードの繰り返しは終わる。j) 対象世界の情報を表す基本項目は、情報構造と対応する。顧客データおよび請求データに関しても、図1からデータ表現方法を理解できる。

### (3) 情報構造

請求書作成プログラムの入出力データは対象世界“販売および請求書作成業務”の情報を表す。図1上部に示すように、入出力データが表す情報には、事物“テレビ”、“85番売上”、“田中”等、および、事物相互の関係“85番売上の売物はテレビであること”、“85番売上の売先は田中であること”等がある。k) 各事物は、事物の種類“製品”、“売上”、“顧客”等に属する。事物の種類は図1上部の太線の四角形で示す。その四角形の中に各事物を示す。1) 事物の性質は、項目“品名”、“単価”等の値“テレビ”、“100000円”等で表す。図1では、事物の種類を示す四角形の下に、各項目名を並べる。その項目名の順に、各事物における各項目の値を並べる。m) 事物を代表する項目“品名”、“売上番号”等を特に事物識別用項目という。図1では、事物の種類を示す四角形の下に、事物識別用項目名を下線付きで示す。n) 事物相互の各関係は、関係の種類“売物”、“売先”等に属する。事物相互の関係の種類は、図1上部の太線の菱形で示す。事物相互の各関係は、事物相互をつなぐ破線で示す。

入力データが表す情報から、出力データが表す情報を得るには計算が必要である。o) 請求の請求額は、図1上部の矢印で示すように、製品の単価と売上の売上数から順次、売上の売上額、および、顧客の合計を計算して得る。p) 事物“請求”は入力データが直接には表しておらず、顧客の合計が正の値ならば、請求が存在する。同時に、q) 顧客と請求相互の関係“請求先”も存在する。r) 事物“請求”は、事物“顧客”に対応してたかだか一つ存在するので、顧客の事物識別用項目“氏名”を請求の事物識別用項目として借用できる。

プログラムが入出力するデータ集合は、データ表現方法を経由して情報構造と対応する。さらに、入力データが表す情報から、出力データが表す情報を得る



(凡例) 02k k 等のアルファベットは、本文第 2 章の説明(a)~(r)のうち、たとえ、k)で説明する箇所を示す。  
02等の数値は、図3のPSDLプログラムにおいて、たとえ、02行目の文で記述する箇所を示す。

図 1 プログラム入出力データの性質  
Fig. 1 Property of input-output-data of program.

ための計算の方法は、情報構造の中に示している。このことから、入出力データの性質は、入出力データ相互の論理的関係、つまり、プログラム機能を表しているものと考えることができる。

### 3. PSDM

プログラム仕様記述法 PSDM の階層構成、各層の内容および適用例について述べる。

#### 3.1 階層構成

前章で述べたように、プログラム機能を表しているものと考えることができる入出力データの性質を、プログラム仕様として記述する。この記述のしかたの概念を定めるためのプログラム仕様記述法 PSDM は、情報層、データ層およびアクセス層で構成し、その各層へ、順次、前章で述べた情報構造、データ表現方法およびデータ・アクセス方法を記述する。

各層をさらに構造および制約に分ける。たとえば、

前章で述べた売上ファイル等のデータ集合、売上番号等のデータの基本項目、製品等の事物の種類、単価等の項目のように、仕様に現れる対象を型として定めたものを構造という。一方、たとえば、製品の単価と売上の売上数から計算して、売上の売上額を得ることは、型として定めた項目“単価”、“売上数”および“売上額”に課した条件を表す。このように、型に課した条件を制約という。

各層の構造および制約を、さらに図2(2)に示す記述要素に分ける。図2(2)、(3)に示すように、各記述要素は入出力データの各性質へ対応する。次節以降に PSDM の記述要素について述べる。なお、3.5 節に述べるように、PSDM で定める概念を適用したプログラム仕様記述言語 PSDL を作成しており、図3に示す PSDL 記述の請求書作成プログラム仕様中の文を、記述要素の適用例として引用する。同図の 01~21 行が情報層、22~61 行がデータ層、62~65 行がアクセス層を記述したものである。PSDM の定式化に関しては文献 5) を参照されたい。

### 3.2 情報層

#### 3.2.1 構造

情報構造記述のために適用した EAR モデルでは、対象世界に存在する事物を主体(Entity)といい、事物相互の関係を関連(Relationship)という。主体および

関連の型を、以下の記述要素で記述する。

#### (1) 主体型、属性および主キー

主体型は主体の集合を定める型である。属性は主体の性質を表す項目であり、同じ型に属する主体は同じ属性の組を持つ。以後、ある主体型のある属性を“製品/品名”のように、“主体型/属性”の形で書く。個々の主体の性質は属性の値で表す。各属性ごとに属性値の定義域を定める。属性は一つまたは複数の値を持つ。後者の場合、多値属性という。主体識別用の属性を主キーという。主体型名および主キー値により、主体を一意に識別できる。主キーは一つまたは複数の属性からなる。図1に示す製品、売上および顧客のように、主体型がその属性のみからなる主キーを持つならば、正規主体型という。請求のように、主体型がその属性のみからなる主キーを持たず、その主体型の主キーとなる属性を、他の主体型の主キー属性から借用していれば、弱主体型という。

たとえば、正規主体型“製品”を図3の 02 行目の entity-type 文、その属性を 03 行目の description 文で書く。{ }中の HINMEI と YEN は属性値の定義域である。str は文字列、num は数値の定義域を表す。id は主キーの属性を表す。id 省略の属性は主キーでない。弱主体型“請求”は 08 行目の (weak) 付きの文、その属性は 09 行目の文で書く。請求は主キー“氏名”

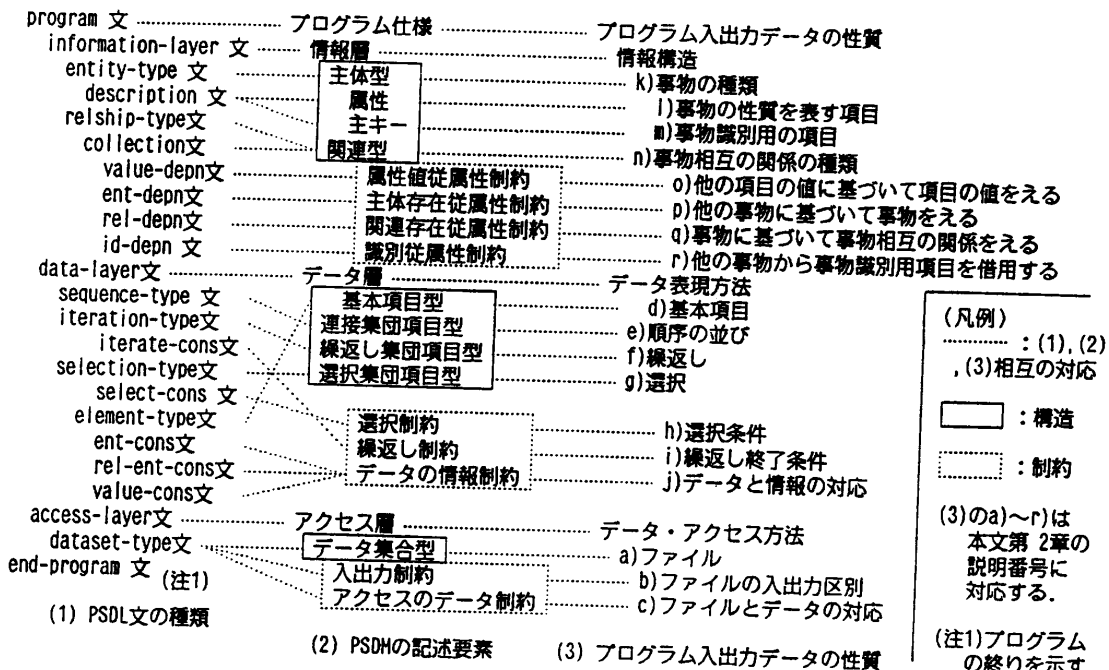


図2 PSDM の概念  
Fig. 2 Outline of PSDM.

```

00 program SEIKYUSYO;
01   information-layer;
02     entity-type SEIHIN;
03       description HINMEI(HINMEI,str,id),TANKA(YEN,num,);
04     entity-type URIAGE;
05       description URIAGEBANGO(BAN,num,id),URAGESU(KO,num,),URAGEGAKU(YEN,num,);
06     entity-type KOKYAKU;
07       description SHIMEI(SHIMEI,str,id),JUSYO(JUSYO,str,),GOKEI(YEN,num,);
08     entity-type(weak) SEIKYU;
09       description SEIKYUGAKU(YEN,num,);
10     relship-type URIMONO;
11       collection /URIAGE,/SEIHIN;
12       value-depn /URIAGE/URAGEGAKU <- FMULT:/URIAGE/URAGESU,/SEIHIN/TANKA;
13     relship-type URISAKI;
14       collection /URIAGE,/KOKYAKU;
15       value-depn /KOKYAKU/GOKEI <- FSUM:/URIAGE/URAGEGAKU;
16     relship-type(weak) SEIKYUSAKI;
17       collection /SEIKYU,/KOKYAKU;
18       id-depn /SEIKYU <- /KOKYAKU;
19       rel-depn(int) PSEIKYUSAKI:/KOKYAKU/SHIMEI,/SEIKYU/(KOKYAKU/SHIMEI);
20       ent-depn /SEIKYU <- /KOKYAKU if PSEIKYU:/KOKYAKU/GOKEI;
21       value-depn /SEIKYU/SEIKYUGAKU <- FSAME:/KOKYAKU/GOKEI;
22   data-layer;
23     iteration-type(root) URIAGE-DATA(U)::=URIAGE-R;
24     iterate-cons(data) U-END(U)= "END";
25     selection-type(rec) URIAGE-R::=U-EG(= "END"),URIAGE-O(NOT = "END");
26     select-cons(data) U-END(U);
27     sequence-type U-EG::=U-END.U-FILL;
28     element-type U-END PIC X(3);
29     element-type U-FILL PIC X(31);
30     sequence-type URIAGE-O::=U-BANGO,U-SAKI,U-MONO,U-TANKA,U-SU;
31     element-type U-BANGO PIC 9(4);
32     element-type U-SAKI PIC X(10);
33     element-type U-MONO PIC X(10);
34     element-type U-TANKA PIC 9(6);
35     element-type U-SU PIC 9(4);
36     rel-ent-cons(info) URISAKI//URIAGE <- U-BANGO(U)/KOKYAKU <- U-SAKI(U);
37     rel-ent-cons(info) URIMONO//URIAGE <- U-BANGO(U)/SEIHIN <- U-MONO(U);
38     value-cons(info) SEIHIN/TANKA[U-MONO(U)] <- U-TANKA(U);
39     value-cons(info) URIAGE/URAGESU[U-BANGO(U)] <- U-SU(U);
40     iteration-type(root) KOKYAKU-DATA(Y)::=KOKYAKU-R;
41     iterate-cons(data) K-END(Y)= "END";
42     selection-type(rec) KOKYAKU-R::=K-EG(= "END"),KOKYAKU-O(NOT = "END");
43     select-cons(data) K-END(Y);
44     sequence-type K-EG::=K-END,K-FILL;
45     element-type K-END PIC X(3);
46     element-type K-FILL PIC X(23);
47     sequence-type KOKYAKU-O::=K-MEI,K-JUSYO;
48     element-type K-MEI PIC X(10);
49     element-type K-JUSYO PIC X(16);
50     ent-cons(info) KOKYAKU <- K-MEI(Y);
51     value-cons(info) KOKYAKU/JUSYO[K-MEI(Y)] <- K-JUSYO(Y);
52     iteration-type(root) SEIKYU-DATA(S)::=SEIKYU-R;
53     iterate-cons(entity) S(SEIKYU);
54     sequence-type(rec) SEIKYU-R::=S-JUSYO,S-MEI,S-GAKU;
55     element-type S-JUSYO PIC X(16);
56     element-type S-MEI PIC X(10);
57     element-type S-GAKU PIC 9(9);
58     ent-cons(data) S-MEI(S) <- SEIKYU;
59     rel-ent-cons(data) S-MEI(S) <- SEIKYUSAKI//KOKYAKU where /SEIKYU <- S-MEI(S);
60     value-cons(data) S-GAKU(S) <- SEIKYU/SEIKYUGAKU[S-MEI(S)];
61     value-cons(data) S-JUSYO(S) <- KOKYAKU/JUSYO[S-MEI(S)];
62   access-layer;
63     dataset-type(file) URIAGE-F,input,34,URIAGE-DATA;
64     dataset-type(file) KOKYAKUF,input,26,KOKYAKU-DATA;
65     dataset-type(file) SEIKYU-F,output,36,SEIKYU-DATA;
66 end-program;

```

図 3 PSDL プログラム仕様  
Fig. 3 PSDL Program Specification.

を顧客から借用しているので、09 行目には id 付きの属性はない。以後、弱主体型が借用する主キーの属性を、たとえば、“(顧客/氏名)”と書く。

## (2) 関連型

関連型は関連の集合を定める型である。個々の関連

は、関連で関係付けた主体の主キー値の組で表す。本論文では、関連は属性を持たない<sup>4)</sup>。図 1 に示す売物および売先のように、正規主体型の間に定義した関連型を正規関連型という。請求先のように、弱主体型も含めた主体型の間に定義した関連型を弱関連型とい

う。請求書作成プログラムの例では2項の関連型を用いたが、一般的には3項以上も使用できる。

同じ主体型の間で定義した関連型もある。たとえば、親子関係は同じ主体型“人”の間の関連型である。親子関係を持つ二つの主体は同じ主体型“人”に属するが、親子関係において、一方の主体は親の機能を果たし、他方の主体は子の機能を果たす。この親や子のように、ある関連型において主体が果たす機能を役割という。以後、ある関連型においてある役割を果たす主体の集合を“親/人”のように、“役割/主体型”の形で書く。その主体の属性は“親/人/氏名”のように、“役割/主体型/属性”の形で書く。関連型の定義に用いた主体型が異なる場合、“/人”、“/人/氏名”のように、役割を省略してもよい。

たとえば、正規関連型“売物”を図3の10行目の relship-type 文、その定義に用いた主体型を11行目の collection 文で書く。図3では役割をすべて省略した。弱関連型“請求先”は16行目の(weak)付きの文、その定義に用いた主体型を17行目の文で書く。

### 3.2.2 制 約

主体型および関連型の制約について述べる。

#### (1) 属性値従属性制約

本制約は、ある主体型に属する主体の属性値を、その主体の他の属性値、および、ある関連型に属する関連でその主体とつながれた主体の属性値から得るための計算の仕様を定める。そのため、本制約では一つの関連型、値を得る対象の一つの役割/主体型/属性、計算のしかたを示す一つの関数、この関数のパラメータとなるいくつかの役割/主体型/属性を指定する。この制約は上記の関連型へ附随する条件である。

たとえば、主体“売上”の売上額を得るには、関連型“売物”、/売上/売上額、積をとる関数、/売上/売上数、/製品/単価を指定する。この制約は、関連型“売物”に属する関連で一対一に対応付けられた二つの主体“売上”と“製品”に関して、売上数と単価の積をとり、この値を売上額とすることを定める。図3では、12行目の value-depn 文のように、附随先の関連型に続けて書く。FMULT は積をとる関数である。主体“顧客”の合計を得るための制約では、関連型“売先”に属する関連で一つの主体“顧客”と一対多に対応付けられた  $n(n \geq 0)$  個の主体“売上”に関して、売上額の総和をとり、この値を顧客の合計とすることを定める。この制約は図3の15行目の文で書く。FSUM は総和をとる関数である。21行目の FSAME

は、パラメータの属性の値と同じ値を表す関数である。なお、同じ主体の属性のみの間に成り立つ属性値制約は、主体型に附随する条件である。

#### (2) 関連存在従属性制約

本制約は、いくつかの主体の存在に基づいて、その主体間に存在する一つの関連を得るための計算の仕様を定める。そのため、本制約では一つの関連型、主体相互の対応条件を示す一つの述語、この述語のパラメータとなるいくつかの役割/主体型/属性、関連の存在条件を示す一つの述語、この述語のパラメータとなるいくつかの役割/主体型/属性を指定する。本制約は上記の関連型へ附随する条件である。主体相互の対応条件を示す述語のパラメータには、少なくとも、本制約が附随する関連型の定義に用いたすべての主体型の主キー属性を指定する。また、その主体型の任意の属性を、関連の存在条件を示す述語のパラメータに指定する。本制約が附随する関連型の定義に用いたすべての主体型から一つずつの任意の主体を取り出し、その主体に対して両述語が真となれば、その主体間に関連が存在する。その他の場合、関連は存在しない。関連の存在条件を示す述語が恒真の場合、その述語を省略する。

たとえば、関連“請求先”を得るには、関連型“請求先”、/顧客/氏名、/請求/(顧客/氏名)、この両主キー属性の値が等しければ真となる主体相互の対応条件を示す述語を指定する。この制約は、主体“顧客”の主キー“氏名”と、主体“請求”の主キー“(顧客/氏名)”の値が等しければ、その主体間に関連“請求先”が存在することを定める。図3では、19行目の rel-depn 文のように、本制約が附随する関連型に続けて書く。PSEIKYUSAKI は主体相互の対応条件を示す述語である。主体相互の対応条件を示すには、主体の主キー値の組を列挙する外延的方法、および、主キー値の組が満たす述語を用いる内包的方法がある。上記の例は後者であり、rel-depn 文に(int)を付ける。

#### (3) 主体存在従属性制約

本制約は、いくつかの主体の存在に基づいて、他に存在するいくつかの主体を得るための計算の仕様を定める。そのため、本制約では一つの関連型、主体を得る対象のいくつかの役割/主体型、その主体の存在の基となる主体が属するいくつかの役割/主体型、主体の存在条件を示す一つの述語、この述語のパラメータとなるいくつかの役割/主体型/属性を指定する。本制約は上記の関連型へ附随する条件であり、関連存在

従属性制約と対で定める。本制約が附随する関連型の定義に用いたすべての主体型は、主体を得る対象の主体型、または、主体存在の基となる主体型として指定する。主体の存在条件を示す述語のパラメータには、主体存在の基となる主体型の任意の属性を指定する。

主体存在の基となる主体型から一つずつ取出した任意の主体に対して、主体の存在条件を示す述語が真となれば、①その主体に対して関連存在従属性制約を満たす主キー値を持つ主体が、主体を得る対象の主体型に存在し、②主体存在の基となった主体と、新たに得た主体との間に関連が存在する。①の関連存在従属性制約を満たす主キー値は、その関連存在従属性制約における主体相互の対応条件を示す述語が真となることを方程式と見なし、その方程式へ、主体存在の基となる主体の属性値を代入し、主体を得る対象の主体型の属性に関して、方程式を解くことにより、得ることができる。なお、方程式は解くことができ、その解法は与えられているものとする。関連存在従属性制約を満たす主キー値が複数あれば、その主キー値を持つ複数の主体が存在する。主体の存在条件を示す述語が恒真の場合、その述語を省略する。

たとえば、主体“顧客”の存在に基づいて主体“請求”を得るには、関連型“請求先”、/請求、/顧客、/顧客/合計、その値が正ならば真となる主体存在条件を示す述語を指定する。この制約は、主体“顧客”の合計が正の値ならば、①顧客の氏名と同じ値を主キー値に持つ主体“請求”が存在し、②その主体“顧客”と“請求”の間に関連“請求先”が存在することを定める。図3では、20行目の ent-depn 文のように、本制約が附随する関連型に続けて書く。PSEIKYU は主体の存在条件を示す述語である。なお、他の主体とは無関係に、主体の属性値を列挙する外延的方法、または、主体の属性値が満たす述語を示す内包的方法で、主体の存在を定めることもできる。この制約は主体型に附随する条件である。

#### (4) 識別従属性制約

本制約は、ある弱主体型の主キー属性を、その弱主体型との間に関連型を定義した主体型のうち、どの主体型の主キーから借用するかを定める。そのため、本制約では一つの関連型、一つの役割/弱主体型、主キー借用先のいくつかの役割/主体型を指定する。本制約は上記の関連型に附随する条件である。たとえば、/請求の主キー属性“(顧客/氏名)”を、/顧客から借用するための制約は、図3の18行目の id-depn

文のように、附随先の関連型に続けて書く。弱主体型の主キー属性が複数ある場合、その一部の属性を他の主体型から借用できる。関連型が3項以上の場合、複数の主体型から主キーを借用できる。

### 3.3 データ層

#### 3.3.1 構造

以下のデータ型で、データ項目の並び方を定める。

##### (1) 基本項目型

それ以上分解しないデータは本型に属する。本型のデータは記号の並びからなる。記号の並び方はCOBOLの PICTURE 句で定めるものとする。データ長は本型ごとに固定である。たとえば、売上データ中の本型“売上番号”は、図3の31行目の element-type 文で書く。

##### (2) 接続集団項目型

順序を持って一つずつ存在する二つ以上の型に属するデータの並びは本型に属する。たとえば、売上レコード中の基本項目の順序の並びは、図3の30行目の sequence-type 文で書く。

##### (3) 繰り返し集団項目型

一つ以上繰り返して存在する同じ型に属するデータの並びは本型に属する。本型には、正整数 1, 2, 3, … をとる指標が附随する。たとえば、売上レコードの繰り返しは、図3の23行目の iteration-type 文で書く。U は指標である。36行目の U-BANGO(U) は、U-BANGO のデータの繰り返しのうち、U 番目のデータを指す。

##### (4) 選択集団項目型

二つ以上存在しうる型に属するデータのうち、いずれか一つがただ一つ存在するデータの並びは本型に属する。たとえば、売上レコードまたは END レコードの選択は、図3の25行目の selection-type 文で書く。

データ型は階層的に定義する。PSDM では、売上データ、顧客データおよび請求データがそれぞれ、独立した一つの木構造となるように、データ型群を定義する。基本項目型のデータは、①たとえば、 $D_1, D_2(I_1), D_3, D_4(I_2)$  のように、上位のデータ型名および指標名で修飾した基本項目型名、また、②  $D_4(I_1, I_2)$  のように、上位の繰り返し集団項目型にも附随するすべての指標が付いた一意の基本項目型名で識別できる。ここで、 $D_1, D_3$  は繰り返し集団項目型、 $I_1, I_2$  はそれぞれ、 $D_1, D_3$  に附随する指標、 $D_2$  は接続集団項目型、 $D_4$  は基本項目型である。本論文では②を用いる。②を基本項目型データ識別項といい、同じ木構造内にある指標

からなる算術式も指標として書いてもよい。

### 3.3.2 制 約

データ型の制約について述べる。

#### (1) データの情報制約

情報を表す基本項目型および指標は、情報層で定めた型に属する主体および関連を表すものと考えることができる。そのため、情報を表す基本項目型データ識別項および指標と、情報層の型との対応を以下のように定める。

①ある主体型に属する主体を表す基本項目型データ識別項および指標と、その主体型の主キー属性を対応させる。たとえば、図3の50行目の `ent-cons` 文は、`K-MEI(Y)` が主体型“顧客”の主体を表すことを定める。

②ある関連型に属する関連、および、その関連型の定義に用いた役割/主体型に属する主体を表す基本項目型データ識別項および指標と、その主体の主キー属性を対応させる。たとえば、36行目の `rel-ent-cons` 文は、二つの矢印の右辺の `U-BANGO(U)` と `U-SAKI(U)` が、左辺の関連型“売先”の関連、および、/売上と/顧客に属する主体を表すことを定める。

③ある主体型に属する主体のある属性の値を表す基本項目型データ識別項および指標と、その属性を対応させる。たとえば、38行目の `value-cons` 文は、矢印の右辺の `U-TANKA(U)` が、左辺の `U-MONO(U)` で定まる主キー値を持つ主体“製品”の単価の属性値を表すことを定める。

データ型と、情報層で定めた型との対応を定めるための①②③は、データがどのような情報を表すかを定めているのでデータの情報制約という。①②③をそれぞれ、主体存在の情報制約、関連および主体存在の情報制約、属性値の情報制約という。基本項目型および指標はたかだか一つの属性に対応するものとする。

基本項目型のデータ間には並びの順序関係がある。しかし、順序を表す関連型を定義していない主体間には順序関係はない。このため、出力データに関する上記の①②では、同じ主体型に属する複数の主体を、出力データへ対応付ける順序が問題となる。①では任意の順序で対応付ける。たとえば、58行目の制約では、主体“請求”を任意の順序で `S-MEI(S)` のデータへ対応付ける。②では他の情報制約、または、他の指標値に対する同じ情報制約の適用により、②で指定した一部の主体型に関して、その主体をすでに出力データへ対応付けているものとする。残りの主体型に関して、

その主体を任意の順序で新たに出力データへ対応付ける。たとえば、59行目の `where` の右辺はすでに対応付けた主体型、左辺は新たに対応付ける主体型を表す。

#### (2) 繰り返し制約

繰り返し集団項目型において、データの繰り返し終了条件を以下のいずれかで示す。

①ある主体型の主キーに対応する基本項目型データ識別項または指標に関して、その値を主キー値とする主体のある属性の値が示す回数だけ繰り返す。

②ある主体型に属する主体の個数だけ繰り返す。

③ある基本項目型のデータまたは指標値に対して、ある述語が真になれば、そのデータまたは指標値の直前で繰り返しが終わる。

図3の24行目の `(data)` 付きの `iterate-cons` 文は③の例であり、`U-END(U)` が“END”になれば、繰り返しが終わる。53行目の `(entity)` 付きの文は②の例である。基本項目型データ識別項中の指標には算術式も書けるので、ゼロ、負数または繰り返し数以上の指標値も現れる。その場合、(1)②の情報制約において、正しい指標値を持つ基本項目型データ識別項および指標に対応する主体が存在するほか、情報制約の適用はないものとする。

#### (3) 選択制約

選択集団項目型で選択する各データ型へ述語を対応させる。その述語を以下の値へ作用させ、ただ一つ真となる述語へ対応するデータ型を選択する。

①ある基本項目型データ識別項または指標の値。

②ある主体型の主キーに対応する基本項目型データ識別項または指標に関して、その値を主キー値とする主体のある属性の値。

③ある主体型に属する主体の個数。

図3の26行目の `(data)` 付きの `select-cons` 文は①の例であり、`U-END(U)` のデータが“END”ならば、25行目の `U-EG` を選択し、“END”でなければ、`URIAGE-O` を選択する。

データの情報制約において、入出力データと直接には対応しない主体を指す場合、基本項目型データ識別項や指標の代りに、定数で主キー値を書いてもよい。

### 3.4 アクセス層

ファイルに相当するデータ集合の型名、およびレコード長に相当するアクセス単位長を、データ集合型として定めたものがアクセス層の構造である。一つのデータ集合中のデータ群は、レコードの順序で結合し



た一つの記号の並びと見なし、データ層の一つの木構造に対応させる。この対応をアクセスのデータ制約という。アクセスのデータ制約、および、データ集合の入出力の区別を定めたものがアクセス層の制約である。たとえば、売上ファイルに関しては、図3の63行目の (file) 付きの dataset-type 文で書く。input/output は入出力の区別、数値はアクセス単位長をバイトで表す。

### 3.5 適用例

おもに以下の制限を設けて、PSDM を適用したプログラム仕様記述言語 PSDL を作成した。

①主体は一つの主キー属性を持つ。多値属性はない。②属性値の定義域は数値または文字列である。③データの木構造の根となるデータ型、および、レコードに対応するデータ型の書き方を特別に設ける。前者の PSDL 文には (root)、後者には (rec) を付ける。④一つの基本項目型データ識別項中の指標はたかだか一つとする。⑤入力および出力別に情報制約の書き方を分ける。前者の PSDL 文には (info)、後者には (data) を付ける。⑥関数および述語の定義手段は設けない。⑦たとえば、 $E_1, E_2$  を主体型、 $A_1, A_2$  を属性、 $F_1, F_2$  を関数とした場合、以下の二つの属性値従属性制約

$$\text{value-depn } /E_1/A_1 \leftarrow F_1: /E_2/A_2;$$

$$\text{value-depn } /E_2/A_2 \leftarrow F_2: /E_1/A_1;$$

においては、属性  $A_1$  の値を、属性  $A_1$  の値自体から順次、関数  $F_2, F_1$  で計算して得るように定めている。このように、ある属性の値を、その属性の値自体から計算して得るように定めた制約群はない。

PSDL プログラムから、実行効率は考慮していない COBOL プログラムを生成するジェネレータ PSD-CG を作成した。そして、図3の PSDL プログラムから COBOL プログラムを生成し、図1に示す入出力データを処理した。このことは、PSDM がプログラムを生成できる形式性を備えていることを示している。なお、PSD-CG のステップ展開率は約10倍であった。PSD-CG が生成するプログラムの効率化に関しては、現在研究中である。

## 4. 従来との比較

COBOL, MODEL<sup>2)</sup>, 森沢<sup>6)</sup> と PSDL の比較を述べる。COBOL は手続き言語、PSDL は非手続き言語である。入出力データの性質に着目した非手続き言語 MODEL では、出力データ項目を直接または中間データを経由して、算術式等により入力データ項目に関係

付ける。このため、PSDL の情報構造仕様とデータ表現仕様が MODEL では分化していない。森沢のプログラム仕様記述法では、入出力データの構文を Backus-Naur Form で記述し、この構文記述へ従属する形で、入出力データ相互の関係等を表す意味記述を行う。PSDL では構文より意味の方が仕様の本質を表すものと考えて、構文記述とは独立して記述できる情報構造仕様を設けた。情報構造仕様、および、データ表現仕様中のデータの情報制約が、意味記述の役割を果たす。

従来の言語では、情報構造仕様とデータ表現仕様が明確には分化していないので、プログラム全体をまとめて理解、拡張または記述する傾向が大きい。PSDL では、情報層からデータ層へ段階的に理解、拡張または記述できる。さらに、対象世界の情報構造を比較的自然に記述できる EAR モデルを適用した。以上のことは、PSDM で重視した理解性、拡張性および記述性へ比較的有效である。MODEL では、繰り返しデータ相互を対応付ける指標の取り扱いが困難とされている<sup>2)</sup>。PSDL では、繰り返しデータ相互の間に、情報層の型や制約を介在できるので、そのような困難は軽減されている。

適用性に関しては、QBE<sup>7)</sup> で記述できる計算の仕様を、情報層で記述できるものと考えている。データ層ではレベル番号、OCCURS 句、REDEFINES 句を用いた COBOL のデータ項目の並びを記述できる。COBOL で直接記述した請求書作成プログラムは約90ステップ(s)となった。PSDL では図3のように約70sとなったので、最小性が特に悪いことはないものと考えている。

## 5. おわりに

プログラム仕様記述法 PSDM を報告した。プログラム仕様を3階層に分けて記述すること、および、対象世界の情報構造を比較的自然に記述できる EAR モデルを適用したことは、PSDM で重視した理解性、拡張性および記述性に比較的有效である。PSDM はプログラムを生成できる形式性も備えている。COBOL と比較して、最小性が特に悪いことはないものと考えている。情報層の適用性は QBE に相当し、データ層の適用性は COBOL に相当するものと考えている。拡張 EAR モデルの定式化に関しては、別途報告した<sup>8)</sup>。

プログラム再利用技術の次は、プログラム量産化技術の研究が必須である。ある適用領域に関する知識

を、PSDMにおける主体型、関連型および制約等の形で知識ベース化し、その領域の多数のプログラムを自動合成する量産化技術へ本研究を発展させる。

謝辞 本研究をご指導、ご支援いただいた伊吹公夫元特別研究室長、高村真司元データ処理研究部長、橋本昭洋知能処理研究部長、花田收悦ソフトウェア生産技術研究所長、徳山五郎図書館情報大学教授へ厚くお礼申し上げます。

### 参 考 文 献

- 1) Liskov, B. H. and Zilles, S. N.: Specification Techniques for Data Abstractions, *IEEE TOSE*, Vol. SE-1, No. 1, pp. 7-19 (1975).
- 2) Prywes, N. S.: Automatic Generation of Computer Programs, *Advance in Computer*, Vol. 16, pp. 57-125, Academic Press, New York (1977).
- 3) Chen, P. P.: The Entity-Relationship Model—Toward a Unified View of Data, *ACM TODS*, Vol. 1, No. 1, pp. 9-36 (1976).
- 4) van Griethuysen, J. J. et al. (ed.): Appendix D. The Entity-Attribute-Relationship Approaches, Concepts and Terminology for the Conceptual Schema and the Information Base,

ISO TC 97/SC 5-N 695 (1982).

- 5) 橋本正明: EAR モデルに基づく情報構造記述を用いたプログラム仕様記述法 PSDM, 情報処理学会ソフトウェア工学研究会資料, 40-11 (1985).
- 6) 森沢好臣: 構文と意味の記述によるプログラムの作成, 情報処理, Vol. 25, No. 11, pp. 1255-1260 (1984).
- 7) Zloof, M. M.: Query by Example: a Data Base Language, *IBM Syst. J.*, No. 4, pp. 324-343 (1977).
- 8) 橋本正明: プログラム仕様記述のための計算指向 EAR モデル, 情報処理学会論文誌, Vol. 27, No. 3, pp. 330-338 (1986).

(昭和 60 年 5 月 23 日受付)

(昭和 61 年 4 月 17 日採録)



橋本 正明 (正会員)

昭和 43 年九州大学工学部電子工学科卒業。昭和 45 年同大学院修士課程修了。同年日本電信電話公社電気通信研究所入社。現在 NTT 情報通信処理研究所勤務。これまでおも

にオペレーティング・システム, 中間言語マシン, データベース設計支援ツール, プログラム仕様記述法の研究実用化に従事。電子通信学会会員。