

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

電電公社 横須賀研究所 橋本正明

1. はじめに

ソフトウェアの仕様記述法の適用領域は、①システムに対するユーザ要求を表すシステム要求仕様、②ソフトウェアを構成するモジュール相互の関係を表すソフトウェア構造仕様、③モジュールの機能を表すプログラム仕様の領域に分けることができる。③のプログラム仕様記述法に望まれる性質として、形式性、記述性、理解性、最小性、適用性および拡張性がある¹⁾。この全性質を同時に備えた仕様記述法はまだない。実用的な仕様記述法を作るには、仕様記述法の用途に必要な性質を重視した方法を取るのが現実的である。筆者は、部分変更を伴うソフトウェアの再利用のため、仕様記述言語で記述されたコードの再利用を容易にすることを狙った。このため、理解性、拡張性、記述性および形式性を重視して、抽象モデルを用いる方法を取った。この場合、適用性が抽象モデルに大きく依存するので、仕様を階層に分けて、各階層毎に適した抽象モデルを選ぶことにより適用性に留意した。

仕様記述法には、データ又は手続きに着目するものがある。本論文の仕様記述法が属する前者では、データに着目した仕様を用いてプログラム仕様を記述する。プログラムで処理されるデータ本来の性質は、以下の3側面から捕えることができる。①データ上に情報として表される対象世界に存在する事物の種類、及び、事物相互の関係の種類等を定める情報構造。②データ項目の並び方、データ項目内の記号の並び方、及び、データ項目と①との対応を定めるデータ表現方法。③プログラムとの間で、②で表現されるデータを授受するファイル等のデータ集合、授/受の区別、及び、レコード長等の授受単位を定めるデータ・アクセス方法。このため、プログラム仕様は、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3種の仕様からなることができる。

情報構造仕様は対象世界に基づいて決められる。データ表現仕様は、データ項目の並びの順序がプログラムの処理時間等に影響することから、情報構造仕様の外に、データ処理効率も考慮して決められる。データ・アクセス仕様は、データ表現仕様の外に、データ集合の性質も考慮して決められる。このように、各仕様は相異なる要因に基づいて階層的に決められるので、プログラム仕様を階層に分けて記述できる。従来、データに着目した仕様を用いた言語²⁾³⁾はあるが、プログラム仕様を上記の3種の仕様に分け、情報構造を自然に表せるEAR(Entity Attribute Relationship)モデル⁴⁾⁵⁾を用いたものは見当たらない。

筆者は、プログラム仕様記述の観点から、EARモデルにおける制約(constraint)を定式化することにより、以下の3階層にプログラム仕様を分けて記述すること、及び、EARモデルを用いることを特徴とした仕様記述法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。③データ・アクセス仕様を記述するアクセス層。本論文の目的はPSDMの内容を報告することである。本論文では、EARモデルにお

ける関連には属性はなく⁵⁾、主体の属性は一価関数に限定。又、データ・アクセス方法は固定長レコード順アクセスに限定した。以下、第2章にPSDM、第3章に適用例PSDL(Program Specification Description Language)を述べる。用語は主に文献⁶⁾⁷⁾による。

2. PSDM

概要および各階層の詳細を述べる。

2.1 概要

例えば、売上データと顧客データを入力して、請求データを出力する請求書作成プログラムの売上データに関して、情報構造、データ表現方法およびデータ・アクセス方法の3側面から捕えられた図1(a)に示す性質を各々、図1(b)に示す情報層、データ層およびアクセス層の仕様として記述する。顧客データと請求データに関しても同様にする。データ層およびアクセス層の仕様は売上、顧客および請求の各データに分けて記述する。しかし、同じプログラムで処理される各データ上に表される情報は、相互に関係を持つので、情報層の仕様はまとめて記述する。このように、入出力データ相互の関係は情報層で記述できる。

各階層の仕様は構造および制約へ分けて記述する。

(1) 構造(structure)

構造は、仕様を表される以下の対象を型(type)として記述したものである。①情報層では、対象世界に存在する事物の種類、及び、事物相互の関係の種類。②データ層では、集団項目、基本項目、及び、基本項目内の記号の並び方。③アクセス層では、データ集合、データ授/受の区別および授受単位。①はEARモデルに基づいて記述する。②ではCOBOL同様に、集団項目と基本項目の階層でデータ項目の並び方を記述する。

(2) 制約

制約は、型に課せられる以下の条件を記述したものである。①情報層では、型に属する事物の存在等に関する条件。②データ層では、同じ型のデータ項目の繰返し回数等に関する条件、及び、基本項目の型と情報層内の型との対応。③アクセス層では、データ集合の型とデータ層内の型との対応。このように、各階層間の対応は制約で記述できる。

2.2 情報層

2.2.1 構造

対象世界に存在する事物、及び、事物相互の関係を各々、主体および関連として捕える。例えば、顧客“田中太郎”や売上番号85番の売上は主体であり、85番の売上の売上先が田中太郎であるという関係は関連である。以下、構造について述べる。

(1) 主体(entity)及び主体型

各主体は顧客や売上等、主体の種類を表す主体型に属する。主体型を E_j 、主体を e_{ij} と書く。型は集合であり、 e_{ij} が E_j に属することを $e_{ij} \in E_j$ と書く。

(2) 関連(relation)、関連型および役(role)

関係(relation)は、2つ以上の集合間の要素の対応付けを表す集合であり、主体型 E_{ij} ($j=1, \dots, n, n \geq 2$)間の関係を関連型という。関連型の要素($e_{i1h}, \dots$

$(e_{ijh}) (e_{ijh} \in E_{ij}, j=1, \dots, n)$ を関連という。関連型を R_i 、関連を r_{ijh} 、 r_{ijh} が R_i に属することは $r_{ijh} \in R_i$ と書く。 $E_{ij} (j=1, \dots, n)$ の中に同じ型があっても良い。例えば、親子は同じ型“人”の主体相互の関連である。親子の関連の中の2つの主体は各々、親と子の役を持つ。役は L_{ij} 、役 L_{ij} を持つ E_{ij} の主体の集合を L_{ij}/E_{ij} と書く。

(3) 属性(attribute), 属性値および定義域(domain)

属性値の組で主体の性質を表す。各属性値は決められた定義域に属する。定義域を V_0 、属性値を v_{op} と書く。属性は主体型から定義域への一価関数 A_{ijk} で表す。例えば、顧客の属性“氏名”は、主体型“顧客”から文字列の集合である定義域への関数である。

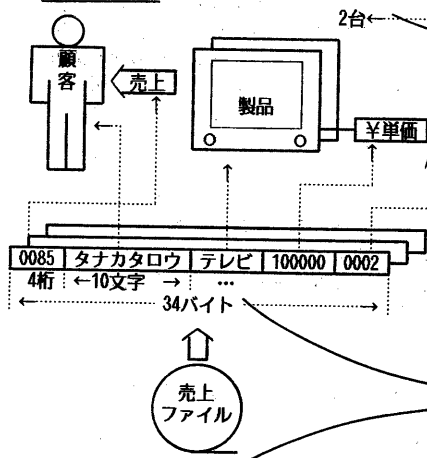
$e_{ijh} \in E_{ij}$ の A_{ijk} の値は $A_{ijk}(e_{ijh})$ と書く。同じ型の主体は同じ属性を持つ。 E_{ij} の主体の属性は E_{ij}/A_{ijk} と書く。役 L_{ij} を持つ E_{ij} の主体の属性は $L_{ij}/E_{ij}/A_{ijk}$ 又は $L_{ij}/E_{ij}/(A_{ij1}, \dots, A_{ijm})$ と書く。

(4) 主体および関連の主キー(primary key)

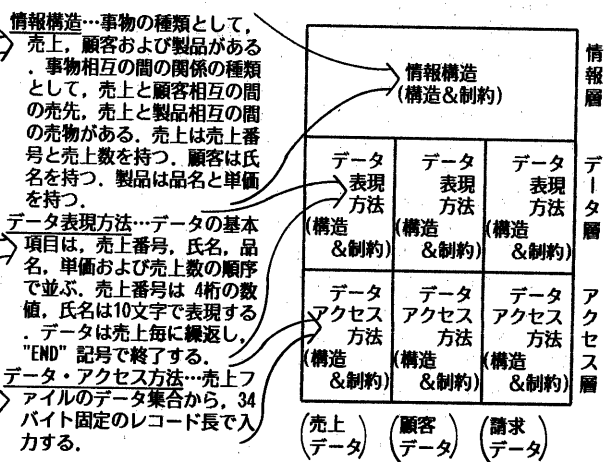
主体型の中で、各主体を識別するために用いられる属性を、主体のキーという。例えば、氏名は顧客のキーである。属性の組からなるキーもある。1つの主体に複数のキーがある場合は、1つを選んで主キーとする。主体型に、その主体の属性のみからなるキーがある場合、正規(regular)主体型という。主体型に、その主体の属性の外に、他の型の主体の主キーも含むキーがある場合、弱(weak)主体型という。弱主体型の主体は、そのキーに含まれる主キーを持つ他の主体、及び、その主体との間の関連を用いて識別できる。関連の主キーは、関連に含まれる主体の主キーの組からなる。例えば、売上番号と氏名は売上先の主キーである。関連に含まれる全ての主体が正規主体型の場合は、その関連型を正規関連型という。少なくとも1つの弱主体型が含まれる場合は弱関連型という。

図2の情報層に示すように、ER図(entity-relationship diagram)⁴⁾で構造を表現できる。PSDMでは、プログラムで処理されるデータは、個々の主体や関連を列挙する形で情報を表しているものとする。

2.2.2 制約



(a) 売上データの性質



(b) プログラム仕様の階層

図1 PSDMの概念

以下の種類の制約を用いる。

(1) 識別従属性

相異なる $L_{ij}/E_{ij} (j=1, \dots, n, n \geq 2)$ 間の弱関連型を R_i とする。 $L_{ij}/E_{ij} (j=1, \dots, n)$ のうち、1つの弱主体型を $L_{i'j'}/E_{i'j'}$ とし、 $E_{i'j'}$ を含まないある組を $L_{i'j'}/E_{i'j'}$ ($j'=1, \dots, n', n' \leq n-1$) とする。各 $L_{i'j'}/E_{i'j'}$ の主体、及び、その主体との間の R_i の関連を用いて、 $L_{i'j'}/E_{i'j'}$ の主体を識別する場合、 $L_{i'j'}/E_{i'j'}$ は R_i に基づいて、 $L_{i'j'}/E_{i'j'}$ ($j'=1, \dots, n'$) に識別従属するといひ、以下のように書く。

$L_{i'j'}/E_{i'j'}$ は R_i に基づいて、 $L_{i'j'}/E_{i'j'}$ ($j'=1, \dots, n'$) に識別従属するといひ、以下のように書く。
ここで、IDは識別従属性を表す。例えば、(顧客/氏名)を主キーに持つ弱主体型“請求”は、請求先に基づいて、顧客へ識別従属する。 $L_{i'j'}/E_{i'j'}$ の主体の主キーを $L_{i'j'}/E_{i'j'}/A_{i'jk_j}$ ($j'=1, \dots, n', n' \leq n-1$)、 $L_{i'j'}/E_{i'j'}/A_{i'jk_j}$ とし、主キー値が $v_{i'jk_j}$ ($j'=1, \dots, n', n' \leq n-1$)、 $v_{i'jk_j}$ の $e_{i'j'}$ は、 $L_{i'j'}/E_{i'j'}$ の部分集合 $\{e_{i'j'}/A_{i'jk_j} | (e_{i'j'}/A_{i'jk_j}) \in R_i \text{ かつ } e_{i'j'}/A_{i'jk_j} = v_{i'jk_j} (j'=1, \dots, n')\}$ かつ $A_{i'jk_j}(e_{i'j'}/A_{i'jk_j}) = v_{i'jk_j} (j'=1, \dots, n')$ の中で、 $A_{i'jk_j}(e_{i'j'}/A_{i'jk_j}) = v_{i'jk_j}$ となる主体として識別できる。ここで、 R_i から $L_{i'j'}/E_{i'j'}$ ($j'=1, \dots, n'$)、 $L_{i'j'}/E_{i'j'}$ のみを残す射影(projection)演算で R_i が得られるものとする。以後、弱主体型 $E_{i'j'}$ の主体の属性ではなくても、 $E_{i'j'}$ の主体の主キーに含まれる属性は、 $L_{i'j'}/E_{i'j'}/(A_{i'j'1}, \dots, A_{i'j'm})$ に含めて書くことができ、 $A_{i'j'1}$ の代りに $(E_{i'j'}/A_{i'j'k_j})$ と書く。

(2) 属性値従属性

相異なる $L_{ij}/E_{ij} (j=1, \dots, n, n \geq 2)$ 間の関連型を R_i とする。 $L_{ij}/E_{ij} (j=1, \dots, n)$ のうち、ある1つを $L_{i'j'}/E_{i'j'}$ とし、ある組を $L_{i'j'}/E_{i'j'}$ ($j'=1, \dots, n', n' \leq n$) とする。 R_i から $L_{i'j'}/E_{i'j'}$ 及び各 $L_{i'j'}/E_{i'j'}$ のみを残す射影演算で、 R_i が得られるものとする。 $L_{i'j'}/E_{i'j'}$ の任意の主体を $e_{i'j'}$ とし、 $e_{i'j'}$ を含む R_i の任意の関連を $(e_{i'j'}/A_{i'jk_j}, e_{i'j'}/A_{i'jk_j}, \dots, e_{i'j'}/A_{i'jk_j}) (e_{i'j'}/A_{i'jk_j} \in L_{i'j'}/E_{i'j'}, j'=1, \dots, n')$ とする。 $e_{i'j'}$ の主キーは除くある属性を $L_{i'j'}/E_{i'j'}/A_{i'jk_j}$ とし、各 $e_{i'j'}/A_{i'jk_j}$ の属性のある組を $L_{i'j'}/E_{i'j'}/(A_{i'j'1}, \dots, A_{i'j'm})$ とする。各 $L_{i'j'}/E_{i'j'}$ に $L_{i'j'}/E_{i'j'}$ と同じものがあ

ても良いが、 A_{1vk_v} と $A_{1j'1}, \dots, A_{1j'm_j}$ は異なる。要素 $(e_{111_1}, \dots, e_{1n'1_n})$ からなる集合を領域とし、各 $e_{1j'1_j} / L'_{1j'} / E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$ の値のみを演算に用いるある関数 F_v に対して、 $A_{1vk_v}(e_{1vk_v}) = F_v((e_{111_1}, \dots, e_{1n'1_n}) | (e_{1vk_v}, e_{111_1}, \dots, e_{1n'1_n}) \in R_v)$ となる場合、 $L'_{1v}/E'_{1v}/A_{1vk_v}$ は R_v 及び F_v に基づいて、 $L'_{1j'1_j} / E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$ ($j'=1, \dots, n'$)に属性値従属するといひ、以下のように書く。
 $L'_{1v}/E'_{1v}/A_{1vk_v} \stackrel{VD}{=} L'_{1j'1_j} / E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$
 $\dots, L'_{1n'1_n} / E'_{1n'} / (A_{1n'1_1}, \dots, A_{1n'm_n})$
 ここで、VDは属性値従属性を表す。例えば、売上の属性“売上額”は、売物、及び、売上数と単価の乗算に基づいて、同じ主体の属性“売上数”、及び、製品の属性“単価”に属性値従属する。同じ主体の属性のみの間の属性値従属性もある。この場合、各 $L'_{1j'1_j}/E'_{1j'}$ は書かず、 R_v の代りに E_j と書く。

(3) 関連の存在従属性
 相異なる $L'_{1j'1_j}/E'_{1j'}$ ($j'=1, \dots, n, n \geq 2$)間の関連型を R_i とする。各 $L'_{1j'1_j}/E'_{1j'}$ の主体 $e_{1j'1_j}$ に関して、少なくとも主キーを含む属性のある組を $L'_{1j'1_j}/E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$ ($m_j \geq 1$)とし、その値を $v_{1j'1_j}, \dots, v_{1j'm_j}$ とする。各 $e_{1j'1_j}$ の属性の別のある組を $L'_{1j'1_j}/E'_{1j'} / (A'_{1j'1_1}, \dots, A'_{1j'm'_j})$ ($m'_j \geq 0$)とし、その値を $v'_{1j'1_j}, \dots, v'_{1j'm'_j}$ とする。関連に含まれる主体相互の対応条件を示すある述語 $P_i(v_{111_1}, \dots, v_{11m_1}, \dots, v_{1n'1_n}, \dots, v_{1n'm_n})$ 、及び、関連の存在条件を示すある述語 $P_e(v'_{111_1}, \dots, v'_{11m'_1}, \dots, v'_{1n'1_n}, \dots, v'_{1n'm'_n})$ が共に真ならば、 $(e_{111_1}, \dots, e_{1n'1_n}) \in R_i$ となる場合、 R_i は P_i 及び P_e に基づいて、 $L'_{1j'1_j}/E'_{1j'}$ ($j'=1, \dots, n$)に存在従属するといひ、以下のように書く。
 $R_i \stackrel{RD}{=} P_i : L'_{111_1}/E'_{11'} / (A_{111_1}, \dots, A_{11m_1}), \dots, L'_{1n'1_n}/E'_{1n'} / (A_{1n'1_1}, \dots, A_{1n'm_n})$
 $\text{if } P_e : L'_{111_1}/E'_{11'} / (A'_{111_1}, \dots, A'_{11m'_1}), \dots, L'_{1n'1_n}/E'_{1n'} / (A'_{1n'1_1}, \dots, A'_{1n'm'_n})$
 ここで、RDは関連の存在従属性を表す。 $m'_j=0$ の場合、

P_e は $e_{1j'1_j}$ の属性値を反映しない。 P_e が恒真ならば、if以降を書かない。例えば、請求先は、顧客の主キー“氏名”と請求の主キー(顧客/氏名)の値が等しい対応条件に基づいて、顧客と請求に存在従属する。

(4) 主体の存在従属性

相異なる $L'_{1j'1_j}/E'_{1j'}$ ($j'=1, \dots, n, n \geq 2$)間の関連型を R_i とする。 $L'_{1j'1_j}/E'_{1j'}$ ($j'=1, \dots, n$)を2つに分けた組を $L'_{1j'1_j} / E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$ ($j'=1, \dots, n, n' \geq 1, n''=n-n'$)とする。両者に同じものは含まれない。各 $L'_{1j'1_j} / E'_{1j'}$ の主体 $e_{1j'1_j}$ の属性のある組を $L'_{1j'1_j} / E'_{1j'} / (A_{1j'1_1}, \dots, A_{1j'm_j})$ ($m_j \geq 0$)とし、その値を $v_{1j'1_j}, \dots, v_{1j'm_j}$ とする。
 $e_{1j'1_j} \in L'_{1j'1_j} / E'_{1j'}$ ($j'=1, \dots, n'$)、 $e_{1j''1_{n''}} \in L'_{1j''1_{n''}} / E'_{1j''}$ ($j''=1, \dots, n''$)を含む関連 r_{ih} が R_i の存在従属性の条件を満たし、かつ、主体の存在条件を示すある述語 $P_e(v_{111_1}, \dots, v_{11m_1}, \dots, v_{1n'1_n}, \dots, v_{1n'm_n})$ が真ならば、 $e_{1j'1_j} \in L'_{1j'1_j} / E'_{1j'}$ ($j'=1, \dots, n'$)、かつ、 $r_{ih} \in R_i$ となる場合、 $L'_{1j'1_j} / E'_{1j'}$ ($j'=1, \dots, n'$)は R_i 及び P_e に基づいて、 $L'_{1j'1_j} / E'_{1j'}$ ($j'=1, \dots, n'$)に存在従属するといひ、以下のように書く。
 $L'_{111_1} / E'_{11'} / \dots, L'_{1n'1_n} / E'_{1n'} \stackrel{ED}{=} L'_{1j'1_j} / E'_{1j'}$

if $P_e : L'_{111_1} / E'_{11'} / (A_{111_1}, \dots, A_{11m_1}), \dots, L'_{1n'1_n} / E'_{1n'} / (A_{1n'1_1}, \dots, A_{1n'm_n})$
 ここで、EDは主体の存在従属性を表す。例えば、請求は請求先、及び、顧客の属性“合計”が正值である存在条件に基づき、顧客に存在従属する。 P_e が恒真ならば、if以降を書かない。 $m'_j=0$ の場合、 P_e は $e_{1j'1_j}$ の属性値を反映しない。 $L'_{1j'1_j} / E'_{1j'}$ の中に弱主体型がある場合、その型の主体の識別に必要な関連も、主体と共に存在するものとする。

E_j の主体に関して、少なくとも主キーを含む属性のある組を A_{1jk} ($k=1, \dots, m$)とする。主体の存在条件を表すある述語 P_e があり、 $\{e_{1j'1_j} | A_{1j'1_j}(e_{1j'1_j}) = v_{j'1_j}\}$

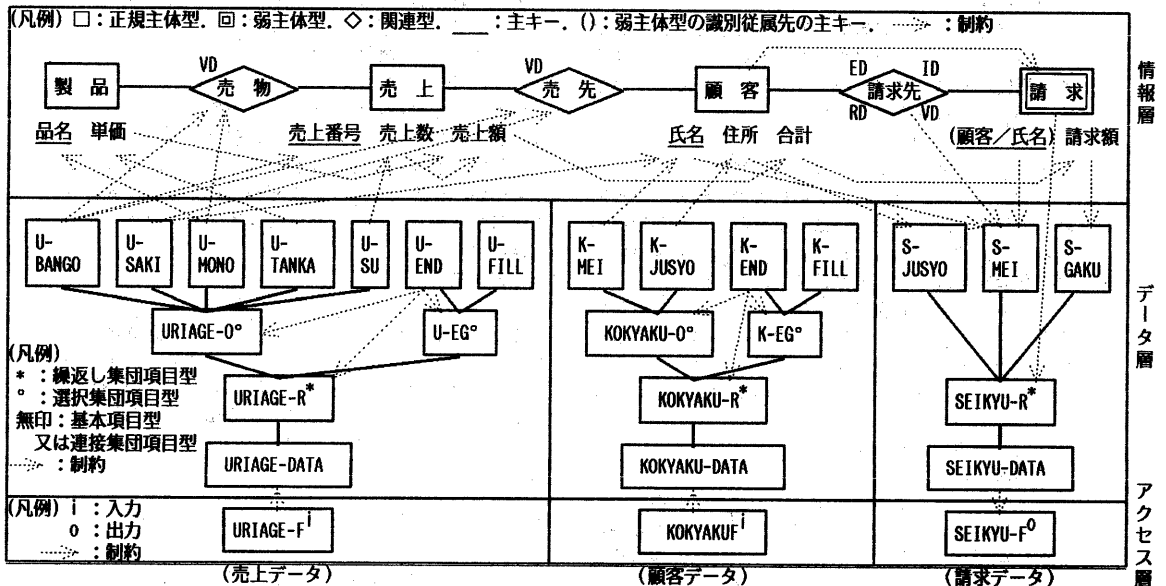


図2 請求書作成プログラムのPSD図

..., $A_{jm}(e_{ij}) = v_{jm}$ かつ $P_0(v_{j1}, \dots, v_{jm})$ が真) (E_j となる場合, P_0 に基づく主体が E_j に存在するといひ、以下のように書く。

$$E_j \leftarrow P_0 : A_{j1}, \dots, A_{jm}$$

ここで、ECは主体の存在制約を表す。

図2の情報層に示すように、ER図に制約を付記する。プログラムの入力データから出力データを得る演算は、制約で記述できる。制約は広くても、高々1つの関連に含まれる主体群の範囲の条件しか表せない。その範囲外の属性を制約で参照したい場合は、以下のいずれかの方法を用いる。①範囲内の主体と、参照される属性を持つ主体との間に関連および制約を設け、参照される属性値へ適切な演算を作用させて得られた値を、範囲内の主体の属性値とする。②参照される属性を持つ主体も関連に含める。

2.3 データ層

2.3.1 構造

以下の型を用いて、データ項目の並び方、及び、基本項目内の記号の並び方を定める。型を D_q と書く。

(1)基本項目型

それ以上、分解できないデータは本型に属する。基本項目内の記号の並び方も本型で定める。基本項目内の記号の並び方は例えば、COBOLと同様に定める。基本項目は固定長である。

(2)連結集団項目型

順序を持って1つずつ存在する2つ以上の型のデータの並びは本型に属する。相異なる $D_{qr}(r=1, \dots, n)$ のデータの並びを持つ D_q を、以下のように書く。

$$D_q ::= D_{q1} \dots D_{qn}$$

(3)繰返し集団項目型

1つ以上、繰返して存在する同じ型のデータの並びは本型に属する。 D_q のデータの並びを持つ D_q を以下のように書く。本型には正整数をとる指標 I_{qq} が附随し、指標付き型記号 $D_{qq}(I_{qq})$ は I_{qq} 番目のデータを指す。

$$D_{qq} ::= D_q^{*I_{qq}}$$

(4)選択集団項目型

2つ以上、存在し得る型のデータのうち、いずれか1つのデータが、ただ1つ存在する並びは本型に属する。相異なる $D_{qr}(r=1, \dots, n)$ のいずれかのデータを持つ D_q を、以下のように書く。

$$D_q ::= D_{q1} \mid \dots \mid D_{qn}$$

図2のデータ層に示すように、木状の構造図(structure diagram) ⁸⁾ で構造を表現できる。基本項目型に属する個々のデータは、以下のいずれかで識別できる。

① $D_1, D_2(I_2), D_3, D_4(I_4)$ のように、根から順次修飾された指標付き型記号。② $D_4(I_2, I_4)$ のように、識別に必要な全指標が付いた一意の基本項目型記号。同じ木構造内の他の繰返し集団項目型の指標からなる算術式を、指標として書くことができる。①②を基本項目型のデータの識別項という。

2.3.2 制約

以下の種類の制約を用いる。

(1)情報制約

以下のように、基本項目型および指標と、情報層の型とを対応させる。以後、基本項目型データの識別項および指標を O_s と書く。① E_j の1つの主体の存在を表す $O_{js}(s=1, \dots, m, m \geq 1)$ の各々を、 E_j の主体の主キーの各属性に対応させる。② R_i の1つの関連、及

び、その関連に含まれる $L_{ij}/E_{ij}(j=1, \dots, n, n \geq 2)$ の1つずつの主体の存在を表す $O_{ij1}, \dots, O_{ijn_j}(j=1, \dots, n, n_j \geq 1)$ の各々を、各 L_{ij}/E_{ij} の主キーの各属性に対応させる。③ $O_{js}(s=1, \dots, m, m \geq 1)$ の各々が①又は②により、 E_j の主体の主キーの各属性に対応しているものとする。 $O_s(s=1, \dots, m)$ で表される主キー値を持つ E_j の主体の属性値を表す O_{js} は、属性 E_j/A_{jk} に対応させる。以上を情報制約といひ、以下のように書く。

$$\textcircled{1} O_{j1}, \dots, O_{jn} \xrightarrow{\text{ERI}} E_j$$

$$\textcircled{2} O_{ij1}, \dots, O_{ijn_1}, \dots, O_{in1}, \dots, O_{inn_n} \xrightarrow{\text{ERI}} R_i$$

$$\textcircled{3} O_{js} \cdot [O_{j1}, \dots, O_{jm}] \xrightarrow{\text{VI}} E_j/A_{jk}$$

ここで、EIは主体の存在、ERIは主体および関連の存在、VIは属性値の情報制約を表す。情報層の存在制約ECで与えられる主体に対応する②の $O_{ij1}, \dots, O_{ijn_j}$ 、③の $O_{j1}, \dots, O_{jm}$ は定数で書く。各々の基本項目型および指標は、高々1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が、順序を表す関連が指定されていない主体間には順序関係はない。このため、出力データに対する①では、順序は考慮せずに、 E_j の各主体をデータに対応させる。②では一部の ij に関して、他の情報制約、又、他の指標値に対する同じ制約の適用により、 $O_{ij1}, \dots, O_{ijn_j}$ の各データが決まっている場合、残りの ij に関して②を満たす各 L_{ij}/E_{ij} の主体を $O_{ij1}, \dots, O_{ijn_j}$ に対応させる。②を満たす L_{ij}/E_{ij} の主体が複数存在する場合は①と同様に、基本項目内の記号列と、その基本項目型に対応する属性の定義域内の値とは、相互に対応しているものとする。

(2)繰返し制約

以下の値を $D_q ::= D_q^{*I_{qq}}$ の繰返しデータの個数とする。① $O_{js}(s=1, \dots, m, m \geq 1)$ で定まる主キー値を持つ E_j の主体の A_{jk} の値。② E_j の主体数。又、③ O_{js} で定まるデータ又は指標に対して、ある述語 P_{qr} が真ならば、そのデータ又は指標値の直前で繰返しが終了する。以上を繰返し制約といひ、以下のように書く。

$$\textcircled{1} D_q ::= D_q^{*I_{qq}} E_j/A_{jk}[O_{j1}, \dots, O_{jm}]$$

$$\textcircled{2} D_q ::= D_q^{*I_{qq}} E_j$$

$$\textcircled{3} D_q ::= D_q^{*I_{qq}} O_{js}$$

ここで、ICは繰返し制約を表す。①の $O_{js}(s=1, \dots, m)$ は定数でも良い。指標に算術式を書けるので、零等、本制約の規定範囲外の指標値も現れる。その場合、規定範囲内の指標値を持つ情報制約(1)②のデータのみに対応する主体が存在すること以外は、情報制約の適用はないものとする。

(3)選択制約

$D_q ::= D_{q1} \mid \dots \mid D_{qn}$ の各 $D_{qr}(r=1, \dots, n)$ へ対応する述語 $P_{qr}(r=1, \dots, n)$ を以下のデータ又は値へ作用させ、ただ1つ真となる P_{qr} へ対応する D_{qr} を選択する。① O_{js} で定まるデータ。② $O_{js}(s=1, \dots, m)$ で定まる主キー値を持つ E_j の主体の属性 A_{jk} の値。③ E_j の主体数。以上を選択制約といひ、以下のように書く。

$$\textcircled{1} D_q ::= D_{q1} : P_{q1} \mid \dots \mid D_{qn} : P_{qn} O_{js}$$

$$\textcircled{2} D_q ::= D_{q1} : P_{q1} \mid \dots \mid D_{qn} : P_{qn} E_j/A_{jk}[O_{j1}, \dots, O_{jm}]$$

$$\textcircled{3} D_q ::= D_{q1} : P_{q1} \mid \dots \mid D_{qn} : P_{qn} E_j$$

ここで、SCは選択制約を表す。②の $O_{js}(s=1, \dots, m)$

は定数でも良い。

図2のデータ層に示すように構造へ制約を付記する。

2.4 アクセス層

構造にはデータ集合型があり、データ集合名、データ入力/出力の区別および授受単位長を定める。1つのデータ集合型の中のデータ群は、アクセス順序で結合された1つの記号列と見なし、データ層の1つの木構造に対応させる。この対応をデータ制約という。データ集合型を S_t 、入力/出力をinput/output、授受単位長を l_t 、データ層における木構造の根となるデータ型を D_t とすると、構造および制約を以下のように書く。

$S_t, \text{input}, l_t, D_t$ 又は
 $S_t, \text{output}, l_t, D_t$

図2のアクセス層に構造および制約を図示する。図2全体をPSD図という。

3. 適用例

3.1 PSDL

PSDMを適用して、主に以下の制限を設けたPSDL、及び、COBOLプログラム・ジェネレータのプロトタイプPSD-CGを試作中である。①主体の主キーは1つの属性からなる。②属性値の定義域は数値または文字列からなる。③データの木構造の根となるデータ型、及び、レコードに対応するデータ型の書き方を特別に設ける。④データ型はCOBOL同様に、木構造の根から深さ優先で記述する。⑤入力/出力別に情報制約の書き方を分ける。⑥情報制約における入出力データの参照可能範囲は、それまでに入出力したレコード、及び、参照を行なう制約で出力データを設定中の出力レコードのみである。⑦関数および述語の定義手段は設けない。付図に、PSDLで記述された請求書作成プログラムを示す。PSD-CGで生成されるCOBOLプログラムの効率化は考慮しなかった。

3.2 従来言語との比較

COBOL⁷⁾、MODEL²⁾及びQBE³⁾と、PSDLとの簡単な比較を述べる。COBOLは手続き言語であるが、PSDLは非手続き言語である。データに着目した仕様を用いた非手続き言語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ており、情報構造仕様とデータ表現仕様が分化していない。非手続きデータベース言語QBEで用いられている関係モデル⁹⁾では、関係は任意の定義域の集まりである。一方、EARモデルでは、対象世界に存在する事物、及び、事物間の関係を主体および関連に対応させること等により、対象世界の自然な見方ができる⁴⁾。

PSDLでは、情報層からデータ層へ階層的に、仕様の理解、拡張および記述ができる。COBOLやMODELでは、情報構造仕様とデータ表現仕様が分化していないことから、プログラムをまとめて理解、拡張および記述する傾向が大きい。更に、PSDLでは対象世界の自然な見方ができるEARモデルを用いているので、PSDLの理解性、拡張性および記述性は比較的良好なものと考えられる。MODELでは、繰返しデータ相互に対応付ける指標の取り扱いが困難とされている²⁾。APDLでは、繰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。PSD-CGは、PSDL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。COBOLで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。一方、APDLでは、付図のように約70sとなったので、APDLの最小性が特に悪いことはない。

適用性に関しては、以下のように考えられる。主体型と関連型をQBEの表に対応させた場合、情報層の制約を用いて、QBEの検索、更新、挿入および削除に相当する機能を記述できる。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、データ型で記述できる。ポインタを用いたデータのような特殊なデータに関しては、そのデータを解析するサブ・プログラムを介してアクセスする。そのサブ・プログラムはPSDLで記述できる。入力データのエラー処理に関しては、各層にエラー処理の仕様を記述する。

4. おわりに

プログラム仕様記述法PSDMおよび適用例PSDLを報告した。データ本来の性質に基づいて、3階層に仕様を分けて記述すること、及び、対象世界の自然な見方ができるEARモデルを用いたことにより、PSDMで重視した理解性、拡張性および記述性は比較的良好なものと考えられる。PSD-CGは、PSDLがプログラムを自動生成できる形式性を備えていることを示している。PSDMの最小性はCOBOLと比較して、特に悪いことはない。情報層およびデータ層の適用性は各々、QBEおよびCOBOLと少なくとも同等であると考えられる。

今回はPSDMの基本的な内容を報告しており、今後、以下の項目を研究の予定である。①可変長レコード・アクセス、乱アクセス、データベース・アクセス、属性を持つ関連、及び、入力データ・チェック専用の制約等も入れたPSDMの拡張。②関数および述語の定義方法。③効率の良いプログラムの生成法。④仕様検証法。⑤データベース設計へ情報構造仕様を流用することによるデータベースとプログラムの仕様設計の融合。⑥画面上のPSD図を用いた仕様設計法。⑦PSDMに基づくソフトウェア再利用方法。⑧評価。

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, Academic Press, pp. 57-125(1977).
- 3) Zloof, H. M.: Query by Example: a Data Base Language, IBM Syst. J., No. 4, pp. 324-343(1977).
- 4) Chen, P. P.: The Entity-Relationship Model-Toward a Unified View of Data, ACM TODS, Vol. 1, No. 1, pp. 9-36(1976).
- 5) van Griethuysen, J. J. et al. (ed): Appendix D. The Entity-Attribute-Relationship Approaches, Concepts and Terminology for the Conceptual Schema, ISO TC97/SC5/WG3, March 15(1982).
- 6) 穂鷹良介: 講座 データベースの論理設計(1), 情報処理, Vol. 24, No. 5, pp. 651-665, May(1983).
- 7) 日本工業規格(JIS) 電子計算機プログラム言語 COBOL, JIS C 6205-1980, p. 383, 日本規格協会(1980).
- 8) Jackson, M. A.: Principles of Program Design,

Academic Press(1975).
9)Codd,E.F.:A Relational Model of Data for Large
Shared Data Banks,Comm. ACM, Vol. 13, No. 6, PP. 377-
387(1970).

10) 橋本正明, 稲田満, 佐藤匠正: プログラム仕様記
述に必要な概念構成を与えるプログラム・アーキテ
クチャ, 情報処理学会第25回全国大会論文集(1), pp
493-494(1982).

```
program SEIKYUSYO;
information-layer;
entity-type SEIHIN;
  description HINMEI{HINMEI,str,id},TANKA{YEN,num,}; *1
entity-type URIAGE;
  description URIAGEBANGO{BAN,num,id},URIAGESU{KO,num,},URIAGEGAKU{YEN,num,};
entity-type KOKYAKU;
  description SHIMEI{SHIMEI,str,id},JUSYO{JUSYO,str,},GOKEI{YEN,num,};
entity-type(weak) SEIKYU; *2
  description SEIKYUGAKU{YEN,num,};
relship-type URIMONO;
  collection /URIAGE,/SEIHIN; *3
  value-depn /URIAGE/URIAGEGAKU <- FMULT:/URIAGE/URIAGESU,/SEIHIN/TANKA; *4
relship-type URISAKI;
  collection /URIAGE,/KOKYAKU;
  value-depn /KOKYAKU/GOKEI <- FSUM:/URIAGE/URIAGEGAKU; *5
relship-type(weak) SEIKYUSAKI; *6
  collection /SEIKYU,/KOKYAKU;
  id-depn /SEIKYU <- /KOKYAKU; *7
  rel-depn(int) PSEIKYUSAKI:/KOKYAKU/SHIMEI,/SEIKYU/(KOKYAKU/SHIMEI); *8
  ent-depn /SEIKYU <- /KOKYAKU if PSEIKYU if SEIKYU:/KOKYAKU/GOKEI; *9
  value-depn /SEIKYU/SEIKYUGAKU <- FSAME:/KOKYAKU/GOKEI;
data-layer;
iteration-type(root) URIAGE-DATA{U}::=URIAGE-R; *10
iterate-cons(data) U-END(U){= "END"}; *11
selection-type(rec) URIAGE-R::=U-EG{= "END"},URIAGE-O{NOT = "END"};
select-cons(data) U-END(U); *12
sequence-type U-EG::=U-END,U-FILL;
element-type U-END PIC X(3);
element-type U-FILL PIC X(31);
sequence-type URIAGE-O::=U-BANGO,U-SAKI,U-MONO,U-TANKA,U-SU;
element-type U-BANGO PIC 9(4);
element-type U-SAKI PIC X(10);
element-type U-MONO PIC X(10);
element-type U-TANKA PIC 9(6);
element-type U-SU PIC 9(4);
rel-ent-cons(info) URISAKI//URIAGE <- U-BANGO(U),/KOKYAKU <- U-SAKI(U); *13
rel-ent-cons(info) URIMONO//URIAGE <- U-BANGO(U),/SEIHIN <- U-MONO(U);
value-cons(info) SEIHIN/TANKA[U-MONO(U)] <- U-TANKA(U); *14
value-cons(info) URIAGE/URIAGESU[U-BANGO(U)] <- U-SU(U);
iteration-type(root) KOKYAKU-DATA{Y}::=KOKYAKU-R;
iterate-cons(data) K-END(Y){= "END"};
selection-type(rec) KOKYAKU-R::=K-EG{= "END"},KOKYAKU-O{NOT = "END"};
select-cons(data) K-END(Y);
sequence-type K-EG::=K-END,K-FILL;
element-type K-END PIC X(3);
element-type K-FILL PIC X(23);
sequence-type KOKYAKU-O::=K-MEI,K-JUSYO;
element-type K-MEI PIC X(10);
element-type K-JUSYO PIC X(16);
ent-cons(info) KOKYAKU <- K-MEI(Y); *15
value-cons(info) KOKYAKU/JUSYO[K-MEI(Y)] <- K-JUSYO(Y);
iteration-type(root) SEIKYU-DATA{S}::=SEIKYU-R;
iterate-cons(info)(car-num) S{SEIKYU}; *16
sequence-type(rec) SEIKYU-R::=S-JUSYO,S-MEI,S-GAKU;
element-type S-JUSYO PIC X(16);
element-type S-MEI PIC X(10);
element-type S-GAKU PIC ¥9(9);
ent-cons(data) S-MEI(S) <- SEIKYU; *17
rel-ent-cons(data) S-MEI(S) <- SEIKYUSAKI//KOKYAKU where /SEIKYU <- S-MEI(S); *18
value-cons(data) S-GAKU(S) <- SEIKYU/SEIKYUGAKU[S-MEI(S)]; *19
value-cons(data) S-JUSYO(S) <- KOKYAKU/JUSYO[S-MEI(S)];
access-layer;
dataset-type(file) URIAGE-F,input,34,URIAGE-DATA; *20
dataset-type(file) KOKYAKUF,input,26,KOKYAKU-DATA;
dataset-type(file) SEIKYU-F,output,36,SEIKYU-DATA; *20)ファイルに対するデータ・アクセス方法.
end-program;
```

付図 PSDLで記述された請求書作成プログラム