
ショートノート

プログラム仕様記述法 PSDM を用いたプログラム量産方式の提案†

橋 本 正 明†† 伊 吹 公 夫†††

ソフトウェア製造効率化のためにプログラム仕様記述法 PSDM を用いたプログラム量産方式を提案する。情報化社会の進展に伴い、ソフトウェア分野にも量産化が必要となる。本方式では、事務処理分野を業種・事業規模などで分けたような問題領域を対象に問題領域仕様の原型を PSDM に基づいて記述し、この原型から多数のプログラムを合成する。本小論では本方式の考え方を述べる。

1. ま え が き

情報化社会の進展に伴い、ソフトウェア需要はますます増え、ソフトウェア分野にも量産化が必要となる。このため、事務処理分野を業種・事業規模などで分けたような問題領域を対象に問題領域仕様の原型を PSDM^{1),2)} に基づいて記述し、この原型から多数のプログラムを合成することを特徴とする量産方式³⁾を研究中である。問題領域仕様は領域内の業務方法を示すものである。

プログラム部品化ではプログラム単品ごとに仕様や構造を設計の上、部品を組み合わせてプログラムを作成する。一方、本量産方式のプログラム合成では製造対象システムの特殊性を問題領域仕様原型へ反映の上、プログラム単品ごとに入出力ファイルを指定する。そして、問題領域仕様からプログラム仕様を自動抽出し、この仕様から COBOL などのプログラムを自動生成する。このように、本方式にはプログラム単品ごとに仕様や構造を設計することが不要になる利点がある。

パラメータ化プログラミングでは異なる機能を持つプログラムごとに原型を記述し、パラメータへの値の設定により原型から個々のプログラムを合成する。しかし、事務処理分野では全システムの特殊性をすべてパラメータ化するのは非現実的なので、プログラム合成時に原型の変更が必要になる。本量産方式でも同じように原型の変更が必要になるが、PSDM の適用に

より変更が比較的容易になる利点がある。その理由を以下に述べる。① PSDM は理解性と拡張性を比較的備えている¹⁾。② 事務データ処理では、処理内容は比較的定形化しているが、データの入出力方法が多様である⁴⁾。したがって、入出力データ形式に関する記述と、概念に関する記述を分離した PSDM においては、頻度が多くなる入出力データ形式変更の波及範囲が小さい。③ PSDM では問題領域仕様にいくつものプログラム仕様を融合して記述できるので、異なる機能を持ついくつかのプログラムに現れている同一概念の変更は一回で済む。

本小論の目的は本方式を提案することである。なお、文献¹⁾と同様に入出力ファイルは順編成に限定して本方式の考え方を以下に示す。本論中の太字の説明は文献^{1),2)}を参照されたい。

2. 問題領域仕様の記述法

問題領域仕様は製品、売上、顧客、請求のような“概念”に関するもの、および、売上傳票や請求伝票のような“ファイル”に関するものに分ける。

概念は製品や顧客のような“物”、および、売上や請求のような“事”、つまり、事物を単位として捕える。また、売上と製品の間には売物という関係があるように事物相互の関係も捕える。事物は主体といい、事物相互の関係は関連という。そして、主体の種類は主体型で記述し、関連の種類は関連型で記述する。主体の性質は主体型ごとに記述した属性の値で表す。主体識別用の属性は主キーとして記述する。たとえば、主体“製品”の性質は属性“品名”や“単価”の値で表し、属性“品名”は主キーである。主体型がその属性からなる主キーを持つならば正規主体型として記述し、主キーの属性を他の主体型から借用していれば弱主体型として記述する。

† A Proposal of a Program Mass Production Method Using Program Specification Description Method PSDM by MASAAKI HASHIMOTO (ATR Communication Systems Research Laboratories) and KIMIO IBUKI (Department of Information Technology, Faculty of Engineering, Tokyo Engineering University).

†† ATR 通信システム研究所

††† 東京工科大学工学部情報工学科

主体“売上”の属性“売上額”の値は、同じ主体の属性“売上数”と、この主体に関連“売物”でつながる主体“製品”の属性“単価”の値との積で得るように、属性値の計算方法は属性値従属性制約で記述する。主体“請求”は、主体“顧客”の属性“合計”の値が正の場合に存在するように、主体の存在を決めるための計算方法は主体存在従属性制約で記述する。この場合、主体“請求”と“顧客”の間に関連“請求先”が存在するように、関連の存在を決めるための計算方法は関連存在従属性制約で記述する。弱主体型の主キーの属性を借用する先の主体型は識別従属性制約で記述する。以上は情報層に記述する。なお、上記の制約のいくつかが連立方程式を形成することにより、ある属性の値をこの値自体から計算して得るように¹⁾記述してはならない。

入出力ファイルのデータ形式は基本項目型、連結集団項目型、繰返し集団項目型、指標、繰返し制約、選

択集団項目型および選択制約で記述する。また、基本項目型や指標と、情報層内の型との対応は、データの情報制約で記述する。以上はデータ層に記述する。

データ形式以外のファイルの性質に関してはファイル名をデータ集合型名、レコード長をアクセス単位長、ファイル名とデータ形式記述との対応をアクセスのデータ制約で記述する。以上はアクセス層に記述する。

3. 問題領域仕様原型

一つの問題領域には多くのシステムがあり、この全システムの相違点も含めた問題領域仕様は現実には作成できない。このため、問題領域仕様の原型を作成する。原型は固定部と可変部に分ける。可変部には選択可能な仕様やパラメータを記述する。

原型の概念図を図1に示す。情報層に示す製品に関しては製品ごとに一律に単価を決めることもあれば、

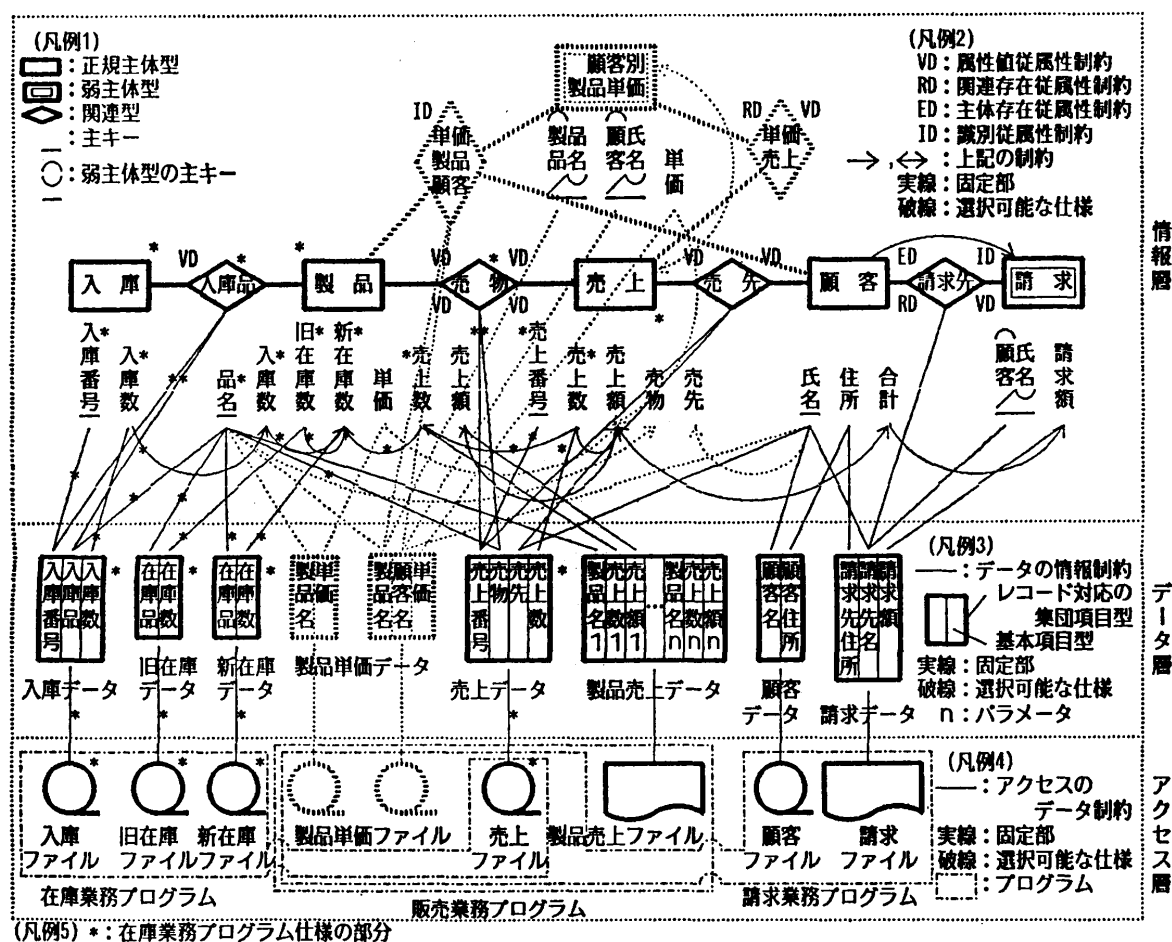


図1 問題領域仕様原型の概念図

Fig. 1 Outline of specification-prototype of a problem-region.

```

information-layer;
entity-type 入庫;
  description 入庫番号(番,num,id),入庫数(個,num,);
entity-type 製品;
  description 品名(品名,str,id),入庫数(個,num,);
    旧在庫数(個,num,),新在庫数(個,num,);
    単価(円,num,),売上数(個,num,),売上額(円,num,);
    value-depn(entity) 新在庫数 <- FADDSUB:旧在庫数,入庫数,売上数;
entity-type 売上;
  description 売上番号(番,num,id),売上数(個,num,),売上額(円,num,);
    売物(品名,str,),売先(氏名,str,);
entity-type(weak) 顧客別製品単価;
  description 単価(円,num,);
entity-type 顧客;
  description 氏名(氏名,str,id),住所(住所,str,),合計(円,num,);
entity-type(weak) 請求;
  description 請求額(円,num,);
relship-type 入庫品;
  collection /入庫,/製品;
    value-depn /製品/入庫数 <- FSUM:/入庫/入庫数;
relship-type 売物;
  collection /製品,/売上;
    value-depn /売上/売上額 <- FMULT:/売上/売上数,/製品/単価;
    value-depn /製品/売上額 <- FSUM:/売上/売上額;
    value-depn /製品/売上数 <- FSUM:/売上/売上数;
    value-depn /売上/売物 <- FSAMESTR:/製品/品名;
relship-type 売先;
  collection /売上,/顧客;
    value-depn /顧客/合計 <- FSUM:/売上/売上額;
    value-depn /売上/売先 <- FSAMESTR:/顧客/氏名;
relship-type(weak) 単価製品顧客;
  collection /顧客別製品単価,/製品,/顧客;
    id-depn /顧客別製品単価 <- /製品,/顧客;
relship-type 単価売上;
  collection /顧客別製品単価,/売上;
    rel-depn(int) PTRUE:/顧客別製品単価/(製品/品名);
      /顧客別製品単価/(顧客/氏名),/売上/売上番号
      if PTANKAURIAGE:/顧客別製品単価/(製品/品名),/売上/売物;
        /顧客別製品単価/(顧客/氏名),/売上/売先;
    value-depn /売上/売上額 <- FMULT:/売上/売上数,/顧客別製品単価/単価;
relship-type(weak) 請求先;
  collection /顧客,/請求;
    id-depn /請求 <- /顧客;
    rel-depn(int) PSEIKYUSAKI:/顧客/氏名,/請求/(顧客/氏名);
    ent-depn /請求 <- /顧客 if PSEIKYU:/顧客/合計;
    value-depn /請求/請求額 <- FSAME:/顧客/合計;

```

(説明) FADDSUB…第1パラメータ値と第2パラメータ値の和から、第3パラメータ値の差をとる関数。
 FSUM… $n(n \geq 0)$ 個のパラメータ値の総和をとる関数。 FMULT…二つのパラメータ値の積をとる関数。
 FSAMESTR…パラメータ値と同じ文字列をとる関数。 PTRUE…恒真の述語。 PTANKAURIAGE…第1パラメータ値
 と第2パラメータ値がおおの、第3パラメータ値と第4パラメータ値に等しければ真、その他は偽となる述語。
 PSEIKYUSAKI…二つのパラメータ値が等しければ真、その他は偽となる述語。 PSEIKYU…パラメータ値が正なら
 ば真、その他は偽となる述語。 FSAME…パラメータ値と同じ値をとる関数。
 破線の下線…問題領域仕様原型の可変部。 実線の下線…在庫業務プログラム仕様の部分。

図2 問題領域仕様原型の PSDL 記述 (情報層)

Fig. 2 PSDL-Description of specification-prototype of a problem-region (information-layer).

顧客対応に製品ごとの単価を決めることもある。前者では主体型「製品」の属性「単価」を用いる。後者では主体型「顧客別製品単価」の属性「単価」を用いる。これらは選択可能な仕様であり、破線で図示する。データ層に示す製品売上ファイルに関しては帳票1行に印刷する製品件数をパラメータとし、「 n 」で図示

する。PSDMを適用したプログラム仕様記述言語PSDLで図1の原型を記述したもののうち、紙面の都合により情報層のみを図2に示す。可変部には破線の下線を付けて図示する。

4. プログラム合成

問題領域仕様原型可変部の選択可能な仕様は選択し、パラメータには値を設定する。それから変更を加え、製造対象システムへ適合した問題領域仕様を得る。これらの作業は図1のような図形を扱うエディタなどを用いれば比較的容易になる。

次に、問題領域仕様のアクセス層のファイル群から製造対象のプログラム単品ごとに入出力ファイルを指定し、入出力制約を加える。そして、プログラム単品ごとに、後述の例に示すように入力ファイルから出力ファイルへ至るデータフローパス上の問題領域仕様の部分をすべて自動抽出する。これが文献¹⁾に示すプログラム仕様となる。それから PSD-CG などを用いて COBOL などのプログラムを自動生成する。図1のアクセス層に示すように、プログラムと入出力ファイルとの対応も蓄積しておけば、プログラム名の指定によりプログラム仕様を抽出できる。たとえば、在庫業務プログラムを指定すれば、その入出力ファイル間のデータフローパス上の問題領域仕様の部分、つまり、図1では“*”印付きの部分、図2では実線の下線付きの部分抽出し、在庫業務プログラム仕様とする。

5. ソフトウェア・ライフサイクルと量産組織

問題領域仕様原型は最初からその領域全体を見通して開発することは難しい。最初は典型的な情報システムを例にとり、分析を加えて原型を作成する。そして、他のシステムのプログラム合成へも適用しながら改善情報を得て、原型を改編していく。

量産組織は情報システム製造グループおよび問題領域仕様原型開発グループからなる。開発グループは原型の作成・改編とともに、原型の利用を高めるため製造グループを指導する。このため、開発グループには問題領域や量産方式に精通する要員が要るが少人数でよい。一方、多数の製造グループが行うプログラム合成作業は比較的容易になる。

6. 適用領域

問題領域仕様原型は利用回数が多いほど経済的なので、本方式は多数の類似プログラムが必要な領域に適する。最適領域として POS (Point Of Sales: 販売時点情報管理) システムの領域などがある。情報化社会の進展に伴い、最適領域はさらに増えよう。

7. むすび

本方式は、多数の類似プログラムが必要な領域において、問題領域仕様原型を開発するための高度な作業は少人数に限定し、多数のプログラム合成作業は比較的容易にするので、現実的な量産化を実現できよう。

なお、本研究は、NTT 情報通信処理研究所において行ったものである。

参考文献

- 1) 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, 情報処理学会論文誌, Vol. 27, No. 7, pp. 697-706 (1986).
- 2) 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).
- 3) 橋本正明, 伊吹公夫: プログラム仕様記述法 PSDM を用いたプログラム量産化, 情報処理学会ソフトウェア工学研究会資料, 46-1 (1986).
- 4) 魚田勝臣: 事務処理用簡易言語, 情報処理, Vol. 22, No. 6, pp. 545-549 (1981).

(昭和 61 年 4 月 28 日受付)

(昭和 63 年 5 月 10 日採録)



橋本 正明 (正会員)

昭和 21 年生。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勤務。昭和 63 年 ATR 通信システム研究所出向。これまで主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、プログラム自動作成の研究実用化に従事。著書「データ中心のプログラム仕様記述法」(井上書院)。工学博士。電子情報通信学会会員。



伊吹 公夫 (正会員)

昭和 7 年京都市に生まれる。昭和 30 年京都大学工学部電気工学科卒業。昭和 32 年同大学大学院修士課程修了。同年、NTT 入社。計算機、電子交換機、情報処理、情報処理理論、情報システムの研究実用化に従事。昭和 61 年東京工科大学工学部情報工学科教授、現在に至る。この間、米国イリノイ大学客員研究員、東京工業大学非常勤講師などを兼務。現在、情報システム、ソフトウェア学の体系化に興味を持つ。著書に「ソフトウェア学通論」などがある。工学博士。電子情報通信学会、ソフトウェア科学会各会員。