

高次部品化における概念辞書の構造

4 G - 4

岡本克己, 橋本正明, 竹中豊文
ATR通信システム研究所

1. はじめに

ソフトウェアの生産性向上を図るために、部品合成によるプログラムの自動作成が様々な形で研究されている。部品合成によるプログラム自動作成で用いられる部品は、①ソースコードを部品化したもの、②仕様を部品化したもの、③設計情報を部品化したもの、の3つに分類されている¹⁾。その中で筆者らは、仕様を部品化して再利用するための高次部品化²⁾について研究中である。

高次部品化のように仕様を部品として取り扱う場合、様々な性質を持った部品が多数存在し、またそれらの部品に対して複雑な操作が行われる。従って、どのような構造で部品を蓄積するかが、部品を再利用していく上で重要な課題となっている³⁾。

そこで本稿の第2章では高次部品化で取り扱う仕様を記述するための言語PSDL^{4), 5)}を説明し、さらに高次部品化の概要を述べる。第3章では、部品を蓄積するための構造として、部品の構造や部品相互の関係などを述べる。なお、様々な部品を再利用するには部品の理解性を高めなければならないので、本稿では複合部品の考え方を適用し、部品の理解性を高める工夫を行った。

2. PSDLと高次部品化

PSDLは、プログラムの入出力データの性質に着目した非手続き的なプログラム仕様記述言語であり、入出力データの性質を、①入出力データが表す対象世界に存在する事物や、その事物相互の関係に基づいて情報の枠組を定めた情報構造、②入出力データ中のデータ項目の並び方や、データ項目と情報構造の対応を定めたデータ形式、③ファイルなどの性質や、データ入出力の区別などを定めたデータ・アクセス方法、の3つの側面から捉え、各々を情報層、データ層、アクセス層に分け、さらに各層を構造と制約に分けて記述する。

また、情報層の記述にはERモデルと、エンティティやアトリビュート値、リレーションシップを導出するための制約を用いている。ERモデルや制約は人の認識方法と合致しているので、PSDLの情報層で対象世界を記述したものは、概念の記述であるといえる。

そこで高次部品化とは、PSDLによる概念の記述を部品化して再利用することである。この高次部品化で取り扱う部品を概念部品と呼ぶ。

リレーションシップ・タイプを含んだ概念部品には、そのリレーションシップ・タイプで関係付けられたエンティティ・タイプやそのアトリビュ

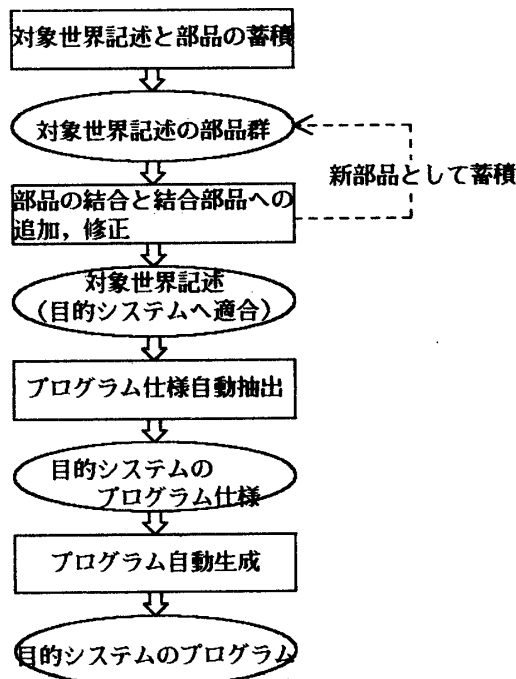


図1 高次部品化によるプログラム合成過程

ート、制約も含まれる。それらの概念部品は原則として同じ名前を持つエンティティ・タイプやリレーションシップ・タイプを重ね合わせることで、相互に結合することができ、その際に、不足部品の追加や部品の修正を行う。

結合された概念部品群にデータ層とアクセス層の仕様を与えることによって、完全なプログラム仕様ができ、別途研究中のプログラム自動生成法を用いることによっても目的システムのプログラム・コードを得ることができる。図1に高次部品化によるプログラム合成の過程を示す。

3. 概念辞書の構造

本章では、概念辞書の構造として部品の構造や部品相互の関係などを述べる。

3.1 部品の構造

部品の構造は原則として、PSDL記述要素の相互関係で決まる。図2の実線の矢印で示すように、エンティティ・タイプはアトリビュートを構成要素として持ち、そのエンティティ・タイプはリレーションシップ・タイプの構成要素となっている。さらに制約は、エンティティ・タイプ、アトリビュート、リレーションシップ・タイプを構成要素として持っている。従って、これらの相互関係を概念辞書の構造に反映した。

また、複数のエンティティ・タイプやリレーションシップ・タイプによって構成された部品を1

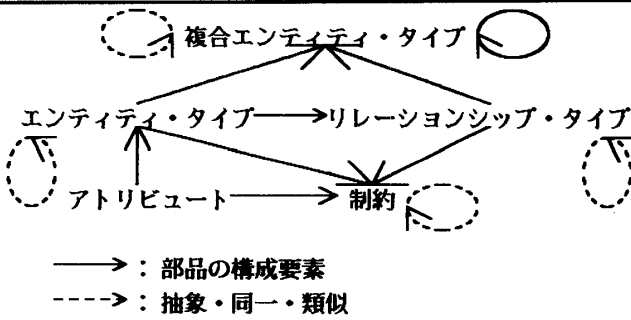


図2 概念辞書の構造

つのエンティティ・タイプとして認識した方が対象世界を理解し易い場合がある。このような部品を複合エンティティ・タイプと呼び、これも概念辞書の構造に反映した。この複合エンティティ・タイプの構成要素には、複合エンティティ・タイプ自身も使用可能である。複合エンティティ・タイプと他の部品とのつながりは、その部品と、複合エンティティ・タイプの構成要素とのつながりから決まる。

3.2 抽象・同一・類似

概念部品の中には、一部分だけが異なるような部品が幾つか存在する。これらの部品を抽象化して階層的に蓄積することによって、辞書の記述量を減らすことができる。この抽象化は、図3に示すようにエンティティ・タイプ、リレーションシップ・タイプ共に可能である。抽象化された部品と他の部品とのつながりは、その部品と、サブ・エンティティ・タイプやサブ・リレーションシップ・タイプのつながりから決まる。

また概念部品を再利用していく上で、ユーザが意図した部品そのものがない場合に、その代替部品を用いることができれば有効である。代替部品として、概念は同じであるが構造が異なっている同一概念の部品、概念が類似している類似概念部品が存在する。

抽象、同一、類似の関係も概念辞書の構造に含めるが、これらは部品の構造そのものを表すものではない。従って図2の点線の矢印に示すように、部品の構造を表すリンクとは異なる種類のリンクを張るものとする。

3.3 複合エンティティ・タイプの有用性

本項では、3.1項の複合エンティティ・タイプの有用性について述べる。複合エンティティ・タ

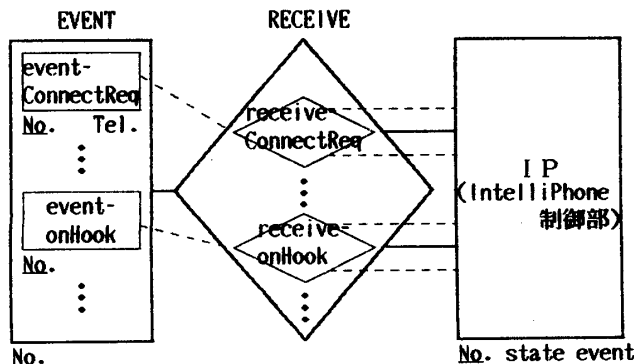


図3 概念の抽象化

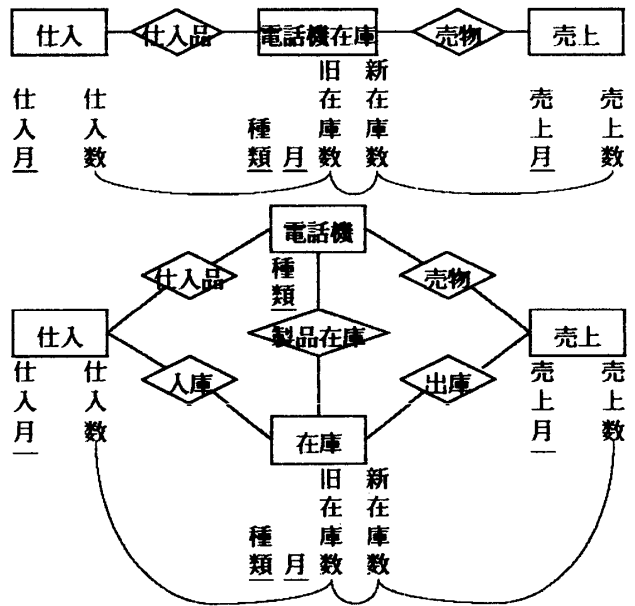


図4 概念の同一性

イプを用いると、対象世界を詳細情報のレベルで見るとはならず、意味を持つまとまった単位でマクロに対象世界を認識できるので理解性が向上する。また、前項で述べた同一概念や類似概念を扱う際、例えば図4に示すように異なる構造を持った概念は、複合エンティティ・タイプのレベルで同一性や類似性を比較すればよい。

4. おわりに

高次部品化の課題の中で特に重要な概念辞書の構造について報告した。今後は、概念部品の理解性を高めるためのツールとしてER図エディタを作成すると共に、概念の同一性や類似性、複合エンティティ・タイプについて研究を深めて、エディタに組み込む予定である。

参考文献

- 1) 古宮誠一, 原田実: 部品合成による自動プログラミング, 情報処理, Vol.28, No.10, pp.1329-1345 (1987).
- 2) 岡本克己, 橋本正明: 制約指向の概念モデルを用いた高次部品化によるプログラム合成, 情報処理学会ソフトウェア基礎論研究会資料, Vol.32, No.1, pp.1-10 (1989).
- 3) 橋本正明: E A R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P S D M, 情報処理学会論文誌, Vol.27, No.7 (1986).
- 4) Katsumi OKAMOTO, Masaaki HASHIMOTO: On Real-Time Software Specification Description with a Conceptual Data Model-Based Language, Proceedings of International Conference on Computing and Information 90 (1990).
- 5) G.D.Battista, H.Kangassalo, R.Tamassia: Definition libraries for conceptual modelling, Data & Knowledge Engineering, Vol.4, pp.245-260 (1989).
- 6) 橋本正明, 岡本克己: 非手続き型言語の集合と写像の性質に着目した入出力データの構造不一致検出・解決法, 情報処理学会プログラミング言語研究会資料, 掲載予定 (1990.6).