

2S-3

オブジェクト指向に基づく
部品原型と部品生成方式

永松 祐嗣* 前澤 裕行* 樽屋 克彦** 山本 利昭**
(*(株)日立製作所システム開発研究所 **(株)日立製作所戸塚工場)

1. はじめに

ソフトウェア生産性向上のための有効な手段として、既に正当性・信頼性が確認されたソフトウェアを部品として蓄積しておき、ソフトウェアの新規開発時、あるいは保守時に、再利用する部品再利用方式がある。

ここでは、信頼性、汎用性が高い、プログラム部品体系を実現するために、オブジェクト指向モデルに基づくプログラム部品原型と部品生成方式について提案する。また、本方式の交換ソフトウェアへの適用について述べる。

2. 部品生成方式概要

部品の再利用率を上げるには、汎用的な部品を用意し、利用する状況に応じて、部品を変更・修正(カスタマイズ)する必要がある。

既存の汎用部品の多くは、部品の変更・修正を人手で行わなければならない。この場合、どこをどのように変更すればよいか、の判断が難しくなる。もとのプログラムを詳しく知っていることが必要で、作った本人にすらどう変更すべきかがわからず、結局新しく作ってしまうことも多い。

一部の抽象データ型部品は、利用者が、変更部分・変更方法を意識しなくても、利用する状況に応じたパラメータ値を与えることにより、部品をカスタマイズすることが可能である[1]。しかし、抽象データ型部品は、データに対する汎用化ができず、変更方法をきめこまかく指定することが困難であり、部品の利用方法に制約が生じる。

そこで筆者等は、変更部分・変更方法に関するきめこまかな情報を付加した部品の雛形である「部品原型」を提案し、部品原型から、利用者の指定に従って、適当な変形を施して部品を生成する部品ジェネレータを開発した。

この方式を、オブジェクト指向によりモデリングすると図1ようになる。

即ち、オブジェクト指向モデルでは、全ての実体が、あるクラスのインスタンスである。クラスは、インスタンスの雛型と考えられる。そこで、部品原型が部品のクラスであり、部品が部品原型のインスタンスと見なした。

様々な利用状況に応じて、部品原型に生成パラメータを与え、その情報に従って部品原型に変更を施すことにより、部品原型から、種々の部品をインスタンスとして生成する。

部品原型の内部は、生成を制御するための記述と、プログラム部品に変形される雛型の記述とから成る。

以下、部品原型と部品ジェネレータの機能を挙げる。

(1)部分計算

生成パラメータ(部品の利用状況に関する情報)をもと

に、変数の初期値設定、型合わせ、メッセージの送り先の決定等を、あらかじめ部品生成時にやり、効率の良い部品に特殊化する。

(2)マクロ

マクロを導入し、部品原型中に現れるマクロは、生成式の指定に従って、マクロ値に置換する。

(3)部品生成制御

部品原型中にfor文、while文、if文を記述することにより、繰返しや条件判定の部品生成制御が可能(※は、プログラムコードに対するメタ記述を表わす)。

(4)継承

部品原型間を有機的に関連づけるため、オブジェクト指向モデルの特徴の一つである、継承関係を利用する。部品原型中にinclude文を記述することにより、継承した部分を自部品原型内にコピーすることができる。

上記の(1)~(4)の機能により、生成制御記述をきめこまかく設定し、人手による修正を要しないような枠組みを与えることができる。

3. 交換ソフトウェアにおける部品と部品生成方式

本方式を交換ソフトウェアに適用するため、CHILL言語用の部品ジェネレータを開発した。以下で述べる部品は、全てCHILL言語で記述されている。

交換ソフトウェアは、一般に以下のような特徴を持っている。

(1)ソフトウェアの寿命が長い

交換ソフトウェアは、一度開発されると、10年以上使用するものもめずらしくない。この間、仕様の変更による保守作業が度々行なわれ、保守コストは膨大なものになる。

(2)有限状態マシンモデル

交換ソフトウェアのプログラム構造は、機能ブロックにより階層化される。

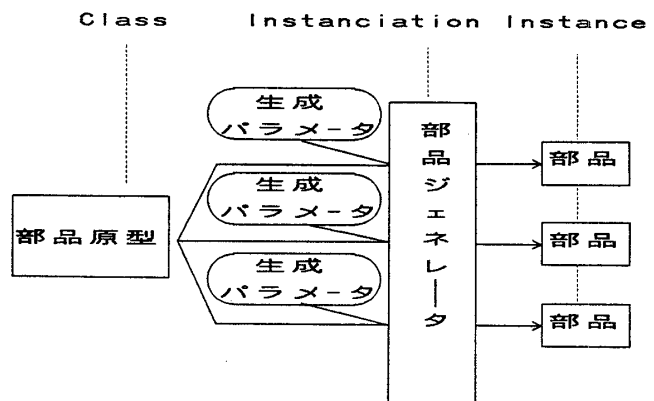


図 1 部品生成方式の原理

Program Parts Prototype and Program Parts Generator based on Object-Oriented Model

Yuji Nagamatsu, Hiroyuki Maezawa,
Katsuhiko Taruya, Toshiaki Yamamoto,
Hitachi, Ltd.

機能ブロックは、交換ソフトウェアを構成する機能要素単位であり、必要に応じて、より詳細な(サブ)機能ブロックとチャンネルに分割することができる。チャンネルは機能ブロック間の信号の通信路である。

各機能ブロックの動作は、有限状態マシンとしてモデリングできる。

(3)複雑な構造体データを使用

現在の交換ソフトウェアで用いるデータは、構造体、ポインタで結合したテーブル等、複雑な構造をしている。

上記(1)の特徴から、ソフトウェアの保守コストを低下させ、変更ソフトウェアの信頼性を高める必要がある。このため、最初からモジュール性の高いプログラムを構築しておかなければならない。即ち、モジュール性の高いプログラムを部品として、用意しておき、部品を使ってプログラムを構築しておく。そして、修正が発生した時は、修正部分の部品を交換することで、より簡単に、より信頼性の高い、保守変更作業を遂行することが可能となる。

上記(2)(3)の特徴から、交換ソフトウェアの部品(原型)の体系として、パッケージ部品(原型)とLSI部品(原型)という2種類に分類し設定した。

パッケージ部品(原型)は、交換プログラムの機能ブロックに対応し、機能ブロックの動作制御又は、状態遷移を実現した部品(原型)である。パッケージ部品(原型)は、有限状態マシンに相当するオブジェクトである。

LSI部品(原型)は、各機能ブロック中の、状態遷移に伴うタスクを実現する部品(原型)で、交換プログラム

の各種局データ(番号表、ロギング表、など)、と手続き(補足、参照、更新等)を一体化したオブジェクトである。

パッケージ部品(原型)は、プログラムのコントロール(制御構造)部分、LSI部品(原型)は、プログラムのデータ部分を記述している。

LSI部品(原型)は、ハードウェアのLSIと同様に、パッケージ部品(原型)に組み込んで使用する。組み込むLSI部品を変えることにより、異なるデータで同じような制御構造を記述したプログラムを実現できる。

これにより、交換プログラムの局データ部分と、制御部分が独立し、モジュール性の高い部品を実現できる。そして、局データの変更に対しては、変更の対象となる局データ部分を実現しているLSI部品を交換し、制御構造の変更は、該当するパッケージ部品の交換を行なうことで、簡単に、信頼性の高い、保守変更作業を遂行することが可能となる。

パッケージ部品原型が、LSI部品原型を中に取り込んで、部品(インスタンス)を生成する例として、電話呼制御部品(CSM_P:Circuit Switching Machine_part)の生成を例に説明する(図2)。

CSM_Pパッケージ部品原型は、図2に示すように、回線接続に関する制御が記述されている。即ち、加入者機能ブロックからの、加入者からの疑似発呼信号を、受け取ると、決定表を参照して、タスク名を決定し、このタスク名からタスクテーブルを参照して、タスクを実行する(これらの処理は電話呼制御処理の1部である)。

CSM_Pパッケージ部品原型は、決定表LSI部品原型(det_key)、タスクテーブルLSI部品原型(tskt)、ロギング表LSI部品原型(log)を組み込む(%include)ことによりCSM_P部品を生成する。

これらLSI部品原型は、表テーブルの名前、表に登録するデータのモード名、テーブルサイズ等がパラメータになっており、ユーザの設定により、種々のLSI部品(インスタンス)が生成できる。そして、取り込むLSI部品の種類に応じて、パッケージ部品原型から種々のパッケージ部品(インスタンス)が生成できる。

このように、部品原型のもつ柔軟性、適合性により、再利用の可能性が一段と高くなる。

4. おわりに

オブジェクト指向モデルに基づく、部品原型と、部品生成方式と、交換ソフトウェアへの適用について述べた。

【参考文献】

- [1] Goguen, J.A.: Parameterized Programming, IEEE Transactions on Software Engineering, Vol. SE-10, No. 5, 1984
- [2] E. Chigira, et al.: Development of Switching Software from Object-Oriented Program Parts Prototype, GLOBECOM, 1987
- [3] 樽屋 他: 交換ソフトウェア部品化方式の一考察 1988, 信学会, 交換研究会 SE87-152

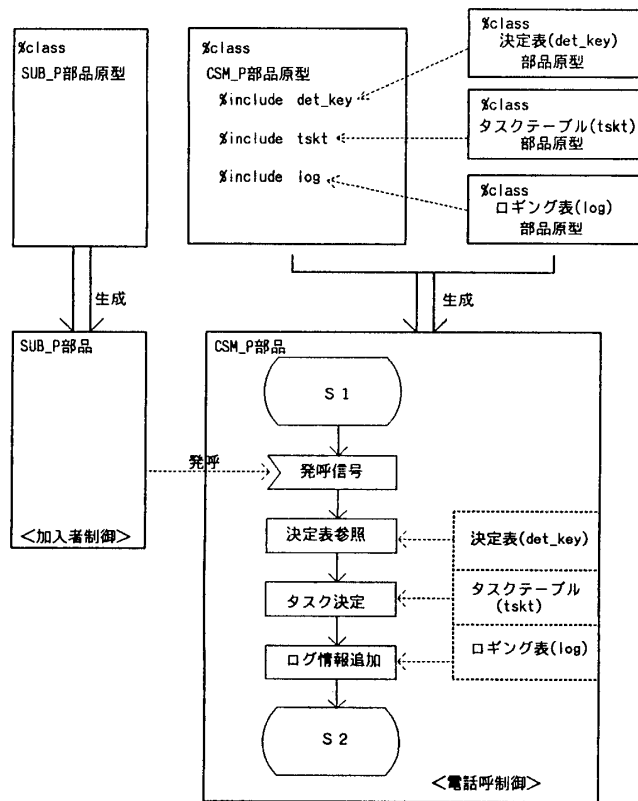


図 2 電話呼制御部品 (CSM_P) の生成