

# 部品合成によるプログラムの自動生成機能の実現方法

清田昌宏<sup>†</sup> Fang Miao<sup>†</sup> 山崎雄大<sup>†</sup> 上之園和宏<sup>‡</sup> 古宮誠一<sup>†</sup>

芝浦工業大学<sup>†</sup> 芝浦工業大学大学院<sup>‡</sup>

## 1. はじめに

情報システムの大規模化・複雑化に伴い、人手によって行われるソフトウェア開発では生産性や品質が低下してしまう。このような理由から、高品質・高生産性を得るために様々なソフトウェアの開発支援ツールが求められている。

本研究はソフトウェアの高品質、高生産性を確保するために、プログラムを自動生成するプログラム生成系、すなわち自動プログラミングシステムの実現を目指す。

本稿における自動プログラミングシステムとは人間向きの高度な仕様記述から、ユーザーの要求を満足するプログラムを、オブジェクト指向の再利用の考え方に基づいて、部品合成による技術を用い、完全自動で生成するツールである[1]。また自動プログラミングシステムは高生産性を確保するとともに、高品質を確保しなければならない。よって自動プログラミングシステムによって生成されるプログラムはバグが存在せず、与えられた要求仕様を満足するものでなければならない。

## 2. 部品合成による自動プログラミング

ユーザーの要求仕様を単純な要求仕様、すなわち部分仕様の集まりに分解し、個々の要求仕様を満足するプログラム、すなわち部品をあらかじめ用意、または生成し、これらを組み合わせるという方法が部品合成による自動プログラミングの方法である。部品合成による自動プログラミングの生成過程は図1のように3つのフェーズに分けられる[1]。

特にフェーズ2では部分仕様を基に該当する部品を検索する。検索した部品が部分仕様に合致しない場合はカスタマイズを行う。部品には、プログラムの制御構造を表す“骨組み部品”と、骨組み部品に呼び出されて機能の一部を構成す

る“部分部品”の二種類がある。

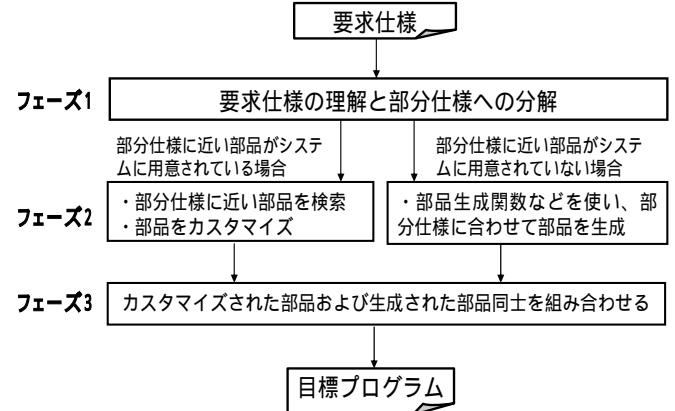


図1 部品合成による自動プログラミングの生成過程[1]

## 3. 自動プログラミングシステムで利用する技術

以下に部品合成による自動プログラミングシステムを実現するための再利用技術について述べる。

**フレームワーク**[2]：開発者がカスタマイズできるアプリケーションの骨組み。要求によって変化する部分であるホットスポットと、変化しない部分であるフローズスポットからなる。フレームワークを利用したプログラム開発では部品をホットスポットに埋め込むことで目標プログラムを作成する。

**継承**：あるクラスを雛形にして機能を付け加えることにより派生クラスを生成すること。

**オブジェクトコンポジション**：あるオブジェクトを内部に含む他のオブジェクトを作成することで、オブジェクトの持つ機能を拡張すること。

理想的には既存の部品をオブジェクトコンポジションにより組み合わせるだけで必要とする機能をすべて実現できるように用意すべきである。しかし、要求に合致した部品が存在しない場合は継承を用い、既存の部品を要求にあった部品にカスタマイズし、それらを組み合わせることで目標とする機能を実現する[3]。

## 4. 部品化再利用とオブジェクト指向

### 4.1. 部分部品の作成方法

オブジェクト指向の利点の一つとして、“再

“An Automatic Programming System by Composition of Reusable Program Components,” written by

<sup>†</sup>Masahiro Kiyota, Fang Miao, Yudai Yamazaki, Seiichi Komiya, Shibaura Institute Of Technology

<sup>‡</sup>Kazuhiro Uenosono, Graduate School Of Shibaura Institute Of Technology

利用性の向上”があげられる。この理由の一つとして、“継承”があげられる。親クラスを継承してサブクラスを生成すれば、親クラスのメソッドを自動的に継承するので、サブクラス特有のメソッドのみを記述すれば良いからである。部品合成による自動プログラミングでは部品を再利用するので、再利用性を向上させるために部品をオブジェクトとして作成する。

オブジェクト指向では一般的に具体的な、あるいは抽象的な“物”をオブジェクトに対応付ける。しかし、このようにすると再利用の単位が大きすぎて、実際の差分が少なくても記述量が多くなってしまい、再利用性が上がらない。例えば、“名前”と“住所”と個人情報の属性を表示するメソッドからなる“個人情報”クラスがあるとすると、この時、個人情報クラスを継承して、“電話番号”を追加したサブクラスを生成すると、個人情報の属性を表示するメソッドを全て書き換えなければならない。

このような問題を回避する方法として、オブジェクトを“名前”や“住所”などのデータ項目ごとに対応付ける方法が考えられる。このようにすると新たに記述しなければならないのは追加したデータ項目に対応するメソッドのみでよいので、再利用性を上げることができる。本研究ではデータ項目に対応したオブジェクトを部分部品として作成する[4]。

#### 4.2. 骨組み部品の作成方法

再利用性向上の技術の一つとして、先にあげたフレームワークがある。従来サブルーチンを共通部品として再利用していたが、メインルーチンの方を共通部品として再利用しようというものである。例えば“予約する”という処理において、扱うデータ項目に差があったとしても、“予約する”という処理のおおまかな流れには差が無いと考えられる。そこで共通の処理の流れをフレームワークとして作成する。本研究における骨組み部品はメインルーチンの処理の流れをフレームワークとして作成したものになる。

#### 4.3. 部品のカスタマイズ

本研究では部品のカスタマイズをユーザーからの要求を基に自動で行う。自動で要求に合うように部品をカスタマイズするには何らかの知識が必要である。本研究ではこのような知識を、部品と一対一に対応するフレームを用いて表現することを検討している。フレームには“どのような要求が与えられたらどのようなカスタマイズを行うのか”という知識と推論機能をあらかじめ持たせておく。フレームのスロットに要求仕様から獲得した情報が入力されることで、フレ

ームのデモンと呼ばれる付加手続きが起動し、部品のカスタマイズを行う。図2に部品のカスタマイズの例を示す。

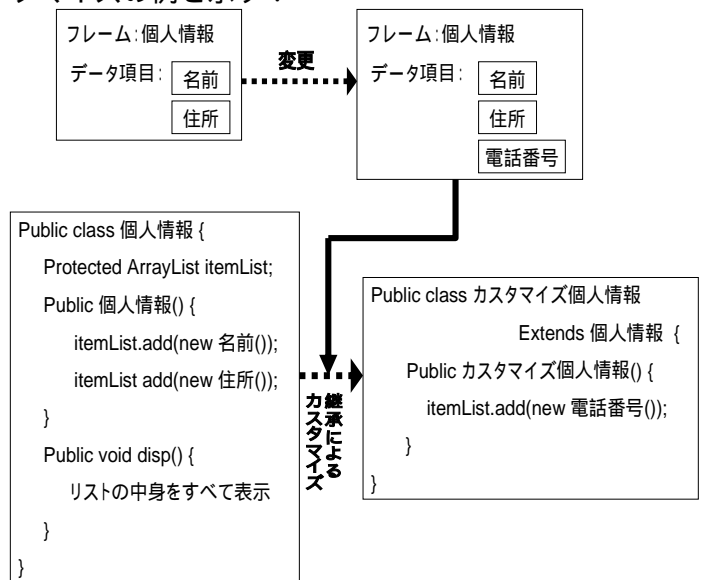


図2. 部品のカスタマイズの例

## 5. まとめ

本稿では、高品質・高生産性を確保するための手段として、オブジェクト指向の再利用の考え方に基づいた部品合成による自動プログラミングシステムを提案し、処理の流れを示した。また、システムで用いられる、再利用効率を上げるための部品の作成方法と、そのカスタマイズの手法について例と共に説明した。

## 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, vol28, no.10, pp.1329-1345, Oct. 1988.
- [2] 株式会社オージス総研, オブジェクトの広場, 00 入門, Happy Squeaking!!-オブジェクト指向再入門 第5回, <http://www.ogis-ri.co.jp/otc/>, (2004年10月現在)
- [3] 本位田 真一, 吉田 和樹, “オブジェクト指向における再利用のためのデザインパターン,” ソフトバンク パブリッシング株式会社, 1999
- [4] アプリテック株式会社, ビジネスロジック部品-実用的で効果的な部品化再利用システム, <http://www.applitech.co.jp/compo/>, (2005年1月現在)
- [5] 新田 克己, “知識と推論,” サイエンス社, 2002