

モジュール構造を考慮した形式的細粒度部品の提案 A Fine-Grained Component for Module Structured Formal Software

横山 仁[†]
Jin Yokoyama

織田 健[†]
Takeshi Oda

1 はじめに

形式手法 B-Method で開発された既存のソフトウェアから再利用可能な部品の生成及び登録をしておき、要求モデルとその部品らを元に合成することで、実行可能なソフトウェアを自動生成することが期待できる。しかしモジュール構造を考慮した部品では再利用性などの課題が発生する。そこで本稿ではモジュール構造を考慮した形式的細粒度部品の提案する。

2 背景

2.1 形式手法 B-Method

形式手法は数学を基盤としたソフトウェアの仕様記述、開発、検証技術である。形式手法は意味と構文が数学的に厳密に定義された形式で仕様記述言語を用いた仕様の記述により、仕様の曖昧さを排除し無矛盾性を保証する [1]。B-Method は形式手法の 1 つであり、仕様記述から実装までの流れを網羅している。記述はモデル、リファインメント、実装の 3 つから成る。モデルは抽象機械として記述される。

2.2 B-Method におけるモジュール構造

B-Method ではモジュール構造の設計も可能である。抽象機械は SEES、USES、INCLUDES、EXTENDS の 4 種の節を用いて他の抽象機械を取り込む。抽象機械は変数等を宣言し、それらの制約を設定し、操作部では制約を満たすように変数に代入する。その際に先述の 4 種の節の使い分けで、他の抽象機械の変数等の参照可能範囲や操作呼び出しの公開範囲の違い等を実現する。

実装では SEES、IMPORTS、EXTENDS の 3 種の節を使うことができる。実装ではリファイン元の抽象機械以外も参照することが可能である。実装からの参照は抽象機械側での参照の有無を問わないため、外部ライブラリの使用時でも抽象機械側で参照する必要がない。

以下にモデルもしくは実装で使う節の一部を記す。
INCLUDES 参照先の抽象機械の値、制約、操作を取り込む。取り込んだ操作の一部を参照元の操作として外部の抽象機械に公開することが可能。
EXTENDS 取り込んだ操作を全て公開するような INCLUDES と同等。
IMPORTS 実装のみで用いる。機能は INCLUDES 節に近い。

2.3 形式的細粒度部品

ソフトウェア開発において、関数や手続きをライブラリとしてまとめられる。これらは用途に応じて部品の粒度は異なり、粒度が細かい部品群は部品当たりの機能性を犠牲にするが、再利用性や信頼性の向上等の利点が生じる [3]。

形式手法 B-Method も部品の概念を適用できる。本稿では細粒度部品を関数等より粒度が細かい部品とし、形式的ソフトウェアから生成した細粒度部品を形式的細粒度部品と呼ぶ。この部品はモデルと実装の組であり、また部品のモデルを細分化モデルと呼び、これの生成過程をモデル細分化と呼ぶ。生成前に振舞いが同じ全モデルの表記方法が統一されるように変換し、リポジトリ登録することで字面一致検索から要求を満たす部品を探せる。

2.4 形式的細粒度部品におけるモジュール構造の課題

形式細粒度部品の生成及び結合の手法は中村によって提案されている [2]。しかしこの手法はモジュール構造が考慮されていないため、実際の B-Method 開発と乖離する。そこで本研究ではモジュール構造に対応した形式的細粒度部品の生成及び結合を目的とする。

3 形式的細粒度部品の課題点

形式的細粒度部品においてモジュール構造を考慮した際に複数の課題が発生する。以下の節で分けて述べる。

3.1 部品間の依存問題

異なるモジュール構造から生成された部品同士を利用して新しいソフトウェアを生成出来る方針を取る場合、部品は別の特定の部品に依存しないことが望ましい。そこでモデル細分化前に各抽象機械は参照先の抽象機械の操作部以外を取り込む方針を取る。例えば抽象機械 M1 が M2 を参照している場合、M1 に M2 の操作部以外を取り込んで M1' とする。この取り込みを展開と呼び、既存のソフトウェアの階層構造の違いを吸収する。

3.2 操作呼び出し

操作呼び出し文は呼び出し先を展開せずに部品にそのまま残して問題無いと考えられる。これは要求モデルの操作呼び出しと引数と戻り値の型が一致すれば、中身に関係なく同じ振舞いの操作を呼び出す部品として扱うことができるからである。ただし名前の置換で呼び出し名を抽象化する必要がある。

3.3 実装を展開する必要性

モデルを展開する方針を取ったが、実装は展開しない方針を取る。これはモデルの展開において操作を展開しないため、実装も展開する必要がないためである。そのため例えば実装同士を展開する必要はない。

4 モジュール構造を考慮した形式的細粒度部品の生成及び合成手法

3 章を踏まえ、モジュール構造を考慮した形式細粒度部品の生成及び結合手法を提案する (図 1,2)。なお図中の M1 から M5 は抽象機械、I1 から I5 は実装を表す。

[†]電気通信大学大学院情報理工学研究科情報学専攻

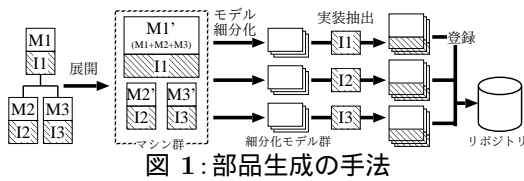


図 1: 部品生成の手法

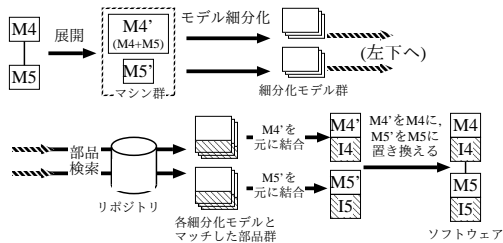


図 2: 部品結合の手法

4.1 部品生成

図 1 を用いて既存のソフトウェアから細粒度部品の生成過程を述べる。生成部品はリポジトリに登録され、4.2 節の結合に用いる。以下に一連の手法概要を述べ、その後各項の詳細を述べる。ただし以下の手法はモジュール構造の要求モデルの各抽象機械ごとに適用する。

展開 各抽象機械の参照先の抽象機械を参照元に展開。モデル細分化 表記方法の統一、操作の切り分け、各操作に関する情報の抜き出し。

実装抽出 各細分化モデルに対応する実装の抽出。

図 1 の展開では、M1 の参照先は M2 と M3 であり、M2 と M3 の操作部以外のコードを M1 に書き写し、M1' とする。参照先がない M2 も、そのまま M2' とする。

モデル細分化の手順は以下の通り。

1. 同じ振舞いのモデルの表記方法が統一されるように抽象機械を変換する。
2. 操作を最小の機能ごとに切り分ける。基本的には 1 行の代入文に分けられる。
3. 分けられた各操作に使用される変数等に関する情報を抽象機械から全て抜き出す。情報とは宣言や制約条件、初期化になる。
4. 各部品内の識別子を抽象化する。例えば変数名 v001、v002 等に置換する。操作呼び出しも OPCODE001 等と置換する。

実装抽出では、モデル細分化から得た細分化モデルに対応する実装を既存のソフトウェアの実装から抽出する。つまり M1' から得た細分化モデルは I1 からのみ実装抽出を行う。これは M1' と M1 の操作部が一致するためである。抽出することでモデルと実装の組で構成された部品を生成し、生成部品をリポジトリに登録する。

4.2 部品結合

無矛盾性を証明済みの要求モデルを入力し、それを元に実装を結合することでソフトウェアを出力できる。この際に 4.1 節にて生成し、リポジトリに登録した部品を使う。以下に一連の手法の概要を述べ、その後図 2 を用いて各自の詳細を述べる。ただし以下の手法はモジュール構造の要求モデルの各抽象機械ごとに適用する。

部品検索 要求モデルの細分化モデルに対応する部品をリポジトリから検索

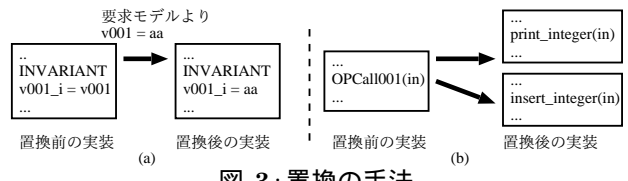


図 3: 置換の手法

部品結合 部品を結合し、要求モデルのソフトウェアを出力

まず要求モデルに対して、展開及び細分化で細分化モデルを生成する。この細分化モデルを検索キーとして、4.1 節で部品に登録したりリポジトリに対して検索を行う。検索の際に完全一致または制約条件を内包するような部品を検索結果として部品結合に用いる。

部品検索で得た部品を結合する。要求モデルを展開したものに各部品のモデルをあてはめ、その部品の実装を結合する。例えば M4' の細分化モデルとマッチングした部品のみを M4' の部品結合に用いる。各部品の識別子は置換されているため、結合時に実装の識別子名を要求モデルと同一にする。例えば図 3 の (a) であれば、要求モデルの変数 aa に対応する v001 を置き換える。また B-Method の操作呼び出し名では、呼び出し元の振舞いは呼び出し先の振舞いに影響せず閉じているため、3.2 節で取った方針から呼び出し先と元で引数と戻値の型が一致し、残りの識別子を一致させれば要求モデルを満たす振舞いになる。そのため図 3 の (b) のように型が一致すれば、異なる操作を呼び出すことができる。

図 2 の最後では M4' を M4 に差し替える。これは展開時に失った参照関係を再現するためである。最終的に結合された実装と元の要求モデルをまとめて、ソフトウェアとして出力する。

5 考察

モジュール構造の要求モデル及び既存のソフトウェアを用意して、提案手法を適用した。結果として、要求モデルを満たす実装を得ることができた。ただし人間の判断で行なった部分もあり、今後は全て機械的に行なえるように手法を細かく定める必要がある。特に操作呼び出しの実装抽出方法や結合時に必要なモジュール構造に関する情報が展開時に不足することが問題となった。

6 おわりに

本稿ではモジュールを考慮した形式的細粒度部品という目標に対し、展開の方針を取り、異なるモジュール構造の部品から新しいモジュール構造のソフトウェアを合成する手法を提案した。今後は手法の各手順を詳細に定める必要がある。

参考文献

- [1] 中島震, 来間啓伸. B Method による形式仕様記述, 近代科学社, 2007.
- [2] 中村文洋, B Method における部品再利用によるソフトウェア合成と高信頼ソフトウェア部品の整備, 情報処理学会論文誌, Vol. 54, No. 8, pp. 2012 - 2024, 2013.
- [3] 橋本靖, 山本晋一郎, 阿草清滋. Program slicing を利用したプログラムカスタマイザ. 電子情報通信学会技術研究報告, Vol. SS94, No. 10, pp. 73 - 80, 1994.