5H-07

モジュール構造に対応したソフトウェア合成手法

横山 仁†
電気通信大学情報理工学部総合情報学科

織田 健‡
電気通信大学情報理工学研究科総合情報学専攻

1 はじめに

我々は既存のソフトウェアから部品の再利用を行い、要求モデルから実行可能なソフトウェアを自動生成をするモデル充足ソフトウェア合成 (MSSS) 手法を提案している [1]。従来の MSSS 手法では抽象機械を階層的に構成するモジュール構造に対応していないため実際のソフトウェア開発への適用には不都合があった。本研究ではモジュール構造に対応した MSSS 手法を提案する。

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仕様記述言語を用いた仕様の記述より、仕様の曖昧さを排除し無矛盾性を保証する [2]。B-Method とは形式手法の 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 (Machine) 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を考慮した設計も可能である。B-Method において、抽象機械は SEES、USES、INCLUDES、EXTENDS の 4 種の節を使うことで、他の抽象機械を取り込むことができる。B-Method において抽象機械は定数や変数を宣言し、それらに対する制約を設定し、操作では制約を満たすか考慮しながら変数に値を代入する。その際に、先述の 4 つの節の使い分けで他の抽象機械の定数の値のみの参照の許可や操作呼び出しの可否等の違いを実現する。詳細は以下に記す [2]。

SEES : 参照先の抽象機械の変数や定数を参照する。
USES : SEES に加え、参照先の抽象機械の変数を元に制約を設定することが可能。
INCLUDES : USES に加え、参照先の指定した一部の操作を参照元の操作として外部に公開できる。
EXTENDS : INCLUDES と比べ、参照先の全ての操作を公開できる。

2.3 MSSS(モデル充足ソフトウェア合成) 手法

MSSS 手法 (図 1) は、B-Method で記述された要求モデルと合致した高信頼ソフトウェアを生成する手法である。MSSS 手法は、MSSS と MSFC 生成の 2 つから成る。MSSS は以下の一連の処理を行う。要求モデルを入力とし、操作を原則 1 代入ごとに分け、各代入に対する制約条件抽出を行うモデル細分化をする。その細分化モデルを用いて部品を検索する。検索結果の部品を結合し、

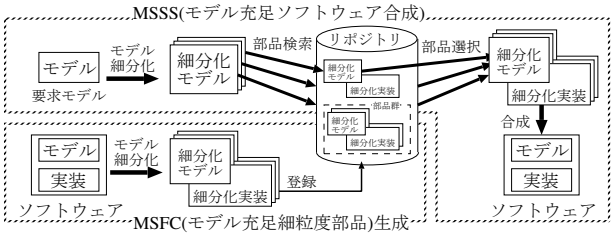

図 1: MSSS 手法

要求モデルを満たす B-Method のソフトウェアを自動生成する。MSFC 生成は B-Method で開発された既存のソフトウェアのモデルと実装の組を入力することで、非常に低機能な 1 操作のみのモデルと実装の組に細分化して、リポジトリに登録する。その際、部品検索時の名前依存を避けるため、細分化モデル内の識別子の名前は規則に沿って “v001, v002, ...” のように置換される。これらによって、部品の高い再利用性を実現している。

2.4 MSSS の課題

実際の B-Method の開発ではモジュール構造を考慮して、SEES 等の 4 種の節が利用される。しかし従来の MSSS 手法ではモジュール構造を考慮していないため、入力される抽象機械は 1 つに限られ、仕様はその内に全て記述されているものとしていた。これは本来の B-Method のソフトウェア開発と乖離しているため、構造化されたモデルを入力できない。そこで本研究ではモジュール構造に適応した MSSS 手法を検討する。

3 モジュール構造に対応した MSSS 手法の注意点

モジュール構造に対応した MSSS 手法について、方針の定め方で注意すべき点を述べる。

もし部品は SEES 等の節も残すような、参照関係を持ちうるものとして方針を定めた場合、細分化モデルと同等の振る舞いをする部品があっても SEES 節等の有無で別のものとみなされ、検索出来ないと考える。例えば細分化モデルと部品モデルで全体の操作や制約条件が一致していても、識別子を宣言している部分が別のモデルを参照しているか、1 つのモデルで完結しているかで異なる場合は検索できないと考えられる。このように部品の再利用性が著しく下がる。そこでモデル細分化前に一度全ての抽象機械を 1 つの抽象機械に全て展開し、従来通りのモデル細分化を行う方針を取った。

この方針を取った際に問題になるのが、展開することで元のモジュール構造の情報を失ってしまうことである。そのままモデルの結合を行えば、複数の抽象機械の入力に対し、出力される抽象機械は 1 つになる。また B-Method では別の抽象機械で同じ名前の識別子を宣言することが可能なため、名前空間も問題となる。

そこで展開時に再現の為の情報を残すことにした。残

A Software Synthsis Method Corresponding to Module Structure
†Jin Yokoyama, The University of Electro-Communications Department of Informatics
‡Takeshi Oda, The University of Electro-Communications Graduate school of Informatics

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

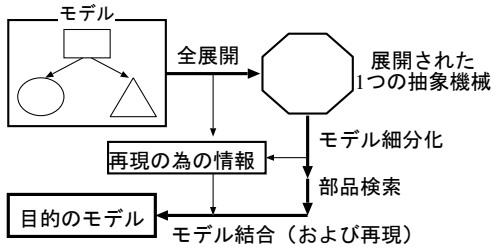

図 2: モジュール構造に対応した MSSS 手法

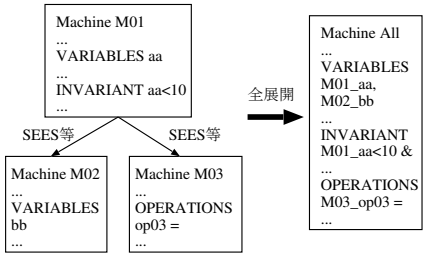

図 3: 全展開の例

す情報は、SEES 節等の抽象機械間の参照関係と各識別子が宣言されている抽象機械名である。例えば抽象機械 M01 が抽象機械 M02 を SEES 参照している情報を残し、M01 に変数 aa があるなら展開時に M01_aa と付与して情報を残す。これらの情報を元にモデルの結合時で入力されたモデルを再現する。

4 モジュール構造に対応した MSSS の提案

モジュール構造に対応した MSSS の全体の流れを図 2 に示す。従来の MSSS 手法のモデル細分化に手を加え、全展開とモデルの結合の手順を追加した。

4.1 全展開

図 3 のように各抽象機械の全識別子に対し、所属している抽象機械の名前を付与する。次に全ての抽象機械を 1 つの抽象機械に全展開する。展開の仕方は各節の列挙方法に従う。例えば VARIABLES であれば “,”、制約条件であれば “&” を各節の最終行以外の行の末尾に追加して繋げていく。

4.2 モデル細分化

途中までは従来のモデル細分化を行う。最終的に細分化された抽象機械の名前を置換するが、この際に VARIABLES や OPERATION の各操作の名前等、各識別子の置換前後の対応関係の情報を残す。例えば M01_aa が v001、M02_bb が v002 に置換された情報を残す。

4.3 部品検索

部品検索は従来通り、細分化モデルと部品モデルの字面一致により行う。

4.4 モデルの結合

従来の部品結合は実装の結合のみだが、本手法ではモデルの結合も行う。手順は以下の通り。

(1) 節 4.2 で残した置換前後の名前関係を用いて、部品群の各抽象機械の各識別子の名前を元に戻す。
(2) 図 4 の (a) のように元のモジュール構造の抽象機械群から同じ抽象機械名と SEES、USES、INCLUDES、EXTENDS をのみが記されたモデルを用意する。
(3) 各部品の SETS、CONSTANTS、VARIABLES 節の各行を適切な抽象機械へ配置して再現する。具体的には図 4 の (b) の実線の矢印のように、節 4.1 にて各識別子に付与された抽象機械の名前と一致する抽象機械名を手順 2 で用意した抽象機械群から探し、一致した抽象機械内の各節にて配置する。識別子の数が 2 つ以上に成る場合は “,” で繋げる。例えば M01_aa は M01 の VARIABLES に配置される。
(4) 手順 3 と同様に、部品の INITIALISATION を “||”、操作の名前を “;” で繋げて配置する。つまり OPERATIONS 部分は PRE 以下が無い操作が並ぶ。
(5) 部品モデルの PRE 部分及び THEN 部分をそれらが所属する操作名を元に、手順 4 で配置した操作へ配置する。PRE 部分は “ & ”、THEN 部分は “||” で繋げる。
(6) モジュール構造で最も下位の抽象機械が扱える変数や定数のみが記された制約条件を部品モデルから抜き出して配置する。対象の抽象機械に対して抜き出せる制約条件が無くなったら 1 つ上の抽象機械に移行して同様の処理を行う。この処理を最上位の抽象機械に到達するまで繰り返す。つまりある抽象機械に対し、その抽象機械内もしくは下位の抽象機械にて宣言されている識別子を含む制約条件を抜き出す。図 4 の (b) の波線の矢印のようになる。

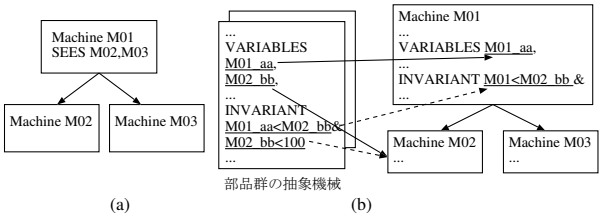

図 4: モデルの結合の一部の例

5 考察

小規模なモデル群ではモデルの結合にて制約条件以外は再現できると考える。入力した抽象機械の各節と再現時の各節にて等しい配置ができたからである。しかし制約条件を再現するアルゴリズムを確立出来ていない。また今回対応したのはモデルのみであり、実装部分の再現や MSFC 生成への影響はまだ考慮できていない。ただし実装でも手法が大きく変わらないと考えている。

6 おわりに

本稿ではモジュール構造に対応した MSSS 手法とそれのアルゴリズムについて提案した。おおまかな手法は決定できたが、制約条件の再現はまだ確立できていない。今後は制約条件の再現、実装部分の再現、MSFC 生成への影響の考慮が課題となる。

参考文献

[1] 中村丈洋. 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電気通信大学 大学院 電気通信学研究科 博士 (工学) 学位論文, 2014.

[2] 中島震, 来間啓伸. B Method による形式仕様記述, 近代科学社, 2007.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.