B-008

モジュール構造と段階的詳細化に対処した形式的ソフトウェア合成手法

Formal Software Synthesis Method Dealing with Modular Structure and Stepwise Refinement

松田 蓮† **織田 健**†
Ren Matsuda **Takeshi Oda**

1 はじめに

ソフトウェア開発の信頼性向上やコスト増大への対処として形式手法が注目されている。我々は形式手法の一つである B Method を用いたソフトウェア合成手法（MSSS 手法）を提案しており [1]、これをモジュール構造と段階的詳細化に同時対応させる手法が提案された [2]。本提案は MSSS 手法をモジュール構造と段階的詳細化に対応させる概要をまとめたが、部品結合手法に不整合が生じていた。よって本稿ではモジュール構造と段階的詳細化に対応した新たな部品結合手法を提案する。

2 背景と目的

2.1 形式手法 B Method

形式手法は数学的基盤に基づく形式仕様記述言語を用いる対象のモデル化や仕様検証などを含む技術である。形式手法の一つである B Method はソフトウェア開発手法で、ソフトウェアの仕様記述からコード生成までの流れを支援する。ソフトウェア仕様（モデル）をリファインメント・実装へと段階的に詳細化され、モデルの無矛盾性検証と詳細化の各段階間の整合性検証によりソフトウェアの正しさを保証する [3]。また B Method では複数のモデルから成るモジュール構造のモデルも記述可能で、各モデルは互いに独立に詳細化される。

2.2 MSSS 手法

我々は B Method で記述した要求モデルを入力とし、要求モデルを満たす合成ソフトウェアを出力する MSSS 手法を提案している [1]。MSSS 手法は既存ソフトウェアから細分化部品を生成するモデル充足細粒度部品生成（MSFC 生成）と、要求モデルからソフトウェアを合成するモデル充足ソフトウェア合成（MSSS）で構成される。なお、本手法は単一のモデルと実装の組が対象である。

MSFC 生成では既存ソフトウェアのモデルを細分化し、各細分化モデルに対応する細分化実装を抽出し、それらの組を細分化実装としてリポジトリに登録する。文字列一致での検索の為、部品の識別子は登録時に全て一意に定まるよう置換される。

MSSS では要求モデルから細分化モデルを生成し、各細分化モデルを検索キーとして対応する細分化部品をリポジトリから取得する。取得した部品群から結合可能な部品を選択・結合し、要求モデルを満たす実装を合成する。合成ソフトウェアの識別子は要求モデルに沿う形に統一する。合成できず不足した部品は人の手で記述する。

2.3 MSSS 手法の複雑なソフトウェア構造への対応

MSSS 手法をモジュール構造と段階的詳細化に対応させる為に、モデルの参照関係を分離するモデル展開と部品結合時に詳細化段数を統一する段数統一を用いてソフトウェアの部品化と合成を行う手法が提案された（図 1）[2]。また、モジュール構造と段階的詳細化両方の性質を併せ持つ詳細化途中から導入されるモジュール（ここでは潜在モジュールと呼ぶ）への対応も提案された。

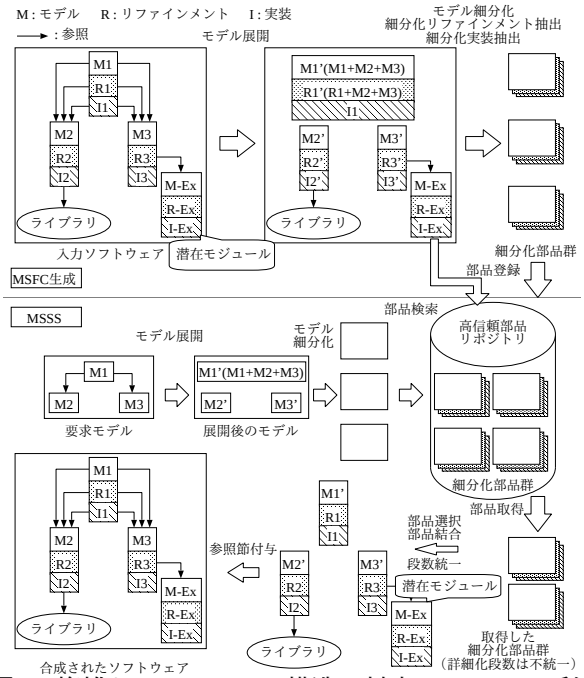

図 1: 複雑なソフトウェア構造に対応した MSSS 手法

2.3.1 モデル展開

モデル展開は参照先のモデルの変数や制約条件を参照元のモデルとリファインメントに書き込み、参照関係を断つ手法である。モジュール構造を成す各モデルは独立に詳細化されるため、参照を分離して部品化し、結合時に要求モデルの参照関係を基に部品の参照関係を構築することでモジュール構造のソフトウェアを合成する。

2.3.2 段数統一

既存ソフトウェアの詳細化段数は一定でなく部品化時に詳細化段数は変更しないため、リポジトリ内の細分化部品は詳細化段数が異なる。異なる詳細化段数の部品を結合する為にリファインメントの複製・統合を利用して詳細化段数を統一する手法を段数統一と呼ぶ。

2.3.3 潜在モジュールへの対応

詳細化途中から導入されるモジュール（潜在モジュール）はモデルでは参照されず、リファインメントから初めて参照される。細分化部品から潜在モジュールへの参照関係を断つと、細分化モデルからは潜在モジュールの情報が得られない為に潜在モジュールを取得できない。

†電気通信大学大学院情報理工学研究科情報学専攻

そのため、部品生成時に細分化リファインメントから潜在モジュールへの参照関係を残す。また潜在モジュールは細分化せずそのまま部品化する方針が取られた。

2.4 先行研究における課題

モデル展開により参照先のモデルの変数や制約条件をモデルとリファインメントに展開すると、上位モジュールの細分化リファインメントに下位モジュールの変数の制約条件が残り、上位モジュールの合成リファインメントに下位モジュールのモデルで定義されるべき変数の制約条件が残り文法エラーが生じるという課題がある。

2.5 研究目的

第 2.3 節の先行研究は不完全で、部品結合に問題があった。そこで本研究ではモジュール構造と段階的詳細化を考慮した部品結合に着目し、MSSS 手法での複雑なソフトウェアの合成を目指す。

3 モジュール構造と段階的詳細化を考慮した部品結合手法

3.1 提案手法の概要

3.1.1 モデル展開の変更点

第 2.4 節の課題解決のために従来のモデル展開を一部変更する。参照先モデルの変数や制約条件は参照元モデルのみに展開し、上位モジュールのリファインメントに参照先モデルの変数や制約条件は展開しない。ただしリファインメントの操作には下位モジュールの変数が残ったまま部品化される。部品の再利用時、細分化リファインメントの操作にある参照先モデルの変数の識別子は要求モデルに沿う形に統一する必要がある。

3.1.2 要求モデルに沿う形の識別子統一

要求モデルのモデル展開で参照元モデルに展開した変数とそれが宣言された参照先モデルは 1 対 1 対応で、部品検索時の要求モデルの識別子置換も置換前後の変数は一意に定まるため、部品検索用の識別子から元の要求モデルの全ての識別子を特定できる。また細分化部品のモデル変数はリファインメント変数・実装変数へと詳細化されるが、モデル展開で参照元モデルに展開した参照先のモデル変数は詳細化先を持たない。よって細分化部品内に詳細化先が存在しないモデル変数は別モデルから展開されたモデル変数だと分かる。つまり検索で用いる細分化モデルの識別子と要求モデルの識別子の対応、細分化部品内の詳細化先の無いモデル変数の特定により細分化リファインメントの操作に登場するモデル変数と対応する要求モデルの識別子は一意に定められる（図 2）。

3.2 部品結合の手順

1. 要求モデルのモデル展開時に参照先の各モデル変数がどのモデルの変数なのかの対応関係を保持する。
2. モデル細分化後、部品を検索・取得する。
3. 取得した部品の細分化モデルの変数について、詳細化のリンク条件の対応関係から部品中に詳細化先が無いモデル変数を特定する。
4. 細分化部品を結合し要求モデルの参照関係を基に結合後の部品に参照関係を付与する。

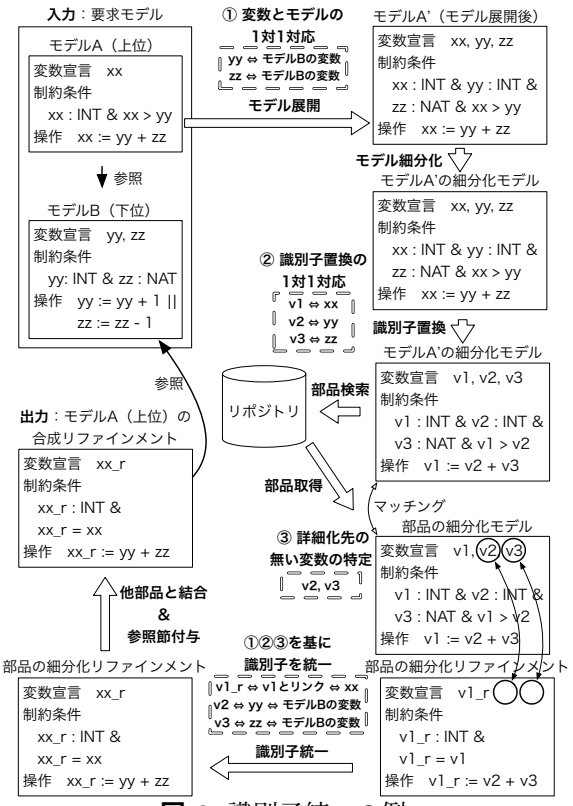

図 2: 識別子統一の例

5. 手順 1 で保持した対応関係と手順 3 で特定したモデル変数を基に、要求モデルに沿うように結合後の部品の識別子を統一する。

3.3 MSFC 生成での識別子置換

MSFC 生成の識別子置換では細分化モデル内で詳細化先が無いモデル変数を特定し、それと同じ識別子に細分化リファインメントの操作内の参照先の変数を置換する。

4 考察

本手法は提案に留まっており、実験による検証が不可欠である。モデル展開の変更と識別子の統一方法の改善により、従来では文法エラーとなる部品の再利用が可能になり MSSS 手法の汎用性の拡張が期待できる。

5 終わりに

本稿ではモデル展開の変更とモデル展開で展開する変数とそれを宣言したモデルの対応関係と識別子の置換前後の対応関係を利用した識別子統一方法を提案した。今後は実験による妥当性検証と判明した課題の検討を行う。

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 電気通信学研究科 博士（工学）学位論文, 2014

[2] 松田 蓮, 織田 健. モジュール構造と段階的詳細化を考慮した形式的ソフトウェア合成手法. 情報処理学会第 84 回全国大会講演論文集, vol.1 pp.295-296, (2022.03)

[3] 来間 啓伸. B メソッドにおける形式仕様記述. 近代科学社. 2007