4L-08

形式的ソフトウェア合成手法の複雑な例題による検証

松田 蓮†
電気通信大学大学院情報学専攻

織田 健‡
電気通信大学大学院情報学専攻

1 はじめに

ソフトウェア開発における信頼性保証やコスト削減の戦略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）を提案している [1]。単純なソフトウェアのみを対象とした本手法をモジュール構造と段階的詳細化に対応させる手法 [2] が提案されたが、手法の検証が不十分だった。そこで本研究では先行研究の検証不足を埋める検証を行い、手法の妥当性を検証する。

2 背景と目的

2.1 形式手法 B Method

形式手法は集合論を軸とした仕様記述によるソフトウェアのモデル化・検証を含む技術である [3]。ソフトウェア仕様（モデル）は中間仕様・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が保証される [3]。また、B Method では複雑なモジュール構造も構成可能で、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。

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実装を抽出し、細分化モデルと実装の組を細分化部品としてリポジトリに登録する。

MSSS では要求モデルから生成した各細分化モデルに対応する細分化部品群をリポジトリから検索・取得する。部品を適切に選択・結合し、要求モデルを満たす実装を合成する。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への対応

MSSS 手法のモジュール構造と段階的詳細化への対応として、モデルの参照関係を断つモデル展開と部品の詳細化段数を統一する段数統一を用いた MSSS 手法の拡張案が提案された（図 1）[2]。これはモジュール構造と段階的詳細化両方の性質を持つ詳細化途中から導入されるモジュール（以後、潜在モジュールと呼ぶ）にも対応した。

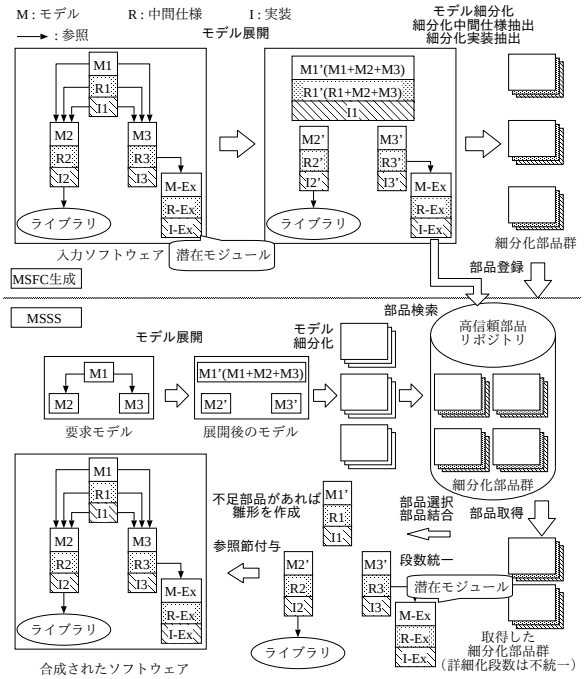

図 1: 複雑なソフトウェア構造に対応した MSSS 手法

2.4 研究目的

2.3 節の提案は単純な要求のみでの検証のため、検証不足が指摘されていた。本研究では、複雑な要求を用いて検証不足を埋め、先行研究の妥当性を検証する。

3 複雑なソフトウェアに対応した MSSS 手法

3.1 MSFC 生成

モジュール構造と段階的詳細化を考慮した MSFC 生成（図 1 上部）では、まずモデル展開により参照先のモデルの情報を参照元のモデルに書き込み参照関係を断つ。続いてモデル展開後の各モデルから生成した細分化モデルを基に細分化中間仕様・実装を抽出する。最後に、生成された部品の識別子を置換しリポジトリに登録する。

3.2 MSSS

モジュール構造と段階的詳細化を考慮した MSSS（図 1 下部）では、まず要求モデルをモデル展開して参照関係を断つ。モデル展開後のモデルから生成した細分化モデルの識別子を置換し部品を検索する。検索で取得できた部品群の詳細化段数を調節し部品を結合する。結合可能な全部品を結合後、要求モデルの参照関係を基に部品に参照関係を構築し、不足部品の雛形を作成する。

4 実験

4.1 概要

本実験は 3 章の複雑なソフトウェアに対応した MSSS 手法 [2] の検証を目的とする。表 1 の通り、本実験は先

Verification of Formal Software Synthesis Method with Complex Example

†Ren Matsuda, The University of Electro-Communications, Graduate School of Information and Communication Engineering

‡Takeshi Oda, The University of Electro-Communications, Graduate School of Information and Communication Engineering

表 1: 検証するソフトウェアの要素

ソフトウェアの要素	先行研究	本実験
モジュール構造	○	○
モジュールの深さ	1 段	2 段
SEES	×	○
INCLUDES	○	○
IMPORTS	○	○
複数ヶ所の SEES	×	○
モジュールの名前付け替え	×	○
ライブラリ	×	○
中間仕様 (下位)	1 段	1 段
中間仕様 (上位)	0 段	2 段
潜在モジュール	○	○

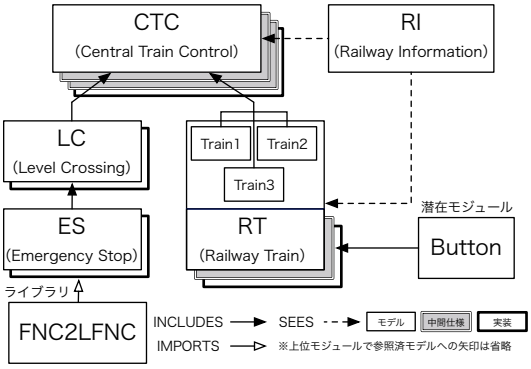

図 2: MSFC 生成に利用したソフトウェア

行研究の実験で検証できた要素を全て網羅しつつそれ以外の要素も含むため、より実践的な検証ができる。実験は 3.1 節の MSFC 生成を検証する実験 (1) と、3.2 節の MSSS を検証する実験 (2) を行う。

4.2 実験 (1)

実験用ソフトウェアと実験手順

列車自動運行システム（図 2）を対象に [2] が示す手順に則り実験した。

評価基準

実験が成功したかどうかは次の基準で判断する。部品は再利用時に整合性が保証できれば良いため、部品自体が整合性を保つかは考慮しなくて良い。

(1) 手順通りに機械的に実験を進められたか?

結果

細分化部品の抽出の際、従来の部品抽出手法 [4] の適用範囲外のため、抽出できない実装が存在した。適用範囲内の部品については問題なく抽出できた。そのため、部品抽出手法の範囲内であれば評価基準 (1) は満たしていたと言える。

4.3 実験 (2)

実験用ソフトウェアと実験手順

図 2 とは異なる列車運行システム（図 3）を対象に [2] が示す手順で実験した。

評価基準

実験が成功したかどうかは次の基準で判断する。

(1) 手順通りに機械的に実験を進められたか?

(2) 合成ソフトウェアの整合性は保証できたか?

結果

実験 (1) で生成に失敗した影響で要求モデルに必要な部品が一部取得できなかったが、それ以外は問題なく取得・結合できた。よって、評価基準 (1) は満たしていたと言える。また、不足部品は人の手で記述したため全自動でソフトウェアを合成した訳ではないが、完成したソフトウェアを B Method の開発環境 Atelier B で検証した結果、整合性は保証できた。

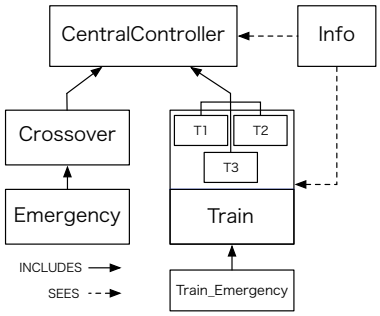

図 3: MSSS で利用した要求モデル

5 考察

5.1 部品抽出手法の不備

部品抽出手法 [4] は細分化モデルに対応する細分化部品を操作の返り値を基に判断する。非決定的値を扱う ANY 文に対応する実装の操作も同様に返り値から対応関係を判断するが、本実験で扱ったソフトウェアは ANY 文の非決定的値を IF 文の条件式に利用しており、[4] はこれを考慮していなかった。その他にも、モデルの変数と実装の変数の対応が 1 対 1 でなければならないなど、実践利用には不十分であると言える。今後は部品抽出手法の改善が必須である。

5.2 手法の妥当性

本実験では先行研究で検証した要素を包含する形で実験したが、5.1 節で述べた不備以外では手法の問題点は見つからなかった。よって、部品抽出手法を適切に整備できれば MSSS 手法の複雑なソフトウェアへの対応手法は妥当性があると考えられる。

6 終わりに

本稿では先行研究による単純なソフトウェアを用いた検証の不足を埋めるべく、複雑なソフトウェアを用いた MSSS 手法の検証を行い、部品抽出手法 [4] の課題点を発見した。今後は部品抽出手法の改善を行い、MSSS 手法の発展を目指す。

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

[4] 叶野 英俊. 形式的ソフトウェア部品生成のための実装抽出手法. 情報処理学会第 80 回全国大会講演論文集, vol.1 pp.219-220, (2018.03)