

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

松田 蓮[†]

電気通信大学情報理工学域

織田 健[‡]

電気通信大学大学院情報学専攻

1 はじめに

ソフトウェア開発の信頼性低下やコスト増大の問題には形式手法が注目されている。我々は形式手法の1つである B Method を用いたソフトウェア合成手法を提案している [1]。しかし、本手法はモジュール構造と段階的詳細化に未対応で、手法の汎用性が低い。これらに別々に対応する手法は提案されていたが、同時に対応可能な手法は未提案であった。よって本研究ではモジュール構造と段階的詳細化に同時に対応可能な手法の提案を目指す。

2 背景と目的

2.1 形式手法 B Method

形式手法は数学を基盤とした仕様記述を核としたモデル化や仕様検証などを含む技術である。B Method は形式的なソフトウェア開発手法で、モデルをリファインメント・実装へと段階的に詳細化することで仕様記述からコード生成までの流れを支援する。モデルの無矛盾性検証や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 [2]。また、B Method では複数のモデルから成るモジュール構造も記述可能である。

2.2 MSSS 手法

我々は B Method で記述された要求モデルを入力とし、細分化した要求モデルに対応する高信頼ソフトウェア部品をリポジトリ内から検索し、それらを機械的に結合して要求モデルを満たすソフトウェアを合成する MSSS 手法 [1] を提案している (図 1)。MSSS 手法は既存ソフトウェアから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細分化モデルを生成し、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。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。結合できず不足した部品は人の手で記述する。

2.3 先行研究

MSSS 手法のモジュール構造への対応 [3][4] では参照先のモデルの情報を参照元へ書き込み参照関係を断つモデル展開を行った各モデルに従来の MSSS 手法を用い、部品結合時に参照関係を構築する手法が提案された。

Formal Software Synthesis Method Considering Module Structure and Stepwise Refinement

[†]Ren Matsuda, The University of Electro-Communications, School of Informatics and Engineering

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

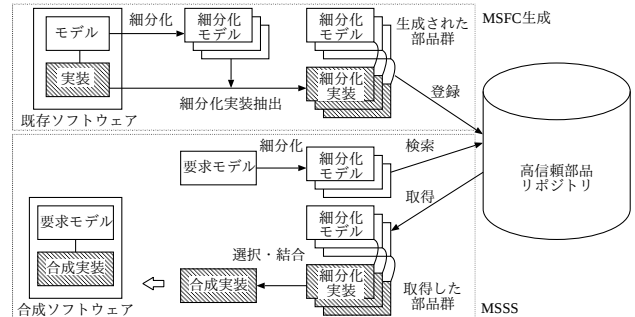


図 1: MSSS 手法

MSSS 手法の段階的詳細化への対応 [5] では細分化モデルを基に細分化リファインメント・実装を抽出する手法と、異なる詳細化段数を持つ部品同士の詳細度を調整して部品を結合可能にする段数統一手法が提案された。

2.4 研究目的

MSSS 手法はモジュール構造と段階的詳細化に対し、互いに独立して研究されていた。本研究ではこれらの手法を統合し、MSSS 手法の汎用性を高めることを目指す。

3 課題と解決方針

モジュール構造に関する手法と段階的詳細化に関する手法を統合すると、モデルでは参照されず詳細化中に導入されるモジュール (ここでは潜在モジュールと呼ぶ) を考慮する必要がある。部品生成時に潜在モジュールへの参照を断つと、部品結合時に要求モデルから潜在モジュールの情報を得られず部品を適切に取得できない。また潜在モジュールはモデルには情報がなく、細分化すると不整合が生じるため、潜在モジュールは細分化せず、これを参照する部品は参照関係を保ったまま登録する。

また、未提案だったモジュール構造を考慮した不足部品対応案は段階的詳細化に関する手法で提案された対応案をほぼそのまま利用すれば問題ないと考えられる。

4 モジュール構造と段階的詳細化を考慮した MSSS 手法

4.1 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 2 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とリファインメントへ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リファインメント・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。

4.2 MSSS

モジュール構造と段階的詳細化を考慮した MSSS (図 2 下部) では、まずモジュール構造を持つ要求モデルを

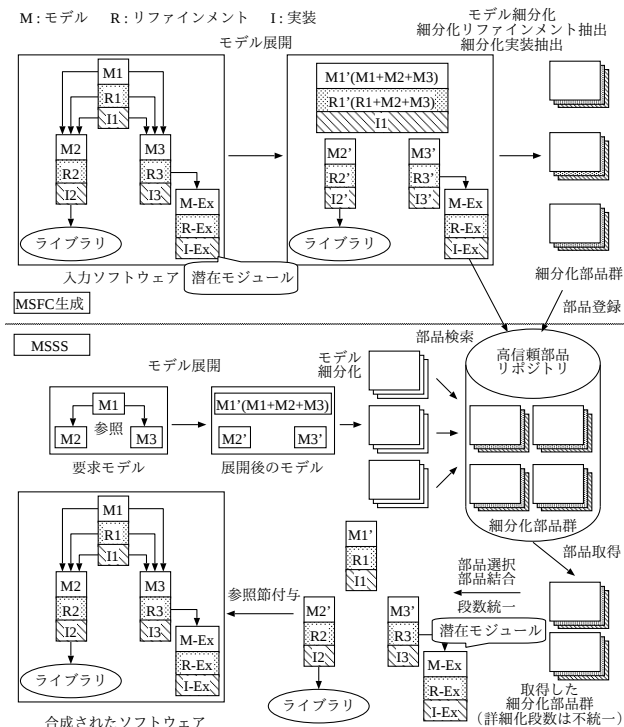


図 2: モジュール構造と段階的詳細化への対応

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を前節と同様に識別子を置換し、検索キーとしてリポジトリ内を検索する。検索により取得できた部品群から部品同士の結合ができるように詳細化段数を調節し部品を結合する。結合できる全ての部品を結合後、要求モデルの参照関係を基に結合した部品にも参照関係を構築し、不足部品の雛形を作成する。

4.3 詳細化段数の統一

既存ソフトウェアの持つ詳細化段数はソフトウェアによって異なるため、生成される部品の詳細化段数は統一されていない状態でリポジトリに登録される。よって部品検索で得られた部品群の詳細化段数は不統一であり、詳細化段数が等しいことを前提とする部品結合ができないため、部品結合時に詳細化段数を統一する。段数の統一には隣接する2つのリファインメント（実装）を詳細化の整合性を崩さない範囲で1つにまとめるリファインメントの統合と、モデルやリファインメントを複製して詳細化段数を増やすリファインメントの追加の2つの手法を利用する。

4.4 潜在モジュールへの対応

潜在モジュール（図2 M-Ex）を参照する細分化リファインメントは参照関係を残し、潜在モジュールはモデル展開しない。また、潜在モジュールは細分化せずそのままリポジトリに登録する。

4.5 不足部品への対応

モジュール構造と段階的詳細化を考慮した不足部品の雛形作成手法では人の手で部品を記述する際に人が把握すべき情報を持つ雛形を作成する。既に合成できた部品からその不足部品が持つべき変数の制約条件、操作名、参照節などを抽出して雛形を作成する。

5 実験

提案したモジュール構造と段階的詳細化を考慮したMSSS手法の評価実験として、各モデルの詳細化段数が0~1段で参照関係が子参照であるソフトウェアを複数作成し（一部は潜在モジュールを持つ）、モジュール構造の要求モデルを満たすソフトウェアを合成する実験を行った。MSFC生成では細分化部品が計38個生成され、MSSSでは要求モデルから生成された計22個の細分化モデルに対し17個の部品が合成できた。不足した5個の部品は雛形を作成し、それを基に人の手で部品を記述した。最終的な合成ソフトウェアは要求モデルを満たすことを示せた。なお、細分化実装抽出時の従来の抽出手法の適用できなかった実装は人が判断して抽出した。

6 考察

段階的詳細化を持つ部品を結合しそれらがモジュール構造を構成できたこと、またその合成ソフトウェアが要求モデルを満たすことを示せたことにより、提案手法の妥当性が示されたと考えられる。また、潜在モジュールへの参照を残した部品の再利用により潜在モジュールを再利用できたため、潜在モジュールへの対応も適切なものであったと考えられる。また、モデル展開をリファインメントにも拡張したが細分化部品生成時に展開したモデルの情報が失われてしまったため、そもそも展開する意味が無かったと分かった。本手法の課題として、実験用ソフトウェアではリファインメントが2段以上あるものや孫参照などの複雑なモジュール構造を持つソフトウェアによる検証ができておらず実験検証が不十分であること、また新たに判明した先行研究の課題として細分化実装の抽出手法が不十分であることが挙げられる。

7 終わりに

本研究ではMSSS手法のモジュール構造への対応と段階的詳細化への対応という独立した2つの研究の統合によるモジュール構造と段階的詳細化に対応したMSSS手法を提案した。統合の課題である潜在モジュールには、それ自体は細分化せずに参照関係を残したまま部品化して対応した。今後は手法の更なる実験検証と、先行研究の不十分な部分の対応が課題となる。

参考文献

- [1] 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. 電気通信大学 電気通信学研究科 博士（工学）学位論文, 2014
- [2] 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
- [3] 横山 仁. モジュール構造を考慮した形式的細粒度部品の提案. 第17回情報科学技術フォーラム論文集, vol.1 pp.135-136, (2018.09)
- [4] 岡安 悠. モジュール構造を考慮した形式的細分化実装の生成手法. 第18回情報科学技術フォーラム論文集, vol.1 pp.127-128, (2019.09)
- [5] 叶野 英俊. リファインメントを考慮した形式的ソフトウェア合成アルゴリズム. 第18回情報科学技術フォーラム論文集, vol.1 pp.129-130, (2019.09)