

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

Generation Method of Fine Grained Formal Implementation Considering Module Structure

岡安 悠[†]
Yuu Okayasu

織田 健[†]
Takeshi Oda

1 はじめに

形式手法である B method で記述されたソフトウェアはモデルと実装の組で構成される。我々は細分化された B Method のモデルに対応する細粒度部品の再利用によりソフトウェアを機械的に構築する手法を提案しているが、モジュール構造への対応はモデルのみにとどまっておらず、実装部分での対応が必要である [1][2]。再利用性を高めるため、細粒度部品は部品間の呼び出し関係を持たない状態ではなければならない。本研究ではモジュール構造のモデルに対応する 3 種類の部品と、細分化実装の生成手法を提案する。

2 研究背景

2.1 形式手法 B Method

B Method は数学に基づく記述により、仕様記述、仕様検証を行える形式手法である [3]。B Method の記述はモデルと実装の組によって構成され、モデルと実装は整合性が保たれている。モデルは他のモデルの変数や操作を参照するモジュール構造を持つことが可能である。

2.2 先行研究

我々は B Method で作られた既存のソフトウェアのモデルと実装から成る部品を、新規要求のモデルに合わせて結合することで新しいソフトウェアを合成する研究を行っている。部品の生成は図 1 のように次の手順で行われる。まず既存ソフトウェアのモデルから操作の個々の代入文に沿って必要な変数、制約条件を抜き出し分割し、細分化モデルを作る。細分化モデルに対して実装から必要な操作、制約条件や変数を抽出し細分化実装を作る。細分化実装と細分化モデルの組を部品としてリポジトリに登録する。登録した部品は新規要求に沿って結合され、新規ソフトウェアと成る。これらの手法はモジュール構造に対応していない。先行研究ではモデルにおいて、参照先のモデルの変数や制約条件を参照元のモデルに展開するという手法をとり、モジュール構造を持つモデルの細分化、合成手法について提案している [2]。しかしながら実装に関しては提案されていない。本稿ではモジュール構造を持つソフトウェアを構築可能な部品と細分化実装の生成手法について述べる。

3 モジュール構造に対応した 3 種類の部品

従来手法で作成された部品はモジュール構造ではないモデルと実装を扱えるが、モジュール構造を持つモデルと実装を扱うには、この部品だけでは不十分である。モジュール構造では他モジュールを参照する上位モジュールと、参照される下位モジュールという参照関係がで

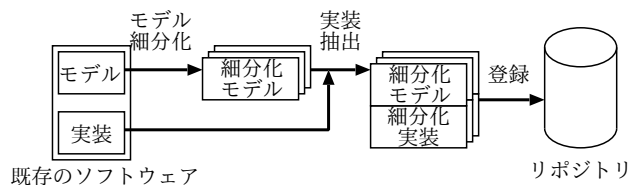


図 1: 部品生成

る。参照関係がある構造を構築可能な部品に関して考慮すべき点として以下があげられる。

部品間の依存関係 部品は全てのモジュール構造に対応できなければならない。しかし、細分化された実装やモデルは参照した操作名や変数名が残っており、他の部品との依存関係がある。部品に依存関係が残っていると、汎用性が低下するため、部品同士の依存関係を取り除かなければならない。

結合時の参照関係付与 部品は依存関係がない状態でリポジトリに登録されている。しかし、合成時にはモデルに沿って依存関係を付与し、モジュール同士の参照関係を構築できる必要がある。

部品の粒度 汎用性を高めるために、部品は 1 つの命令文に沿って生成された単純な部品の必要がある。

ライブラリ呼び出し 実装では参照したライブラリの操作を呼び出すライブラリ呼び出しが可能であり、これに対応した部品が必要である。

操作呼び出し 参照したモデルの操作を呼び出す操作呼び出しに対応した部品が必要である。

以上を考慮すると 3 種類の部品が考えられる。各部品の詳細は以下に示す。

操作呼び出しなし部品 従来手法で生成される、操作呼び出しを含まない細分化モデルを持つ細分化実装との組で構成される部品。

操作呼び出し部品 モジュール構造を持つモデルと実装で、操作呼び出しに対応した部品。操作呼び出しを持つ細分化モデルと細分化実装の組で構成される。

ライブラリ呼び出し部品 標準的に備わっているライブラリ呼び出しに対応した部品。操作呼び出しを含まない細分化モデルとライブラリ操作呼び出しの組で構成される。

これらの部品の使用によりモジュール構造へ対応できると考えられる。しかしながら、従来手法 [4] では、これらの部品を構成する細分化実装の生成が困難である。そのため、本稿では従来手法を考慮した上で、モジュール構造に対応した細分化実装の生成手法を提案する。

4 細分化実装の生成手法

3 章を考慮した上で細分化実装の生成手法を述べる。細分化実装は以下の手順により生成される。

[†]電気通信大学大学院情報理工学専攻

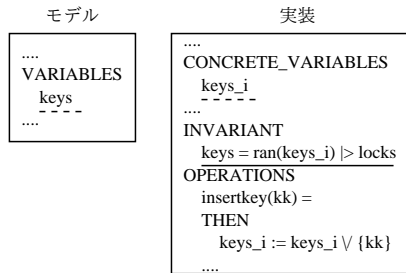


図 2: モデル変数と実装変数の対応関係

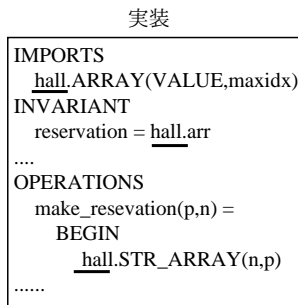


図 3: ライブラリの参照

4.1 参照関係除去

モジュール構造を持つモデルでは展開を行い参照節を除去し、参照関係を断ち細分化モデルを生成している。実装においても同様に参照節を除去し、参照関係を断つ必要がある。しかし、ライブラリ呼び出し細分化実装では IMPORTS 等の参照節を残さなければならない。そのためモデルの参照節に合わせて実装の参照節を取り外す。実装がライブラリを参照している場合、構造の違いによりライブラリの参照節は残す。

4.2 変数の対応付け

モデルの変数と実装の変数の対応関係を特定する。実装ではモデルの変数を扱えないため、図 2 のように実装側で変数を定義し、モデルの変数と実装の変数を対応付け使用する。この対応関係をリンク不変条件と呼び、対応関係にある変数をリンク変数と呼ぶ。これらのリンク不変条件を抽出し、リンク変数を特定する。ライブラリを使用している実装の場合、モデルの変数とライブラリの変数を対応付けている場合がある。これらの対応付けも特定する。

4.3 非決定値生成

モデルは抽象的なものであり、ANY 文等の非決定的な値を定め、使用することができる。しかし、実装では、非決定的な値を用いることができない。そのため、実装では非決定的値を決定的な値に置き換え、代入操作を行う。モデル細分化では、ANY 文のモデル操作から非決定値を作成する操作と作成された非決定値を参照して代入する操作の 2 つの操作を作成する。この時、実装でも 2 つの操作に対応した操作を作成する。

4.4 操作抽出

細分化モデルの代入文及び操作呼び出しに対応した操作を抽出する。細分化モデルの代入文、操作呼び出しと等価な操作を抽出する。抽出には Weiser のプログラム

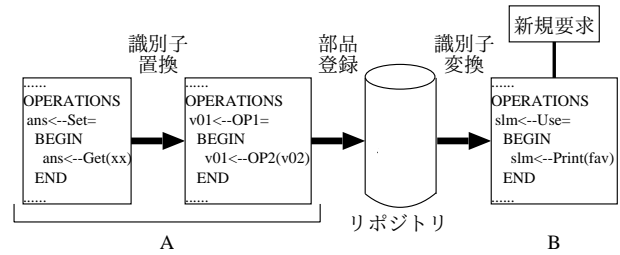


図 4: 識別子置換

スライシング [5] を一部変更して使用する。操作抽出はリンク変数を基準に、抽出した操作に関する制約条件や変数等を抽出していく。細分化モデル内の操作が、引き数も戻り値も無い操作呼び出しである場合、操作のみを抽出する。実装においてライブラリは名前を付けて使用される。例えば図 3 では ARRAY というライブラリに hall と名前をつけて参照している。名前をつけたライブラリの変数においても hall という名前が付く。このようにリンク変数がライブラリ変数の場合、つけられた名前を基準に、制約条件や変数、操作、参照節の抽出を行う。

4.5 識別子置換

生成された細分化実装は識別子が残っており、汎用性に欠けている。そのため図 4 の A のように識別子を全て置換し依存関係を除去する。置換により依存関係を断った細分化モデルと組にしてリポジトリに登録する。登録された部品は、図 4 の B のように、結合時に新規要求に沿って識別子を変換する事で、参照関係を付与し、これらを結合する事でモジュール構造の構築が可能となる。

5 考察

モジュール構造を構築可能な部品と細分化実装の生成手法についての提案した。これら 3 種類の細分化実装を用いた部品であればモジュール構造のソフトウェアを構築することが可能であると考えられる。しかしながら、提案した生成手法は暫定的なものであり、内容について細かい検証が必要である。

6 おわりに

本稿ではモジュール構造を構築可能な細分化実装について生成手法を提案した。今後は提案した手法の有効性確認のため、複雑なモジュール構造を持つモデルと実装を使い、生成手法の検証を行う必要がある。

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] 横山 仁, 織田 健. 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.
- [3] 来間 啓伸. *B* メソッドによる形式仕様記述. 近代科学社, 2007.
- [4] 叶野 英俊, 織田 健. 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.
- [5] M.Weiser. *Program Slicing*. In ICSE, pp.439-449, 1981.