

## 定数値制約領域の拡大による形式的部品の再利用性向上手法

原野和貴<sup>†</sup>

電気通信大学大学院情報学専攻

織田健<sup>‡</sup>

電気通信大学大学院情報学専攻

## 1 はじめに

近年、ソフトウェア開発は複雑化に伴うコストの増大や信頼性の低下が問題視され、部品再利用や形式手法が研究されている。我々は形式手法のBメソッドの仕様と実装の組の部品の再利用による、無矛盾なソフトウェア合成手法を提案している [1]。この手法では、要求と等価な仕様を持つ部品を再利用するが、定数値の不一致による再利用性低下の課題が存在する。本研究では要求や仕様内のスカラー値の定数について、取りうる値の領域の拡大による再利用性向上手法を提案する。

## 2 研究背景

## 2.1 Bメソッド

Bメソッドは数学的基盤に基づくソフトウェア開発を行う形式手法の1つであり、仕様記述からコード生成までを支援する [2]。これは段階的詳細化に基づく仕様から実装への記述が可能であり、機械的な検証による各段階の無矛盾性と詳細化の整合性を保証する。

Bメソッドの定数には識別子を持つ宣言定数と数値が直接記述される数値定数がある。宣言定数は実装で必ず値が与えられ、その定数値は仕様内の制約条件の範囲内で決定される。スカラー値の宣言定数の制約条件は型宣言と最大値等を制限する制約条件、宣言定数間の関係で構成される。本稿では、これらの制約条件による定数値の取りうる値の領域を定数値制約領域と呼ぶ。また、実装への詳細化で新たな宣言定数(実装定数)や宣言定数の制約条件(詳細化情報)が追加されることがある。対して、数値定数は既に定数値を持ち、制約条件を持たない。

## 2.2 モデル充足ソフトウェア合成手法

モデル充足ソフトウェア合成(MSSS)手法はBメソッドの仕様と実装の組の部品を再利用して、要求を満たす無矛盾なソフトウェアを合成する。この手法では、細分化要求と部品の仕様の等価性を文字列一致で判定し、要求を満たす部品が再利用される。また、要求と部品生成時の元のソフトウェアは、事前に文字列統一と細分化をして部品の取得や生成を行う [3]。

## 3 部品の再利用性低下の課題と解決方針

先のMSSS手法の部品取得では、要求と部品の仕様との定数値が異なると部品が再利用されない。これは図1のように、宣言定数の制約条件や数値定数の不一致によって発生する。しかし、これらは定数値以外は一致し定数値の変更による再利用の可能性があり、部品の再利用性が定数値によって低下している。

An Improving Reusability Method of Formal Components by Expansion of Constant Value Constraint Area

<sup>†</sup>Kazuki Harano, The University of Electro-Communications, Graduate School of Informatics

<sup>‡</sup>Takeshi Oda, The University of Electro-Communications, Graduate School of Informatics

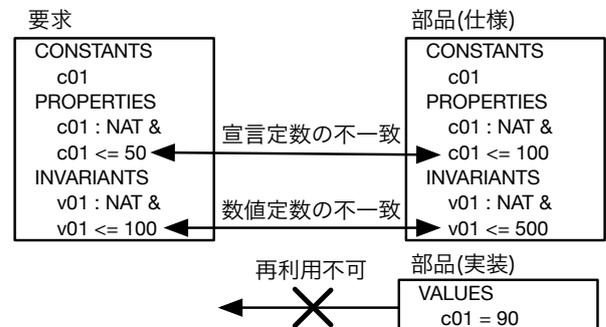


図1: 定数値の不一致による部品の再利用性低下

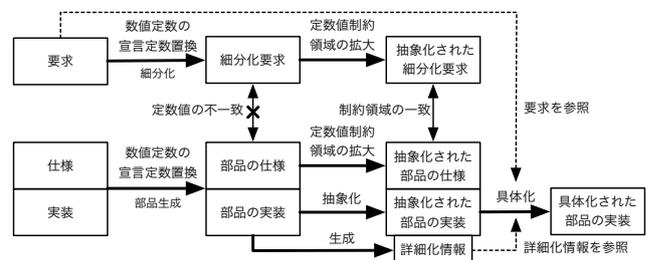


図2: 再利用手法の概要図

この課題に対し、スカラー値の宣言定数の定数値制約領域を無矛盾な範囲に拡大して部品取得し、一致する要求と部品の詳細化情報を元に具体化し再利用する。また、数値定数は宣言定数へ置換しまとめて領域を拡大する。

## 4 定数値制約領域の拡大による再利用手法

本章では、先の方針に従った再利用手法(図2)を説明する。本手法では、事前に数値定数を宣言定数に置換し、宣言定数と同等に取り扱う。その後、要求や部品の仕様で宣言定数の定数値制約領域を拡大し定数値の差を吸収する。また、部品の実装で宣言定数の抽象化と詳細化情報の生成を行い部品生成する。最後に、部品の再利用時に要求と部品の詳細化情報を元に具体化する。以降は手法の各操作を説明する。なお、本手法では宣言定数間の関係は数値を介さずに示されることを前提とする。

## 4.1 数値定数の宣言定数置換

まず、数値定数を宣言定数と同等に扱うために要求や元のソフトウェアを細分化する前に宣言定数に置換する。数値定数は既に定数値を持つ宣言定数と見なし、数値定数を重複しない識別子で置換し識別子と制約条件“識別子 = 数値定数”を持つ宣言定数を追加する。また、仕様内や仕様実装間の等しい値の数値定数群は制約条件や代入文内で共に記述される変数が共通する場合等の間接的に関係する場合に同じ宣言定数としてグループ化する。

## 4.2 定数値制約領域の拡大

数値定数の宣言定数置換と細分化を行った後の細分化要求と部品の仕様に対して、宣言定数の定数値制約領域

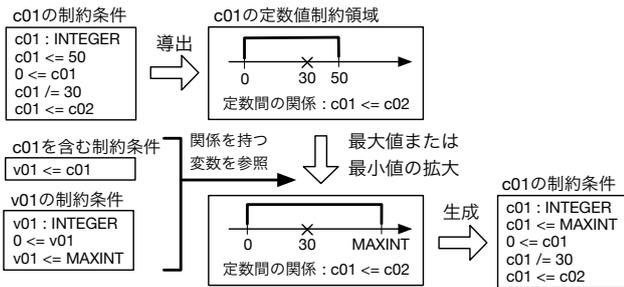


図 3: 定数値制約領域の拡大

を拡大し再利用性を高める。なお、要求については後の具体化が必要となるため元の制約条件を保管する。定数値制約領域は‘型’、‘最大値’、‘最小値’、‘禁止値’、‘定数間の関係’で構成される。この操作では、各宣言定数について制約条件から定数値制約領域の導出、最大値または最小値の拡大、制約条件の生成の3操作を行う(図3)。

#### 4.2.1 定数値制約領域の導出

宣言定数の制約条件から定数値制約領域の導出を行う。定数値制約領域の各情報は‘ $\in$ ’や‘ $\leq$ ’の制約条件や定数間の関係を示す制約条件等、それぞれに対応する制約条件から得る。なお、最大値と最小値は得られる領域が狭くなるように選択する。また、禁止値は元の記述での値のみとする。

#### 4.2.2 最大値または最小値の拡大

得られた定数値制約領域の最大値または最小値を境界値に拡大する。最終的な制約条件は元の記述によらずに統一する必要があり、対称の宣言定数が使われた制約条件や代入文内で共に記述される変数を基準とし拡大後の境界値を決定する。なお、境界値は0、1、MAXINT、MININTとする。例えば図3では、宣言定数  $c01$  と共に記述される  $v01$  を基に最大値を拡大している。

#### 4.2.3 制約条件の生成

拡大した定数値制約領域から導出の操作とはルールが同じな逆の変換をして制約条件を生成する。

#### 4.3 実装での宣言定数の抽象化と詳細化情報の生成

部品生成で得られた実装は、再利用時に新たな定数値を与える必要があるため事前に宣言定数の抽象化を行う。また、再利用時に矛盾の発生しない具体化を行うために予め詳細化情報を生成し部品の付加情報とする。以降はこれらの各操作について説明する。

##### 4.3.1 宣言定数の抽象化

宣言定数の抽象化では、実装内の値を変更する宣言定数の元の値を削除する。抽象化対称の宣言定数は仕様内の宣言定数と後述する一部の実装定数の2種である。元の値の削除は対称の宣言定数へ値を与える代入文の数値を統一の記号(‘?’)に置換する。

実装定数は、仕様内の宣言定数や変数の詳細化により追加される仕様依存のものや実装内のアルゴリズムに依存するものの2種類が存在する。ソフトウェア合成手法は新たな要求に対して実装を対応させる必要があるため、本手法では前者の実装定数の値を変更する。実装内で仕

様依存の実装定数は必ずいずれかの仕様内の宣言定数や変数との間の関係を示す制約条件が存在する。そのため、実装定数がこの制約条件を介して仕様内の宣言定数や変数との関係が特定できる場合は、仕様依存であり値を変更する。対して特定ができない場合は、アルゴリズム依存であり値を変更しない。

##### 4.3.2 詳細化情報の生成

詳細化情報の生成では、仕様内の宣言定数について実装で新たに制約条件が追加された場合に部品の付加情報として保管する。詳細化情報は主に新たな最大値や最小値の制約条件または宣言定数間の関係であり、型は詳細化されない。これらは明示的に示される場合と変数の制約条件内など暗黙的に示される場合がある。前者の場合は、示された制約条件を詳細化情報とするが、多くは暗黙的に示される。暗黙的な場合の詳細化情報の生成は実装内の宣言定数の記述から生成する。例えば、仕様内で互いに関係のない宣言定数が実装の変数の制約条件内で“ $c01 \dots c02$ ”と区間集合で示される場合はこれらの定数間の関係“ $c01 \leq c02$ ”を詳細化情報とする。

#### 4.4 部品内の詳細化を考慮した具体化

取得した部品を結合した合成実装に対して、抽象化を行った宣言定数の統一の記号を定数値で置換し具体化する。仕様内の宣言定数の新たな定数値は、先の操作で保管した要求内の制約条件と部品の詳細化情報の論理積内で取りうる値をユーザにより決定する。対して、実装定数はユーザの考慮しない定数のため上の具体化の情報はなく、仕様や実装内の実装定数と関連する全ての式を提示し値を決定する。

### 5 実験と考察

各操作の可能なものについて、小規模なソフトウェアや制約条件を用いた簡易的な実験によって妥当性があることを確認した。定数値制約領域の拡大は制約条件の元の記述によらない定数値の差の吸収が可能である。また、具体化時に要求と部品の詳細化情報を考慮することで、再利用後の合成実装は要求を満たし無矛盾性を保証する。しかし、数値定数のグループ化の詳細な基準や詳細化情報を考慮した部品取得等の考察が課題である。また、これらの手法を統一のソフトウェアを用いた実験による妥当性や健全性の検証を行う必要がある。

### 6 終わりに

本稿では、定数値が不一致な部品の宣言定数の定数値制約領域の拡大による再利用手法を提案した。今後は、上記の課題への対応や手法の検証等を行いたい。

### 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 電気通信学研究所 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] 三鍋 孝介, 織田 健, 「文字列一致による数学的等価判定可能なモデル分割アルゴリズム」, 第12回情報科学技術フォーラム論文集, vol.1 pp.271-272,(2013.09)