B-009

# 定数宣言の抽象化による値不一致な形式的部品の再利用手法
# A Method for Reusing Formal Components with Inconsistent Values by Abstracting Constant Declarations

原野 和貴† 織田 健†
Kazuki Harano Takeshi Oda

## 1 はじめに

近年、ソフトウェアの複雑化に伴う開発コストの増大や信頼性の低下が問題であり、部品再利用や形式手法が研究されている。形式手法の 1 つに B メソッドがあり、我々は B メソッドで記述されたモデルと実装の組の部品を再利用した、要求モデルを満たし無矛盾な合成実装の自動合成手法を提案している [1]。この手法は要求モデルを満たす部品をモデルの等価性を判定して取得するが、定数値が異なると部品が再利用できない課題が存在する。本研究では、識別子を持つスカラー値の定数を抽象化し等価性を判定して部品を再利用する手法を提案する。

## 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形式手法の 1 つで、仕様記述からコード生成までを支援する [2]。これは仕様に対応するモデルから実装への段階的詳細化に基づき、各段階の無矛盾性と詳細化の整合性を機械的に検証する。

B メソッドにおける定数は識別子を与え宣言される定数 (宣言定数) とコード内に直接存在する数値がある。宣言定数は実装で必ず定数値が与えられ、定数値を制限する制約条件を持つ。スカラー値の宣言定数の制約条件は必ず型宣言を持ち、上限や下限の制限や宣言定数間の関係を示す制約などを持つ。また、実装のみで宣言される定数も存在し、同様に制約条件と具体値が与えられる。

### 2.2 モデル充足ソフトウェア合成手法 (MSSS 手法)

モデル充足ソフトウェア合成 (MSSS) 手法は B メソッドで記述された既存ソフトウェアから生成されるモデルと実装の組を部品とするソフトウェア自動合成手法である [1]。この手法では、要求モデルと部品のモデルの等価性を文字列一致により判定して要求モデルを満たす部品の再利用を行う。また、要求モデルや部品のモデルには事前に推論などによる文字列の統一と操作分割や分割された操作の制約条件の抽出による細分化が行われる。

## 3 課題と解決方針

### 3.1 定数値の不一致による形式的部品の再利用性低下

MSSS 手法では、要求モデルに対し、部品のモデル内の定数値が異なると等価でなく部品は再利用されない。例えば、図 1 に文字列統一と細分化が行われた、省略部分が同じ要求モデルと部品を示す。この時、宣言定数 c01 はその制約条件が異なり、定数値が不一致なため部品は再利用されない。しかし、これは定数値が異なる以外は一致するモデルで、定数値の変更で再利用できる可能性がある。よって、部品の再利用性が低下している。

†電気通信大学大学院情報理工学研究科情報学専攻

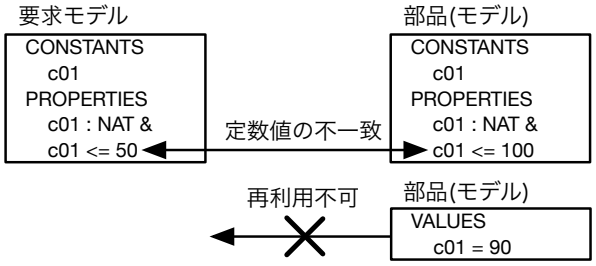


図 1: 定数値の不一致による再利用性低下

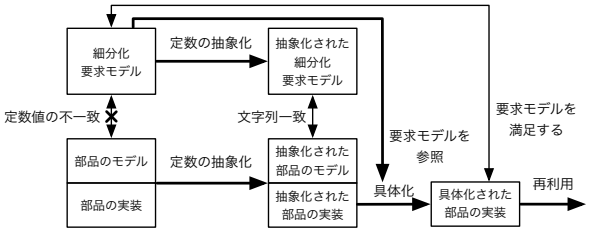


図 2: 宣言定数の抽象化による再利用性向上の概念図

### 3.2 課題解決の方針

上記の課題に対して、宣言定数を抽象化して等価性を判定し、要求に合わせて具体化することで定数値が不一致な部品の再利用を行う。この方針を図 2 に示す。

## 4 宣言定数の抽象化による再利用手法

本章では、上記の方針に従い、スカラー値の宣言定数の抽象化と具体化によって定数値が不一致な部品の再利用手法を説明する。宣言定数の抽象化は、モデルの宣言定数の差の吸収と実装の宣言定数の値の削除を目的とする。これを要求モデルと部品に同様の処理を行い定数値が不一致な部品を再利用可能とする。宣言定数の具体化では、部品の結合後に抽象化された定数値を要求モデルに合わせて具体化する。以下、この再利用手法について、宣言定数の抽象化と具体化の手法、実装で宣言される定数 (実装定数) に対する手法の 3 つに分けて説明する。

なお、再利用の健全性のために、本手法では宣言定数間に関係がある場合、その関係を数値を介さずに c01 <= c02 などと、直接示されていることを前提とする。

### 4.1 宣言定数の抽象化

宣言定数の抽象化は、モデルと実装によって手法が異なり、これらを分けて説明する。

#### 4.1.1 モデルでの宣言定数の抽象化

モデルでの抽象化では、宣言定数の数値を含む制約条件の表記を統一し、後述する特定の制約条件を削除する。この制約条件の削除により、モデル間の宣言定数の差を吸収する。また、要求モデルの宣言定数については、具体化に用いるため削除する制約条件は保管しておく。

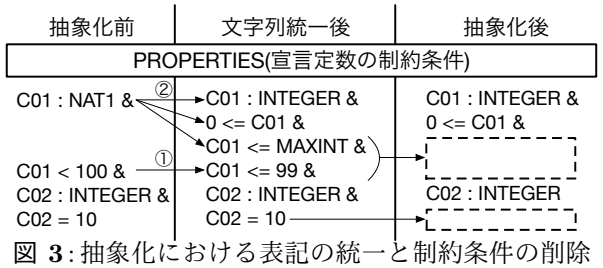


**図 3**: 抽象化における表記の統一と制約条件の削除

**宣言定数の制約条件の表記の統一**

まず、等価判定時の文字列の統一のため、制約条件の削除前に制約条件の表記の統一を行う (図 3 左)。

主な処理は演算子と型宣言に対して行う。演算子では、大小比較演算子を 1 種類 (‘<=’) に統一する (時に ‘/=’ も含む)(図 3 ①)。型宣言では、型への所属述語において、複数の種類の型を最も原始的な型へと統一する (図 3 ②)。

**削除対象とする宣言定数の制約条件**

上記により統一化された制約条件に対し、「1 つの宣言定数と数値のみで構成される比較 (‘<=’) または等値 (‘=’) の制約条件」を削除する (図 3 右)。また、制約条件に 2 つ以上の宣言定数が含まれる場合は、宣言定数間の関係と見なし数値が含まれていても削除しない。

しかし、削除対象の制約条件の中で、0 や 1 の境界値を含む制約条件は、具体化の際に矛盾の生じる値の決定を回避するために削除しない。もし、この削除しない制約条件が複数存在し、それらが包含関係で省略可能であれば、制約条件をまとめる。

#### 4.1.2 実装での宣言定数の抽象化

実装での抽象化では、部品生成で得られた実装に対し、値の変更の必要のある宣言定数を抽象化する。値の変更のある宣言定数は、モデルの宣言定数と、後述する特定の実装定数である。これらの実装内の宣言定数に具体値を与える代入文の数値を統一の記号 (‘?’) に置換する。

### 4.2 宣言定数の具体化

宣言定数の具体化では、取得した部品を結合した合成実装に対して、抽象化された宣言定数に統一の記号を置換して具体値を与える。与える具体値は、モデルの宣言定数では推論後の要求モデル内の制約条件を元に値を決定する。この時、制約条件に等値演算子が存在する場合はその制約条件内の数値を具体値とし、存在しない場合は制約条件内で取りうる値をユーザにより決定する。値の変更の必要のある実装定数の具体化は次節で説明する。

### 4.3 実装定数の抽象化と具体化

ここでは、実装のみで宣言される宣言定数 (実装定数) について、値の変更の必要がある実装定数について述べ、その抽象化と具体化の手法を説明する。

実装定数は、ユーザが考慮していない定数が部品の実装で新しく宣言されるため、通常はアルゴリズムに依存する定数と見なし変更しない。しかし、実装定数がモデル内の宣言定数や変数と関係を持つ時、部品の再利用時はモデル内の定数や変数は元とは異なるため、関係する実装定数も値の変更を行う。ここで、この実装定数をモデル依存の実装定数とし、抽象化と具体化を行う。

モデル依存の実装定数の抽象化は、4.1.2 項の処理でモデル依存の実装定数に対し数値を統一の記号で置換する。実装定数は、それがモデル内の宣言定数と関係が存在する場合、またはモデル内の変数の詳細化時に実装定数が使用される場合にモデル依存と判定する。前者の場合は、実装定数の制約条件内でモデル内の宣言定数が含まれる時にその宣言定数に対してモデル依存と判定する。後者の場合は、実装内の変数は新たに変数が導入されずモデル内の変数の詳細化のみのため、変数の制約条件内で実装定数が含まれると、その実装内の変数とその詳細化前のモデル内の変数に対してモデル依存と判定する。よって、上記以外の実装定数は値を変更しない。

さらに、モデル依存の実装定数の具体化は、4.2 節と同様に制約条件を元に具体値を決定する。しかし、実装定数はユーザが考慮していないため、モデル内の実装定数と関係を持つ定数や変数の制約条件と実装内で実装定数が含まれる全ての式を提示し、値を決定する。

## 5 評価実験

提案した再利用手法の妥当性と一部の健全性を検証する目的で実験を行った。実験では、定数値が異なる要求モデルと既存ソフトウェアで構成される実験モデルを用いたソフトウェア合成実験を行った。実験の結果として、定数値を吸収した部品の取得と結合により合成実装を作成し、無矛盾性と要求モデルとの整合性が証明された。

## 6 考察

本手法の妥当性は、定数値が異なる部品が再利用できることで示される。これは評価実験により、定数の差を吸収した部品の取得と結合が確認され、示された。

また、本手法の健全性はこれを導入して合成された実装が要求モデルを満たし、無矛盾であれば保証される。本実験の実験モデルでは独立な宣言定数などの単純な例において、健全性が示された。しかし、宣言定数間の関係や実装定数、制約条件を残すことの健全性の保証など、検証が足りない点が存在し、現在も実験を継続中である。

最後に、4 章冒頭で述べた前提条件について述べる。B メソッドでは、`c01 <= 1024 & 1024 <= c02` と記述しても証明上では `c01` と `c02` が暗黙的に大小関係を持つと判断され証明される。しかし、4.1.1 項の制約条件の削除で暗黙的な大小関係が消え、再利用時に矛盾が発生する可能性がある。本来、数値 `1024` は同じ値の数値が存在しても同じ `1024` と見なすことは不適切であり、本手法ではこれらから暗黙的な関係を導出しない。したがって、前述の前提条件が満たされることが必要である。

## 7 終わりに

本研究では、スカラー値の宣言定数を抽象化して再利用性を向上させる手法を提案した。今後は、さらなる健全性の検証や数値定数への対応が課題である。

## 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

[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