

B-027

## Java プログラムからの再利用可能コンポーネントの自動抽出システム A System for Automatic Reusable Component Extraction from Java Program

轉法輪 徳子\*、酒井 三四郎\*\*

Noriko Temporin, Sanshiro Sakai

### 1. はじめに

コンポーネントベース開発は、コンポーネントを組み立ててシステムを開発する開発手法であり、費用対効果に優れているとして近年注目されている。そのためコンポーネントは、汎用的な機能を実現し、品質・信頼性が高いなどの条件を満たしている再利用性の高いソフトウェア部品であることが望まれる。

コンポーネントを作成する手法の1つとして、既存のプログラムからコンポーネントと成り得る部分を抽出するという方法がある。鷲崎らは“Extract component”というリファクタリングを行うことで、Java プログラムからコンポーネントを抽出した[1]。その結果、抽出したコンポーネントの約2/3について再利用性が高いと評価されている。ここで、彼らは機械的にコンポーネント候補となる部分を抽出し、コンポーネントへ変形させている。このように、機械的にコンポーネント抽出を行ったとき、必ずしも得られたコンポーネントが再利用性の高いものとはなりえない。

そこで、このプロセスにおいて機械的に候補部分を取り出すだけでなく、候補部分が有用なコンポーネントと成り得るかどうかという判定を加え、提示することを提案する。判定結果を提示することで、再利用性の高くなりうるコンポーネントに絞って抽出を行うことが可能となり、結果として再利用性の高いコンポーネントを抽出できる割合が上がるのではないかと考えている。また、判定結果を提示することでユーザによるコンポーネント候補の選択が容易になることを期待する。判定には、オブジェクト構造に基づく判定と評価基準マトリクス予測に基づく判定の2つを用いた。

### 2. Extract Component Refactoring

まずクラス/インタフェースの関係を表す CRG を定義する。次に、再利用可能なコンポーネントについて定義する。さらに、コンポーネント抽出プロセスについて定義する。

#### CRG(Class Relation Graph: クラス関係グラフ)

既存の Java プログラムを静的解析することでクラス間の継承・具体化・参照関係を示したグラフを作成する。このグラフ上で到達可能な範囲を判定することでコンポーネント化すべき領域を特定する。

#### 定義(再利用可能なコンポーネント)

再利用可能なコンポーネントは以下の条件をすべて満たし、JAR ファイルにパッケージされている Java クラス/インタフェースのセットである。

- (1) JavaBeans 部品として取り扱うことができる。
  - 1、Façade クラスをもつ。
  - 2、Façade クラスはデフォルトコンストラクタをもつ。

3、Façade クラスは `java.io.Serializable` インタフェースを実装する。

4、Façade クラスについて述べたマニフェストファイルを含む1つの JAR ファイルとしてパッケージされている。

- (2) Façade インタフェース宣言はその実装から分離される。

#### Extract component refactoring

コンポーネント抽出には、鷲崎らのリファクタリングプロセス[1]を参考にする。クラスタ獲得後に、それぞれのクラスタの有用性の判定プロセスを加える。

- (1) 対象とする Java プログラムの CRG  $\Gamma$  を作る。
- (2) 抽出基準として、開始ノードを選ぶ。
- (3) クラスタリングアルゴリズムに  $\Gamma$  を適用して、すべてのクラスタ  $S$  を獲得する。
- (4) 獲得したクラスタ  $S$  が有用なクラスタであるかどうかを判定し、それぞれに対する判定結果を提示する。
- (5) 判定結果を参考にして、クラスタ  $S$  から1つのクラスタ  $cs$  を選ぶ。(  $cs$  の開始ノードを Façade クラス  $cf$  とする。)
- (6) Façade インタフェース  $if$  を作る。  $cf$  を更新する。
- (7) Java コンパイラを使用して、  $cf$  と新たに作成した  $if$  のソースコードをコンパイルする。
- (8)  $cf$  のマニフェストファイルを作成し、  $cs$  と  $if$  におけるクラス/インタフェースを1つの JAR ファイルとしてパッケージする。コンポーネントの名前は  $cf$  とする。

### 3. オブジェクト構造による判定

#### 3.1. 有用性の判定

文献[2]中にあるコンポーネント化すべきか否かの条件判断を参考に CRG に適用する。CRG に適用する条件は、オブジェクト構造におけるパリエーションを生む、可変多重度、再帰、サブクラス化(図1を参照)の要素が入っていることとする。いずれかの要素を含んでいる時、構造に基づく判定(判定)を満たしていると判定する。

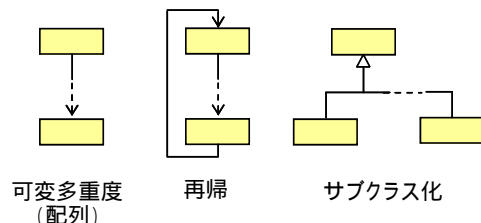


図1 有用性判定の条件

#### 3.2. 評価方法

抽出したコンポーネントに対するマトリクスを計測する。その値を基準として再利用性が高いと評価されるものと、構造における判定で有用であると判定されたコンポー

\*静岡大学大学院情報学研究科

\*\*静岡大学情報学部

ネットを比較し評価する。

用いるメトリクスには DIT と SCCr を使う。DIT は継承度の深さを示したメトリクスであり、ここでの信頼区間（[下限、上限]）を [2, 4] とする。SCCr は自己完結性を示したメトリクスであり、全体のメソッドにおける返り値を持たないメソッドの割合を求める（ただし、セッターメソッド・ゲッターメソッドは除く）。ここでの信頼区間は [0.61, 1.0] とする。DIT と SCCr の少なくとも一方が信頼区間にあるとき、そのコンポーネントは 95% の確率で再利用性が高いと判断する [3]。

適用するプログラムは、j2sdk の demo を始めとするサンプルプログラムを評価対象プログラムとして使用した。13 個のプログラムに対して全てのクラスを開始ノードとして使用し、可能な限りのコンポーネントを抽出した。その結果 89 個のコンポーネントが抽出された。上記の判定基準より、この中の 72 個のコンポーネントは再利用性が高いと評価されている。

### 3.3. 実験結果

実験結果を表 1 に示す。「判定」は構造に基づく判定を満たしているコンポーネントについてのそれぞれの割合を示している。「N.All」は得られたコンポーネント数を、「N.D S」は得られたコンポーネントの中で再利用性が高いと評価されたコンポーネントの数を示している。

表 1 実験結果（構造に基づく判定）

	N.All	N.D S	適合率	再現率
判定なし	89	72	0.81	1.00
判定	33	29	0.88	0.40

表 1 より、有用であると評価されたコンポーネントの約 9 割が再利用性が高いと評価された。しかし、全体の 4 割にみえないコンポーネント数のみしか構造に基づく判定をみだしていなかった。よって、構造に基づく判定を用いることで再利用性の高いコンポーネントの抽出が可能となるが、この判定基準のみではコンポーネントの再現率を大幅に下げってしまう結果となった。

## 4. 評価基準メトリクス予測に基づく判定

### 4.1. 有用性の判定

再利用性の評価基準である DIT、SCCr をあらかじめ予測することで、再利用性の高いコンポーネントの抽出を可能とする。

プログラムの解析を行うことで DIT、SCCr の予測値をあらかじめ求めることが可能である。DIT は継承の数を、SCCr は返り値を持たないメソッドの数を実際に数え上げることで予測することができる。予測値をそれぞれ preDIT、preSCCr とし、それぞれの結果に基づいて 3 段階（再利用性が高い（）、分からない（）、再利用性が低い（×））の総合評価を表 2 にしたがって行う。この総合評価をメトリクス予測に基づく判定（判定 ）の基準として用いる。

表 2 総合評価

preDIT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
preSCCr	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
総合評価	<input type="checkbox"/>		

### 4.2. 評価方法

メトリクス予測に基づく判定における評価と、DIT と SCCr による実際の評価で再利用性が高いと評価されたコンポーネントの個数を比較することで評価を行う。再利用性を計るメトリクス、サンプルプログラムは構造に基づく判定の評価実験を行ったものと同じものを用いる。

### 4.3. 実験結果

実験結果を表 3 に示す。「判定」は表 2 の総合評価によるメトリクス予測に基づく判定を、「判定」の評価結果は、メトリクス予測に基づく判定でのそれぞれの評価（、×）に位置するコンポーネントの数を示している。

表 3 実験結果（メトリクスに基づく判定）

判定	×		
判定 の評価結果	12	9	68
N.D S	0	4	68

再利用性が高い（）と判定したコンポーネントの 68 個すべてが、DIT と SCCr の評価基準においても再利用性が高いと評価された。分からない（）と判定した 9 個中 4 個が再利用性が高いものであった。再利用性が低い（×）と判定したものの 12 個すべてが再利用性が高いとは評価されなかった。

以上の結果より、メトリクス予測に基づく判定を判定基準として示すことで、有用なコンポーネントとなりうるコンポーネントの抽出を行うことが可能になると考える。

## 5. おわりに

メトリクス予測に基づく判定により、コンポーネントの有用性をコンポーネント変換前に示すことで、再利用性の高いコンポーネント候補を選択し、抽出させることが可能となった。また構造に基づく判定により、構造的に有用なコンポーネントの抽出が可能である。よって、構造に基づく判定とメトリクス予測に基づく判定を併用することでより再利用性の高いコンポーネントの抽出が可能になったと言える。

今後の課題としては、構造における判定やメトリクス予測に基づく判定で用いられた評価基準以外に再利用性が図れる基準があれば、それをさらに加えることでより再利用性の高いコンポーネントの抽出が可能となるのではないかと考える。

### 参考文献

- [1]H.Washizaki, Y.Fukazawa.; A technique for automatic component extraction from object-oriented programs by refactoring, Science of Computer Programming, pp.99-116 (2005).
- [2]吉田和樹, 本位田真一: コンポーネントベース・フレームワーク技術におけるコンポーネントの抽出/設計方法論, 情報処理学会研究報告ソフトウェア工学, Vol.2000, No.70, pp.25-35 (2000)
- [3]H.Washizaki, H.Yamamoto, Y.Fukazawa, A metrics suite for measuring reusability of software components, in: Proc.9th IEEE International Symposium on Software Metrics, pp.211-223 (2003)