

メソッドと変数間結合力の可視化によるリファクタリング支援

長谷川 智和[†] 澤田 敬太[†] 大木 幹雄[†]

日本工業大学[†]

1. はじめに

リファクタリングを有効に行うには経験が必要とされ、プログラミング初心者にとってリファクタリングを行うことは容易ではない。本稿では、サブクラス化によりリファクタリングを簡易に行えるようにした支援ツールを試作し、その有効性を評価した結果について述べる。

2. システム概要

試作ツールはC++プログラムを対象にしたもので、リファクタリングのサブクラス抽出を行うためのデータを収集し、可視化するものである。データの収集と可視化の基本的な考え方は、以下にしたがって行う。

(1) メソッドと変数の結合力の定義

メソッドにアクセスする変数との間に以下の結合力が働くものとする。ここでメソッドを m 、変数を v 、アクセス係数を α 、多次元空間での m と v を配置したときの距離を r とし、(式1)で結合力 F を定義する。

$$F = \frac{\alpha \cdot mv}{r} \dots (\text{式1})$$

(2) 多次元空間上での変数の移動

変数の結合力を移動量とし、メソッド数の多次元空間上で移動をあらかじめ定めた回数行う。各メソッドはメソッド数の次元を持つ空間の各座標の定数位置に固定して配置する。変数の移動が安定した状態でメソッドと変数の距離を例えば表1で示すような距離表として保持する。

表1のセル内の数値は各メソッドに対する変数の距離である。メソッドAに対して変数Aは15の距離を持つことを示している。

表1 変数のメソッド間距離表の例

	変数A	変数B	変数C	...
メソッドA	15	80	50	...
メソッドB	80	10	80	...
メソッドC	80	40	10	...
:	:	:	:	...

(3) メソッドと変数のクラスタリング

距離表の変数の列から最小の距離を取り出し、その対応するメソッドを最小距離群とする。表1ではを含むセルを最小距離群として対象のメソッドを保持する。最小距離群の中のメソッドが同じ変数が

ら分散を求める。算出された分散が任意の基準より低いものだけを多次元空間上でクラスタリングすると、図2のように表現できる。

(4) クラスタを用いたサブクラス化

クラスタの組み合わせを総当たりで行い、各組み合わせでクラスタ内にある変数の距離の分散を求める。この分散で基準値より低いものをサブクラス生成候補とする。これを図示すると図1のようになる。

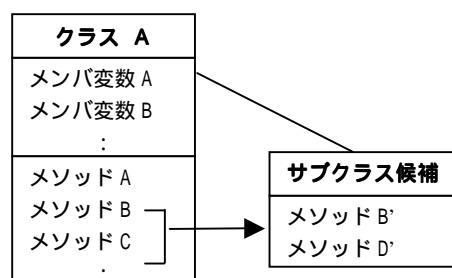


図1 サブクラス化後の抽出したクラス図

3. 機能概要

ツールには次の機能を実装した。

(1) ソースコード解析機能

結合力を定義する際に参照と書込みで重み付けを変えるため、参照回数と書込み回数を分けてカウントする。

図3は、ファイルの読み込みと解析結果の出力である。図3(a)で示したダイアログで、ファイルの読み込みと解析の指示を行う。対象のソースコード、クラス名、各メソッドで使われている変数のアクセス状況が全て表示される。図3(b)で示したダイアログで、変数毎に各変数がアクセスされているメソッドとそのメソッドでアクセスされている変数を表示する。解析後、メソッドと変数間の距離を算出する。

(2) 二次元空間上でのメソッドと変数間結合力の可視化機能

図4で示すとおり、メソッドを同一円周上に配置し、円の中心から変数の移動を連続的に行いアニメーションとして表示する。二次元空間上での移動距離 d は次の(式2)で求める。

$$dr = \int Fds \dots (\text{式2})$$

F : 結合力 ds : ステップ

(3) 二次元空間上でのクラスタによるサブクラス生成の提案機能

アニメーションを行った後、図4で示すようにメソッドと変数の結合力よりクラスタを作成し、クラスタ部分を囲み、サブクラス抽出に適しているかどうかを判断し、サブクラス生成に適した複数のクラスタを囲む。

A Refactoring Assist Tool by the visualization of method - variable combining force

[†] Tomokazu Hasegawa, Keita Sawada, Mikio Ohki

Nippon Institute of Technology

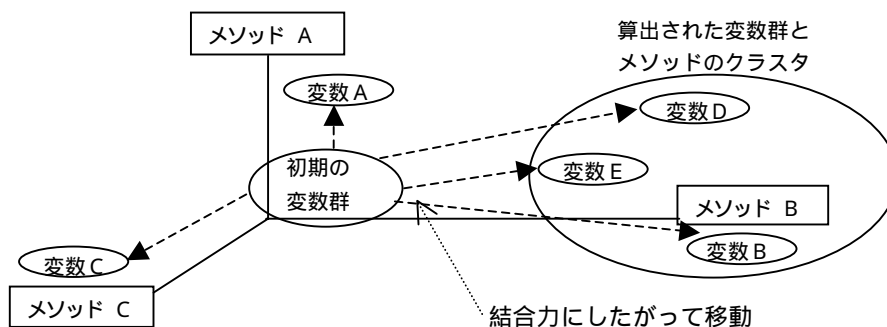


図2 多次元空間に変数とメソッドを配置したときのイメージ

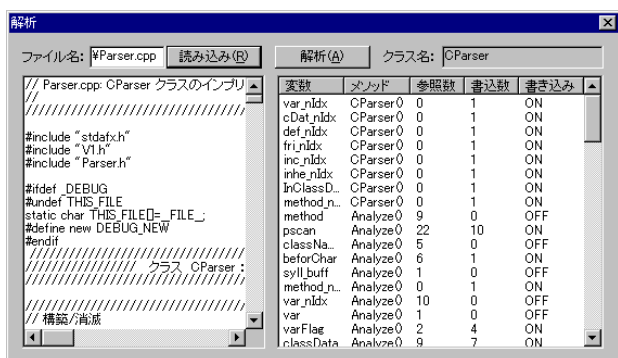


図3 (a) 読み込み、解析指示画面

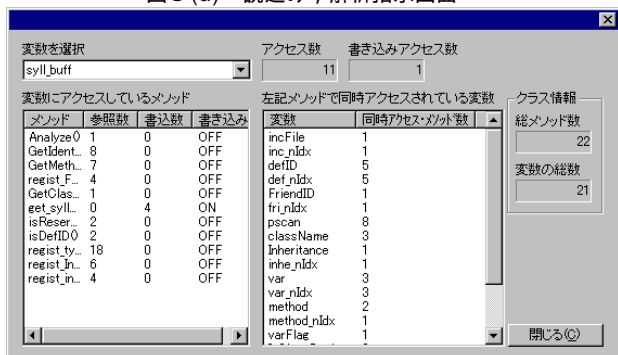


図3 (b) メソッド - 変数のアクセス回数表示例

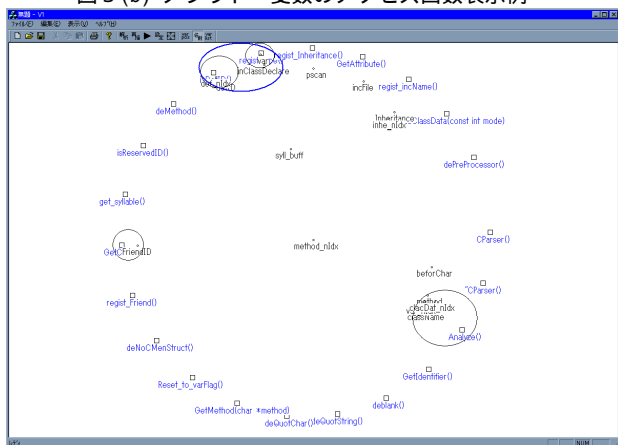


図4 変数とメソッドのアクセス状況アニメーション出力例

4. 特徴

- (1) 保守・管理に有効なサブクラスの生成
結合力の強い変数群とメソッド群で構成されたサブクラスは、相互に影響を与えている集合なので保守・管理に有効である。
- (2) 変数とメソッドの関係の可視化。
変数とメソッドの結合力から移動量を導き出しているため、アニメーションではそのクラス中のメソッドと変数の関係が容易に理解できる。

5. 考察

- 本ツールを利用してクラス抽出の実験を行った。クラスの抽出を試みた対象のプログラムは、ツール自体のソースコード、プログラム設計演習教材のソースコードを使用した。
実験結果より、次のことが判明した。
- (1) アニメーション画面により変数とメソッドの論理的な結合状況が形成される過程をビジュアル的に把握した結果、最終的な変数の配置に偏りができる。
- (2) 大きいクラスに対しては、クラスタリングを行いサブクラスの抽出をすることで、クラスの規模を押さえ、保守・管理に有効なサブクラスを生成することができる。

6. おわりに

本研究では、リファクタリングの支援ツールを試作し、サブクラスの生成に必要なデータ収集の一手法の提案を行った。数値的な側面からみた方法で、保守・管理に有効なサブクラス生成を容易に行うことができた。

参考文献

- [1] Martin Fowler, リファクタリングプログラミングの体質改善テクニック, ピアソン・エデュケーション社, 2000
- [2] Mark Lorenz, Jeff Kidd, オブジェクト指向ソフトウェアメトリックス, プレンティスホール出版, 1995