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

大木研究室　長谷川 智和　澤田 敬太

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考える．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．

$$F = \frac{-\alpha mv}{r} \quad \cdots（式１）$$

（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距離表として保持する．

（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同士での変数から分散を求める．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

（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．

3.　機能概要

（１）ソースコード解析機能

結合力を定義するため，参照回数と書込み回数を分けてカウントする．

（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１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ｄｒは次の（式２）で求める．

$$dr = \int Fds \quad \cdots（式２）$$

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，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．

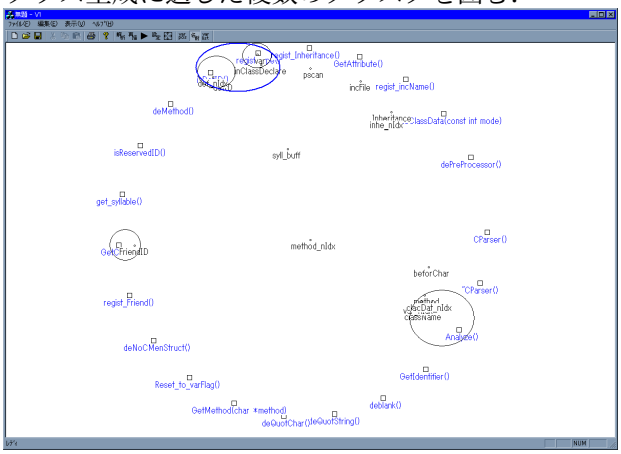

図１ 変数とメソッドのアクセス状況アニメーション出力例

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ので，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

（１）アニメーション画面により変数とメソッドの結合状況が形成された結果，変数に配置の偏りがビジュアル的に把握できる．

（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