C&Kメトリクスを利用したリファクタリング支援ツール

2K−2

秦野 克彦†, 乃村 能成††, 谷口 秀夫††, 牛島 和夫†††
†九州大学大学院システム情報科学府
††九州大学大学院システム情報科学研究院
†††九州システム情報技術研究所

1 はじめに

ソフトウェアは要求に合わせて機能変更や機能拡張され、改版されていく。この結果、構造設計当初のプログラム構造の統一性は崩れることが多く、機能変更や機能拡張の工数も増加してしまう。そこでリファクタリング[1]が有効である。リファクタリングとは、ソフトウェアが提供する機能を変更することなく、プログラムの内部構造を変更することである。このため、既存のプログラムにリファクタリングをうまく適用することで、機能変更や機能拡張の工数を少なくすることが期待できる。

リファクタリングを行うためには、機能変更や機能拡張の工数の増加を招くプログラムの構造上の不具合箇所を検出する必要がある。さらに、不具合箇所を改善する適切なリファクタリング手法を選択し施す必要がある。従来、こうした検出や選択は難しく、作業者の知識や経験を必要とした。

本稿では、リファクタリングの作業を支援するツールについて述べる。このツールはC&Kメトリクス[2]を利用して、リファクタリングを行うべき不具合箇所を自動的に検出する。そして、不具合箇所に応じたリファクタリング手法を提示して、適用するべきリファクタリング手法の選択を支援する。このツールを用いることで、作業者の経験や知識に関係なく、プログラムを改善することが可能になる。

2 ソフトウェアメトリクスを利用したリファクタリングの機構

図1に、ソフトウェアメトリクスを利用したリファクタリングの機構を示す。この機構は、プログラムの構造上の不具合箇所を検出するために、「改善要否判定基準」を用いる。そして、検出した不具合箇所を改善するリファクタリング手法を選択するために、「ソフトウェアメトリクスとリファクタリング手法の相関」を用いる。

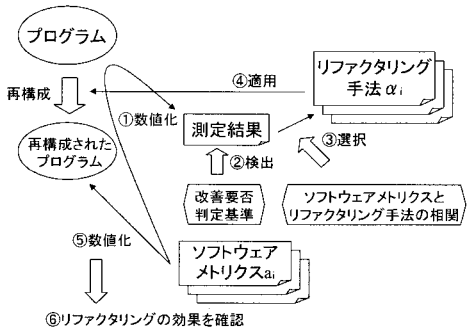

図1 ソフトウェアメトリクスを利用したリファクタリング機構

具体的には、「改善要否判定基準」としてしきい値を定めておく。しきい値より測定値が大きいか小さいかで不具合箇所がどうか判定でき、不具合箇所を自動的に検出できる。また、「ソフトウェアメトリクスとリファクタリング手法の相関」としてリファクタリング手法の適用で測定値が、減少、増加、不変のいずれか調べておく。測定値の変化とリファクタリング手法との関係がわかるため、不具合箇所を改善するリファクタリング手法を自動的に選択できる。

3 リファクタリング支援ツール

3.1 概要

本ツールは、図1の機構のうち、(1)(5)数値化、(2)検出、および(3)選択に関わる機能を実現したものである。ソフトウェアメトリクスとしてC&Kメトリクス、リファクタリング手法としてFowlerのリファクタリング手法[3]を用いている。改善要否判定基準としてC&Kメトリクスのしきい値を定めた。また、ソフトウェアメトリクスとリファクタリング手法の相関として、C&Kメトリクスの観点とFowlerのリファクタリング手法の内容とを調べた。

本ツールは、プログラミング言語Object Pascalで書かれたプログラムを入力データとする。そして、C&Kメトリクスを利用してプログラムを数値化し、測定結

---

a Tool to Support Refactoring Using C&K metrics
Katsuhiko HATANO, Yoshinari NOMURA, Hideo TANIGUCHI , and Kazuo USHIJIMA
Graduate School of Information Science and Electrical Engineering, Kyushu University
Institute of Systems & Information Technologies / KYUSHU
E-mail: hatano,nom,tani@swlab.csce.kyushu-u.ac.jp, ushijima@isit.or.jp

果を画面上に出力する。その際同時に、改善要否判定基準を用いて不具合箇所を検出する。C&K メトリクスによる測定値は値が大きいほど、改善を必要とすることを意味する。そのため、定めたしきい値よりも大きな測定値を取る箇所を不具合箇所とみなしている。さらに、ソフトウェアメトリクスとリファクタリング手法の相関を用いて、Fowler のリファクタリング手法の中から適用するリファクタリング手法を提示する。

3.2 不具合箇所の検出機能

本ツールは、改善要否判定基準を用いて不具合箇所を検出する機能を持つ。この機能の実行結果を図 2 に示す。図 2 はプログラムに対して不具合箇所を検出した結果である。本ツールは、測定対象のプログラムを構成するファイル群を左側に列挙する。そして、C&K メトリクスによるプログラムの測定結果を右側に列挙する。測定結果を画面上に出力する際、不具合箇所を示す数値に対して色を付けて区別する。図 2 では、網かけになっている部分が不具合箇所を示している。

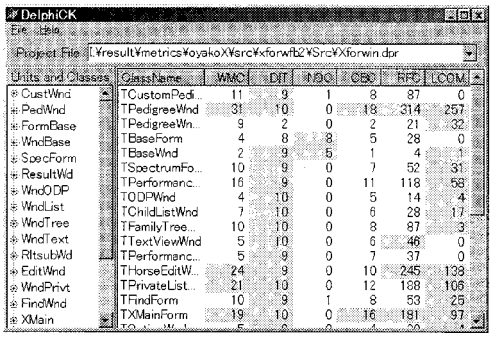

図2 不具合箇所の検出結果

3.3 不具合箇所の改善順序の提示機能

複数の不具合箇所を検出した場合、本ツールは、どの不具合箇所を改善すればよいか、改善順序を提示する機能を持つ。この機能を実行した結果を図 3 に示す。図 3 は、検出された不具合箇所を改善するべき順に示している。例えば、300:NOC:TBaseForm は、改善優先度 300 で尺度 NOC についてクラス TBaseForm を改善するべきことを示す。

本ツールは、優先的に改善するべき不具合箇所を示すために改善優先度を用いている。改善優先度は値の大きい程、優先的に改善するべきことを示す。そのため、改善優先度の値の大きいものから順に示す。(1) 式で、$P_{c,m}$ は改善優先度を意味し、c はクラスを意味し、m は C&K メトリクスの尺度である WMC、DIT、NOC、CBO、RFC、および LCOM のうちのどれかを意味する。$V_{c,m}$ はクラス c に対する尺度 m についての測定値である。t_m は尺度 m のしきい値である。

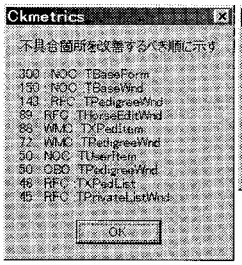

図3 不具合箇所の改善順序

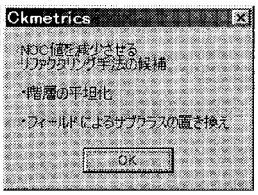

図4 NOCの値を減少させる
リファクタリング手法の候補

$$P_{c,m} \stackrel{\text{def}}{=} \begin{cases} V_{c,m} & (t_m = 0 \text{ の時}) \\ \frac{V_{c,m} - t_m}{t_m} & (\text{それ以外の時}) \end{cases} \quad (1)$$

3.4 リファクタリング手法の提示機能

本ツールは、ソフトウェアメトリクスとリファクタリング手法の相関を用いて、適切なリファクタリング手法を提示する機能を持つ。こうして、適用するリファクタリング手法の決定を支援する。不具合箇所を選んでこの機能を実行した結果を図 4 に示す。図 4 は、クラス TBaseForm の NOC の値を減少させるリファクタリング手法を 2 つ示している。

本ツールは、不具合箇所を改善するリファクタリング手法を提示するために、ソフトウェアメトリクスによる測定値の変化とリファクタリング手法との関係を用いている。検出された不具合箇所を指定すると、不具合箇所の測定値を減少させるリファクタリング手法を示す。

4 おわりに

本稿では、リファクタリングの作業を支援するツールについて述べた。今後は、リファクタリング手法を適用する際に、本ツールを用いる。そして、本ツールの不具合箇所に対する改善順序の提示機能の有効性と、リファクタリング手法の提示機能の有効性を検証する予定である。

参考文献

[1] W.F. Opdyke:“Refactoring Object-Oriented Frameworks,” Ph.D. diss., University of Illinois at Urbana-Champaign, 1992.

[2] S.R.Chidamber and C.F.Kemerer:“ A Metrics Suite for Object Oriented Design,” IEEE Transaction on Software Engineering, Vol.20, No.6, pp.476-493, 1994.

[3] M.Fowler:“Refatoring:Improving The Design of Existing Code,” 1999, 児玉公信, 友野晶夫, 平沢章, 梅沢真史訳:「リファクタリング:プログラムの体質改善テクニック」, ピアソン・エデュケーション, 2000.