

Title	メトリクスの測定によるリファクタリング支援の自動化
Author(s)	田畑, 敦史
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4347
Rights	
Description	Supervisor:鈴木正人, 情報科学研究科, 修士

メトリクスの測定によるリファクタリング支援の自動化

田畑 敦史 (610057)

北陸先端科学技術大学院大学 情報科学研究科

2008 年 2 月 7 日

キーワード: ソフトウェア保守、リファクタリング, BadSmell, ソフトウェアメトリクス, 支援環境.

1 背景と目的

ソフトウェア作成の過程において、度重なる変更の結果、ソースコードが複雑化し保守・拡張が困難となってしまうという問題がある。

そこで、複雑になったコードを改善する作業であるリファクタリングを行うことによって、ソフトウェアの保守・拡張性を維持する必要がある。しかしながら、ソースコード全体の中から改善箇所を発見することは困難である。

本研究は、メトリクスを利用した手法により、ソースコード内からリファクタリングの対象箇所の発見とその解決方法の提案を自動化することにより、リファクタリングを行う際の支援を行うことを目的としている。

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支援の自動化

リファクタリング支援の先行研究として、Fowler によるリファクタリング開始終了判定およびコード改善のための操作の定義による支援と、メトリクスによるリファクタリングの評価支援がある。

Fowler が提案した支援は、BadSmell というリファクタリングの対象を定義し、それがソースコード上に存在するかどうかで開始終了の判定および、それを解決するためにソースコードに対して行うリファクタリング操作の提案を行うことにより、コードの改善に対する支援を行っている。

メトリクスによる支援は、リファクタリング操作前後のメトリクスの値を比較することで、どの程度ソースコードが改善されたかを測定する。

本研究は、この二つの支援を組み合わせることでメトリクスから Fowler が定義する BadSmell を求めることにより、開始終了判定、改善箇所の検出、およびリファクタリング操作候補を求める支援を提案し、それを自動化することにより支援の自動化を実現する。

3 設計と実装

ソフトウェアメトリクスを利用した支援の自動化によるリファクタリングは、(1)メトリクスから BadSmell を求める、(2)BadSmell を解決するためのリファクタリング操作候補を求める、(3) 操作を実行して BadSmell の解消を試みる、という手順をソースコードからすべての BadSmell が無くなるまで繰り返す。

この一連の流れを実装するには、(1)BadSmell を求めるためのメトリクスの測定、(2)メトリクスの値から BadSmell を求める作業、(3)BadSmell を解決するためのリファクタリング操作候補を求める作業、の自動化を行わなければならない。

メトリクスの測定では、BadSmell の特徴からそれに対応するメトリクスを定義し、メトリクスの値の測定を自動化する

BadSmell を求める作業の自動化では、求めたメトリクスの値から BadSmell である箇所を探す作業を自動化する

リファクタリング操作候補を求める作業では、Fowler の提案する BadSmell とリファクタリング操作候補を利用して求める作業を自動化する

GUI による総合開発環境の一部に実装を行う目的で、支援の自動化は Eclipse プラグインとして実装を行う。

ソフトウェアプロジェクト内のソースコードの BadSmell をメトリクスを利用して求め、ソースコードの警告一覧である Problem View に登録することにより、ユーザに BadSmell の存在とそのコード上の箇所を確認させて開始終了判定、改善箇所をユーザに伝える支援を行う。

また、警告を解消するための動作一覧である Quick Fix を利用して、解決するためのリファクタリング操作候補の名前を表示させることにより、ユーザに解決のためのリファクタリング操作候補を求める支援を行う。

4 支援環境の適用実験

自動化した支援による有用性を確認するため、Eclipse プラグインとして実装した自動化の支援を使用して、BadSmell を含むソースコードのリファクタリングを行い、実際にどの程度 BadSmell が発見して解決できるかの実験を行った

手順としては、(1)Problem View による BadSmell の有無の確認、(2)Quick Fix から BadSmell を解決するためのリファクタリング操作候補の確認および選択、(3) リファクタリング操作の実行、を BadSmell が無くなるまで繰り返しリファクタリングを行った。

5 支援の評価

メトリクスを利用した BadSmell の発見では、BadSmell としての特徴が顕著でありメトリクスの値が高い箇所を発見することができたが、BadSmell としての特徴が隠微でメト

リクスの値が低い箇所は、BadSmell であるかの区別ができないため発見することはできなかった。

このことから、改善の必要性が高く保守・拡張の際の大きな妨げとなる部分の発見は可能だが、改善の必要性が低い箇所を見つけるには不十分であることが分かった。

又、支援の自動化により、ソースコード上の BadSmell の有無の確認、およびリファクタリング前後でのメトリクスの値の変化の測定を行う手間が無くなり、作業時間を短縮することができる。

6 まとめ

ソフトウェアメトリクスを利用した支援の自動化では、コード内の改善の必要性が高い箇所の発見に向いていることから、コードの保守・拡張性の低下を防ぐには有用であるといえる。

又、この支援を利用することによって作業時間の短縮させることにより、ソフトウェア開発の過程でリファクタリングを行う際のコストの削減を期待できる。