

プロセスメトリクスを用いたメソッド抽出リファクタリングの推薦に向けて

田中 大樹¹ 崔 恩瀨¹ 吉田 則裕² 藤原 賢二³ 飯田 元¹

概要：開発者はソフトウェア保守作業を行う際、ソースコードの保守性や可読性を向上させるために、リファクタリングを実施する必要がある。本稿は、プロセスメトリクスを用いたメソッド抽出リファクタリングの推薦を行うために、我々の既存の研究を紹介し、本ワークショップで今後の課題を議論する。

Toward Recommendation of Extract Method Refactoring Using Process Metrics

1. はじめに

リファクタリングとは、外部から見たときの振る舞いを保ちつつ、保守性や可読性を向上させるために、ソフトウェアの内部構造を整理することである [1]。Fowler は自分の本でリファクタリングパターンを定義した上に、どのようなコード片に対してどのリファクタリングパターンを実施すべきかが示されている。

開発者はソフトウェア保守作業を行う際、ソースコードの保守性や可読性を向上させるために、リファクタリングを実施する必要がある。しかし、大規模なソースコードからリファクタリング候補を見つけることは困難であるため、開発者のリファクタリング作業を支援するための技術やツールが多く提案されている [2] [3]。しかし、これらの技術やツールを、開発者はあまり利用していないという報告もある [4]。そのため、研究者はより広く利用されるリファクタリング支援ツールを開発するために、開発者のリファクタリング作業の実態をより正確に把握したうえで、支援手法を検討していく必要がある。

ソースコードの特徴を定量的に表現する方法として、プロダクトメトリクス以外に、開発の履歴情報を用いて計測するプロセスメトリクスがある [5]。開発プロセスはソースコードの品質に影響を与える。リファクタリングは一般的に、保守性や可読性が低い品質の悪いコード片を対象に

実施されるため、ソースコードの品質に影響を与える開発プロセスから計測されるプロセスメトリクスと、リファクタリングの実施には相関があると考えられる。しかし、プロセスメトリクスとリファクタリングの関係について、定量的な調査は実施されていない。

そこで、我々の既存研究では、メソッド抽出の対象となったメソッドの特徴を調査し、メソッド抽出の対象となるメソッドを推薦する機械学習モデルを構築し、その有効性を評価した [6]。今後、我々の既存研究および本ワークショップで議論したいことを紹介する。

2. 既存研究の紹介

我々の既存研究ではプロセスメトリクスを用いて、メソッド抽出の対象となったメソッドの特徴を調査した [6]。詳細には、オープンソースプロジェクトである jEdit^{*1} の 7,707 リビジョン個の対象に藤原らの提案した Kenja [7] を用いて、メソッド抽出リファクタリングの検出を行った。その後、ソースコードの品質に影響を与えていると考えられる、コード片の変更回数や、その変更に関わった開発者数等のメトリクスを用いて抽出の対象となったメソッドと、対象とならなかったメソッドを比較し、コード片の変更回数とオーサー数、コード片の変更量である追加・削除・変更行数に、メソッド抽出の対象となったメソッドと、対象とならなかったメソッド間で有意差が確認できた。表 1 は調査対象のプロセスメトリクスの一覧と比較結果を表して

¹ 奈良先端技術大学院大学情報科学研究科

² 名古屋大学大学院情報科学研究科

³ 豊田工業高等専門学校情報工学科

^{*1} <http://www.jedit.org>

表 1 計測プロセスメトリクスと比較結果

メトリクス	有意差あり
ソースコードの変更回数	○
ソースコードのオーサー数 存在期間	○
ソースコードの追加行数	○
ソースコードの削除行数	○
ソースコードの変更行数	○

いる。

また、プロセスメトリクスの調査結果を踏まえ、メソッド抽出の対象となるメソッドを推薦する機械学習モデルを構築し、その有効性を評価した。学習に利用するメトリクスの組み合わせを変化させた 6 種類のモデルを構築し、それらの精度を比較した。実験の結果、従来より利用されてきたプロダクトメトリクスと、本研究の調査対象であるプロセスメトリクスの両方を利用した推薦モデルが、F 値が 0.827 と最も高い精度を示した。

3. 議論したいこと

2 章で紹介した我々の既存研究の結果を踏まえて、本ワークショップでは以下の話題に関して議論を行いたい。

対象リファクタリングパターンに関して： 我々の既存研究では、メソッド抽出リファクタリングパターンのみを対象にした。しかし、Murphy-Hill の調査によると、開発者はメソッド抽出リファクタリングパターン以外にメソッドのインライン化リファクタリングパターンなどのリファクタリングパターンも頻繁に実施している [4]。そのために、メソッド抽出リファクタリングパターン以外のリファクタリングパターンにも支援が必要であり、それらのリファクタリングパターンに関しても調査する必要がある。しかし、Fowler 提案した、63 個のリファクタリングパターンの中でどのリファクタリングパターンを調査対象にするかを決めなければならないがどのリファクタリングパターンを調査対象に関してを検討する必要がある。

リファクタリング検出に関して： 我々の既存研究では、コミット単位でメソッド抽出リファクタリングを検出するツール Kenja[7] を用いた。現在まで多くのリファクタリング検出ツールが提案されてきたが、コミット単位でリファクタリング検出を行うツールは少ない [8]。そのために、コミット単位で効率的にリファクタリング検出を行うためにどうしたらいいかを検討する必要がある。

調査対象プロセスメトリクスに関して： 我々の既存研究では、ソースコードの変更回数、ソースコードのオーサー数、ソースコードの追加行数、ソースコードの削除行数、ソースコードの変更行数の 6 個のプロセスメトリクスを対象にして調査を行った。しかし、バグの

混入など他の多くのプロセスメトリクスが存在する。そのため、既存のプロセスメトリクスからどのプロセスメトリクスを調査対象にするべきかを検討する必要がある。

4. まとめ

本稿では、プロセスメトリクスを用いたメソッド抽出リファクタリングの推薦を行うために、我々の既存の研究を紹介し、本ワークショップで今後の課題を議論する。今後の課題として、より多いリファクタリングパターンおよびプロセスメトリクスを対象に調査を行い、プロセスメトリクスを用いたメソッド抽出リファクタリングの推薦ツールを開発する。

参考文献

- [1] Fowler, M.: *Refactoring: Improving the Design of Existing Code*, Addison Wesley (1999).
- [2] Tsantalis, N. and Chatzigeorgiou, A.: Identification of extract method refactoring opportunities for the decomposition of methods, *The Journal of Systems and Software*, Vol. 84, pp. 1757–1782 (2011).
- [3] 後藤 祥, 吉田則裕, 藤原賢二, 崔 恩澗, 井上克郎: 機械学習を用いたメソッド抽出リファクタリングの推薦方法, *情報処理学会論文誌*, Vol. 56, No. 2, pp. 627–636 (2015).
- [4] Murphy-Hill, E., Parnin, C. and Black, A. P.: How We Refactor, and How We Know It, *IEEE Transactions on Software Engineering*, Vol. 38, No. 1, pp. 5–18 (2012).
- [5] 畑 秀明, 水野 修, 菊野 亨: 不具合予測に関するメトリクスについての研究論文の系統的レビュー, *コンピュータソフトウェア*, Vol. 29, No. 1, pp. 106–117 (2012).
- [6] 田中大樹, 崔 恩澗, 吉田則裕, 藤原賢二, 飯田 元: プロセスメトリクスを用いたメソッド抽出事例の調査と予測モデルの構築, *電子情報通信学会技術研究報告* 512, SS2016-73, 電子情報通信学会 (2017).
- [7] 藤原賢二, 吉田則裕, 飯田 元: 構文情報リポジトリを用いた細粒度リファクタリング検出手法, *情報処理学会論文誌*, Vol. 56, No. 1, pp. 47–59 (2015).
- [8] 崔 恩澗, 藤原賢二, 吉田則裕, 林 晋平: 変更履歴解析に基づくリファクタリング検出技術の調査, *コンピュータソフトウェア*, Vol. 32, No. 12, pp. 2346–2357 (2015).