

令和 4 年 6 月 27 日現在

機関番号：12608

研究種目：基盤研究(C)（一般）

研究期間：2018～2021

課題番号：18K11238

研究課題名（和文）探索と対話の融合による半自動リファクタリング環境の確立

研究課題名（英文）Search-Based and Interactive Environment for Semi-Automated Refactoring

研究代表者

林 晋平（Hayashi, Shinpei）

東京工業大学・情報理工学院・准教授

研究者番号：40541975

交付決定額（研究期間全体）：（直接経費） 3,300,000円

研究成果の概要（和文）：本研究では、より現実的な自動リファクタリングの実現に向けて、保守性に留まらない様々な観点を考慮したリファクタリング探索を実現すると共に、探索の対話的制御のためのリファクタリング結果の効率的なレビュー方法の確立を目指すものである。主たる結果として、ソースコード品質以外にレビューの容易性を考慮する、多目的最適化に基づくリファクタリング探索手法を開発した。また、リファクタリングを適用すべき品質低下箇所を特定する手法を開発した。さらに、複雑にリファクタリングが行われた変更をレビューするための環境を実現した。

研究成果の学術的意義や社会的意義

本研究で提案する手法は、これまで主にソースコードの保守性に基づいていたリファクタリング探索技術を、後段の分析の容易性を考慮しながら行うもので、このような品質特性に基づく手法には学術的新規性がある。また、提案手法の考え方は自動リファクタリング探索手法が出力するリファクタリングの適用可能性を向上させ、これまで十分には利用されてこなかったリファクタリング探索の技術を通常のソフトウェア開発に浸透させることに貢献するため、多様なソフトウェア開発の品質を向上させることが期待できる。

研究成果の概要（英文）：In this research, we aim to realize a more realistic automated refactoring environment, taking into account various perspectives in addition to the maintainability, and an efficient review environment of the recommended refactorings that enables an interactive configuration of the refactorings. As the main results, we have developed a new search-based refactoring technique based on a multi-objective optimization that considers the review effort in addition to the source code quality. We have also developed a method to identify smelly code fragments where refactorings should be applied. Furthermore, we have implemented an environment for reviewing refactoring-related complex changes.

研究分野：ソフトウェア工学

キーワード：リファクタリング ソフトウェア自動進化 探索ベースソフトウェア工学 対話的システム 不吉な臭い

## 様式 C - 19、F - 19 - 1、Z - 19 (共通)

### 1. 研究開始当初の背景

リファクタリングとは、既存プログラムの外的振る舞いを変更せずにその内部構造を変換することで、該当プログラムコードの理解容易性や変更容易性を安全に向上させるプロセスを指す。継続的なソフトウェア開発においては、ソフトウェア品質は次第に劣化していくため、リファクタリングの適用等により品質を維持し続けることは重要である。

また、計算機により適用すべきリファクタリング操作の列を自動的に特定し、その適用により自動的に品質を向上させる自動リファクタリングの方式も研究されてきた。この方式では、プログラムの版を状態、リファクタリング操作を状態遷移とする状態空間に対してメタヒューリスティック探索を試み、あらかじめ定めた適合度関数をよく満たす解を求める。これまでの手法では、ソフトウェア保守性のメトリクス等を組み合わせたものを適合度関数として用い、メトリクス値が最適となるようなプログラムの版を、保守性に優れたものとして開発者に推薦する。

### 2. 研究の目的

残念ながら、前章で述べたような自動リファクタリングが開発者に受け入れられてきたとは言いがたい。このことは、以下のような問題に起因すると考え、これらの解決を目指した。

(1) 提案されたリファクタリング群が、必ずしも利用者の目標にそぐわない。既存の自動リファクタリング手法は、ソフトウェア全体の保守性指標を定量化したメトリクス値を適合度関数に用い、その値を改善するようリファクタリング列を提案する。しかしながら、保守性指標にはトレードオフの関係があり、ある改善が特定の開発者に受け入れられるとは限らない。また、提案されたリファクタリング列のテストやレビューが容易であるかなど、開発者が注目する観点は保守性の改善だけに限らないため、得られたリファクタリング列がこういった観点を含めた開発者の基準を満足しなければ、採用されない。

(2) 現実的な構造変化は、必ずしもこれまでによく知られたリファクタリング操作の組合せとしては表現できず、その途中の過渡的な段階を考慮する必要がある。

(3) 行われたリファクタリング群のレビューが難しい。開発者は必ずしも自動リファクタリングツールを信用しておらず、得られた操作列はレビューによる検証を経る必要がある。しかし、開発者がリファクタリングの影響を正しく把握することは容易ではない。適用されたリファクタリングが多くなったり、複雑になったりすると、大きな変更差分を生み出すため、レビューのための変更把握が特に難しくなる。

### 3. 研究の方法

前章での問題点を解決するため、本研究では、探索に基づく自動リファクタリング方式を拡張する形で、より開発者に受け入れられるようリファクタリングを導出することを目的とする。特に、探索の目標の拡張、探索操作の拡充を行うことにより、探索可能なリファクタリングの範囲を拡大し、また開発者に受け入れやすいリファクタリング列を出力できるようにする。また、実施すべきリファクタリングのレビューを支援するための環境を構築する。これらに基づき、インタラクティブなリファクタリング環境の実現を目指す。

## 4. 研究成果

本研究は、より現実的な自動リファクタリングの実現に向けて、探索処理を向上させるとともに、対話的制御のためのリファクタリング結果の効率的なレビュー方法の確立を目指すものである。主要なものとして、以下の3つの研究成果を達成した。

### (1) 探索に基づくリファクタリング自動推薦手法及びリファクタリング検出法の開発

開発者間の議論経験やモジュールの所有者情報に基づいてリファクタリング結果のレビュー労力を見積もる適応度関数を設計し、ソースコード品質に基づく適応度関数と組み合わせることにより、ヒューリスティック探索に基づく多目的最適化としてリファクタリング探索を定式化し、これを解くことにより実現可能性の高いリファクタリング操作列を導出する手法を開発した。また、特定のメソッド内で行われた変更において行われたリファクタリングをローカルリファクタリングと定義し、これをヒューリスティック探索により検出する手法を提案し、本研究で開発した自動探索ツールのプロトタイプを応用して実現した。オープンソースソフトウェアの変更履歴から新しいローカルリファクタリング操作を特定し、またその一部を自動検出に組み込んだ。さらに、特定したローカルリファクタリングの検出精度を評価した。

### (2) 不吉な臭いの検出および利活用

適用すべきリファクタリングの選定要因として、改善すべきソースコードの低品質箇所を表す不吉な臭いの新しい検出法および分析法を構築した。ソースコードの品質低下の予防も目的として考慮し、すでに品質が低下した箇所だけでなく、将来の品質低下につながる箇所も予防の観点からリファクタリング適用対象モジュールの候補として考え、選定に利用可能な特徴として腐敗モジュールを提案した。オープンソースソフトウェアの変更履歴に基づき、腐敗モジュールの特徴を分析したうえで、これを予測する手法を開発した。また、不吉な臭いの情報を、ソフトウェア変更記述とモジュールとの対応付けに応用する手法を提案するとともに、その特徴を分析した。さらに、リファクタリングの適用範囲を拡大すべく、要求モデル等のソースコード以外のソフトウェア成果物に対しての臭いの検出も試みた。

### (3) リファクタリング変更の理解に貢献するレビュー環境

複数のリファクタリングが混在した差分のレビューを効率よく行えるよう、行われた変更間の依存関係を考慮しながら変更の粒度を調整するためのインタラクティブな分析環境のプロトタイプを複数開発し、それらの評価を行った。また、リファクタリング変更のレビューをより正確に行うため、開発者の行ったソースコード変更を精密に記録できる開発環境も実現した。さらに、GitHubで管理された開発プロジェクトにおけるソースコード変更からリファクタリングの実例を効率よく収集・管理するための、変更の注釈付け環境の設計およびそのプロトタイプの実装を行った。

## 5. 主な発表論文等

〔雑誌論文〕 計23件（うち査読付論文 23件 / うち国際共著 3件 / うちオープンアクセス 23件）

1. 著者名 Shinpei Hayashi, Keisuke Asano, Motoshi Saeki	4. 巻 105-D(5)
2. 論文標題 Automating Bad Smell Detection in Goal Refinement of Goal Models	5. 発行年 2022年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 837-848
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2021KBP0006	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -
1. 著者名 Yotaro Seki, Shinpei Hayashi, Motoshi Saeki	4. 巻 105-D(5)
2. 論文標題 Cataloging Bad Smells in Use Case Descriptions and Automating Their Detection	5. 発行年 2022年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 849-863
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2021KBP0008	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -
1. 著者名 Aoi Takahashi, Natthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki: "An Extensive Study on Smell-Aware Bug Localization	4. 巻 178(110986)
2. 論文標題 An Extensive Study on Smell-Aware Bug Localization	5. 発行年 2021年
3. 雑誌名 Journal of Systems and Software	6. 最初と最後の頁 1-17
掲載論文のDOI (デジタルオブジェクト識別子) 10.1016/j.jss.2021.110986	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -
1. 著者名 Natthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki	4. 巻 E104-D(10)
2. 論文標題 Supporting Proactive Refactoring: An Exploratory Study on Decaying Modules and Their Prediction	5. 発行年 2021年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 1601-1615
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2020EDP7255	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Yoshiki Higo, Shinpei Hayashi, Shinji Kusumoto	4. 巻 165(110571)
2. 論文標題 On Tracking Java Methods with Git Mechanisms	5. 発行年 2020年
3. 雑誌名 Journal of Systems and Software	6. 最初と最後の頁 1-13
掲載論文のDOI (デジタルオブジェクト識別子) 10.1016/j.jss.2020.110571	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Katsuhisa Maruyama, Shinpei Hayashi, Takayuki Omori	4. 巻 E103-D(11)
2. 論文標題 ChangeMacroRecorder: Accurate Recording of Fine-Grained Textual Changes of Source Code	5. 発行年 2020年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 2262-2277
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2020EDK0001	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Yoshiki Higo, Shinpei Hayashi, Hideaki Hata, Meiyappan Nagappan	4. 巻 25(3)
2. 論文標題 Ammonia: an approach for deriving project specific bug patterns	5. 発行年 2020年
3. 雑誌名 Empirical Software Engineering	6. 最初と最後の頁 1951-1979
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/s10664-020-09807-w	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 該当する

1. 著者名 高橋 碧, セーリム ナッタウット, 林 晋平, 佐伯 元司	4. 巻 60(4)
2. 論文標題 情報検索に基づくBug Localizationへの不吉な臭いの利用	5. 発行年 2019年
3. 雑誌名 情報処理学会論文誌	6. 最初と最後の頁 1040-1050
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Bushra Aloraini, Meiyappan Nagappan, Daniel M. German, Shinpei Hayashi, Yoshiki Higo	4. 巻 158
2. 論文標題 An Empirical Study of Security Warnings from Static Application Security Testing Tools	5. 発行年 2019年
3. 雑誌名 Journal of Systems and Software	6. 最初と最後の頁 110427
掲載論文のDOI (デジタルオブジェクト識別子) 10.1016/j.jss.2019.110427	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 該当する

1. 著者名 Natthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki	4. 巻 E101-D(7)
2. 論文標題 An Investigative Study on How Developers Filter and Prioritize Code Smells	5. 発行年 2018年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 1733-1742
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2017KBP0006	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

1. 著者名 Shinpei Hayashi, Fumiki Minami, Motoshi Saeki	4. 巻 E101-D(7)
2. 論文標題 Detecting Architectural Violations Using Responsibility and Dependency Constraints of Components	5. 発行年 2018年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 1780-1789
掲載論文のDOI (デジタルオブジェクト識別子) 10.1587/transinf.2017KBP0023	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

〔学会発表〕 計38件 (うち招待講演 1件 / うち国際学会 17件)

1. 発表者名 陳 磊, 林 晋平
2. 発表標題 探索に基づくリファクタリング推薦におけるレビュー工数見積もりの利用
3. 学会等名 電子情報通信学会ソフトウェアサイエンス研究会 (2022年7月)
4. 発表年 2022年

1. 発表者名 Ryo Kuramoto, Motoshi Saeki, Shinpei Hayashi
2. 発表標題 RefactorHub: A Commit Annotator for Refactoring
3. 学会等名 29th IEEE/ACM International Conference on Program Comprehension (国際学会)
4. 発表年 2021年

1. 発表者名 Mahfouth Alghamdi, Shinpei Hayashi, Takashi Kobayashi, Christoph Treude
2. 発表標題 Characterising the Knowledge about Primitive Variables in Java Code Comments
3. 学会等名 18th IEEE/ACM International Conference on Mining Software Repositories (国際学会)
4. 発表年 2021年

1. 発表者名 Satoshi Yamashita, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 ChangeBeadsThreader: An Interactive Environment for Tailoring Automatically Untangled Changes
3. 学会等名 27th IEEE International Conference on Software Analysis, Evolution and Reengineering (国際学会)
4. 発表年 2020年

1. 発表者名 筒井 湧暉, セーリム ナッタウト, 林 晋平, 佐伯 元司
2. 発表標題 探索に基づくローカルリファクタリングの検出
3. 学会等名 情報処理学会第204回ソフトウェア工学研究発表会
4. 発表年 2020年

1. 発表者名 Ryosuke Funaki, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 The Impact of Systematic Edits in History Slicing
3. 学会等名 16th International Conference on Mining Software Repositories (国際学会)
4. 発表年 2019年

1. 発表者名 Natthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 Toward Proactive Refactoring: An Exploratory Study on Decaying Modules
3. 学会等名 3rd International Workshop on Refactoring (国際学会)
4. 発表年 2019年

1. 発表者名 Yotaro Seki, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 Detecting Bad Smells in Use Case Descriptions
3. 学会等名 27th IEEE International Requirements Engineering Conference (国際学会)
4. 発表年 2019年

1. 発表者名 Natthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 Can Automated Impact Analysis Techniques Help Predict Decaying Modules?
3. 学会等名 35th IEEE International Conference on Software Maintenance and Evolution (国際学会)
4. 発表年 2019年



1. 発表者名 Aoi Takahashi, Matthawute Sae-Lim, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 A Preliminary Study on Using Code Smells to Improve Bug Localization
3. 学会等名 26th IEEE/ACM International Conference on Program Comprehension (国際学会)
4. 発表年 2018年

1. 発表者名 Sarocha Sothornprapakorn, Shinpei Hayashi, Motoshi Saeki
2. 発表標題 Visualizing a Tangled Change for Supporting Its Decomposition and Commit Construction
3. 学会等名 42nd IEEE Computer Software and Applications Conference (国際学会)
4. 発表年 2018年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究協力者	佐伯 元司  (Saeki Motoshi)  (80162254)	南山大学・理工学部・教授    (33917)	

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関		
オーストラリア	The University of Adelaide		
カナダ	University of Waterloo	University of Victoria	