

科学研究費助成事業 研究成果報告書

平成 27 年 6 月 12 日現在

機関番号：12608

研究種目：若手研究(B)

研究期間：2011～2014

課題番号：23700030

研究課題名(和文) 理解と品質維持に適したソースコード修正を支援する開発環境の構築

研究課題名(英文) Supporting Techniques and Tools for Understandable and Quality Source Code Changes

研究代表者

林 晋平 (Hayashi, Shinpei)

東京工業大学・情報理工学(系)研究科・助教

研究者番号：40541975

交付決定額(研究期間全体)：(直接経費) 2,800,000円

研究成果の概要(和文)：本研究は、開発者がソースコードに対して行う変更を支援するために、変更すべき箇所の特
定に加え、開発の後段での理解が容易になるよう、また対象ソフトウェアに与える品質の悪影響を抑制するよう変更を
構成可能とすることを目的としている。変更箇所の特定、変更の再構成および分析に関する種々のツールを開発し、ま
たそれらが高品質な変更の構成に役立つことを示した。

研究成果の概要(英文)：In this research project, in order to support the source code change process of
developers, we have developed supporting techniques and tools for composing source code changes of high
quality in addition to locating the code fragments to be modified. We also showed the effectiveness and
usefulness of our techniques and tools.

研究分野：ソフトウェア工学

キーワード：ソフトウェア進化 リファクタリング ソフトウェア構成管理 ソフトウェア開発環境 機能検索

1. 研究開始当初の背景

ソフトウェアは一度構築された後も、要求変更の対応、環境の変化への適応、バグ修正、リリースのための調整など、絶えず様々に変更される。しかし変更は、例えば機能追加を目的とした変更が意図せず保守性に悪影響を及ぼすなど、その主目的に貢献する他にも対象ソフトウェアに影響するため、注意深く構成する必要がある。またチーム開発では全変更が版管理リポジトリに蓄えられるが、これらは他の開発者によるレビューを受けたり、対象ソフトウェアの理解のために調査されたりするなど、変更そのものも分析や理解の対象となるため、理解が容易なよう変更を構成する必要がある。しかし適切な変更は、開発者のスキルや開発プロジェクトのポリシーや過去の開発経緯など、様々な経験や知識を必要とするため、簡単ではない。

2. 研究の目的

前述した研究背景に基づき、本研究では、開発者がソースコードに対して行う変更を支援するために、変更すべき箇所の特定に加え、開発の後段での理解が容易になるよう、また対象ソフトウェアに与える品質の悪影響を抑制するよう構成可能な開発環境を実現することを目的とした。

3. 研究の方法

研究目的を達成するために必要な要素技術として、以下の研究課題に取り組んだ。

(1) 変更箇所の特定：ソフトウェア変更を円滑に行うためには、変更に先立っての対象プログラムの理解が重要となる。新規機能の追加など、ソフトウェアの変更要求が与えられた際に、ソースコードの修正すべき箇所の候補を特定する。

(2) 変更の再構成・分析：行われた変更を、後段の理解が容易となるよう、後段の構成管理に適した形に再構成する。また、リファクタリング等が混在して理解が困難となったソースコード差分を分析し、開発者にとって理解しやすい形で表示する。

(3) プログラムの高品質化：リファクタリング操作などを適用し、変更対象のプログラムおよびこれから行おうとしている変更内容の品質を向上させる。また、変更対象の責務をプログラムのどのモジュールに割り当てるべきかを推薦し、不適な点を改善する。

4. 研究成果

以下に主な研究成果を述べる。

(1) 変更箇所の特定

ソフトウェアの変更要求が与えられた際に、ソースコードの修正すべき箇所の候補を特定する機能検索手法について、新しい方法論及びツールを実現し、またその有効な利用法について分析を行った。

主なものとして、ソフトウェア要求仕様の記述法のひとつであるユースケース記述のような手順書を対象とした機能検索法を開発した。この手法では、手順書が示す手順間の依存関係を特定し、これを制約として用いることにより、既存の機能検索結果をフィルタリングし、より精度の高い結果を得るものである。依存関係の概要を図1に示す。図上部は対象としている手順を定義したユースケース記述である。この記述には、実際の機能が手順の順番通りに実行されるという時間の関係や、手順1で定義されたデータが手順2に渡されるといったデータの依存関係が存在する。これを、各手順を節、手順間の依存関係を辺としてグラフ化したものが図下部である。このグラフを構造的なクエリとみなし、依存関係に基づく制約を満たす機能検索結果の組を手順に対する検索結果として得る。

実際にベンチマークソフトウェア AquaLushを対象とした比較実験を行ったところ、フィルタリングによるランキングの精度が1~25向上したことを確認できている。

また、こういった機能検索の自動化手法を開発者が有効に活用するための手法の設計方針について、被験者が機能検索を行う様子を観察した実験の結果に基づいて分析した。

(2) 変更の再構成・分析

ソフトウェア開発では、ソースコード本体のみならず、その編集の履歴もさまざまに利用されるため、適切に管理された履歴が後段のプロセスに貢献する。しかし、現実には複雑に絡み合った不適切な履歴が生じうる。

我々のアプローチでは、編集履歴を、理解性や利用性を向上させるために、その編集内容を変えないよう書き換えるリファクタリ

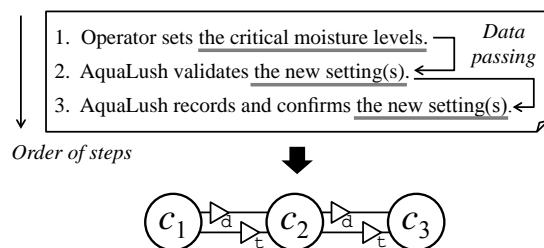


図1 手順間の依存関係

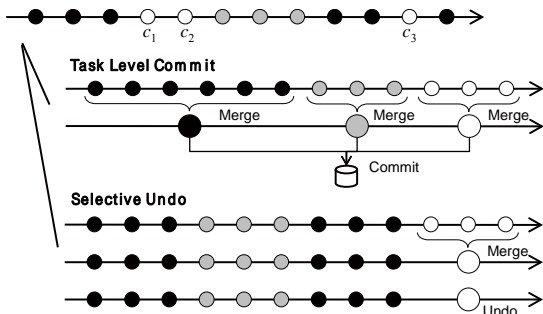


図2 編集履歴リファクタリングの利用

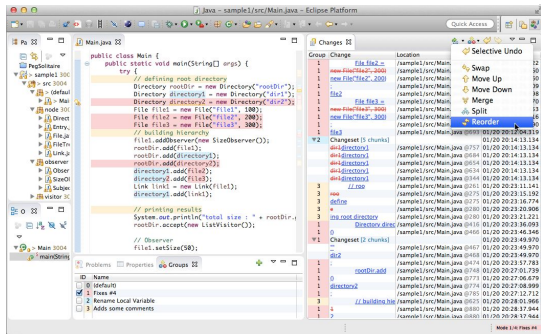


図3 ツール Historef の動作例

ングを行う。本研究では、4つの基本的なリファクタリング及びそれらを組み合わせた大きなリファクタリングを、それらの事前条件も含めて定義した。また、定義したリファクタリングを自動化しコードエディタに組み込んだ支援ツールを実現した。支援ツールを用いることにより、実際に編集履歴のリファクタリングが行えることや、リファクタリングにより図2に示すような分割コミットや変更の選択的取り消しなどの有益な応用が可能となることを示した。さらに、編集履歴リファクタリングが必要となる履歴の兆候を分析することにより、編集操作の自動分類も実現した。

支援ツールHistorefの動作例を図3に示す。この支援ツールは統合開発環境Eclipseのプラグインとして実装されており、この図ではEclipseを用いてJavaのソフトウェア開発を行う際に、複数のタスクを混在させてソースコードの編集を行った様子を示している。支援ツールを利用することにより、複雑にもつれた編集操作を構成管理に適した形に再構成することができる。また、様々なリファクタリング操作の組み合わせによりプログラムの構造を大きく変更した場合も、その編集操作を適切に構成し直す支援を実現している。実際、20のリファクタリング操作が混在する履歴を整理することができる。

変更の分析として、リファクタリングが混在した差分の整理ツールRediffsも開発した。このツールでは、プログラム差分の解析により、版間に行われたリファクタリング操作をあらかじめ検出しておき、これらの操作の影

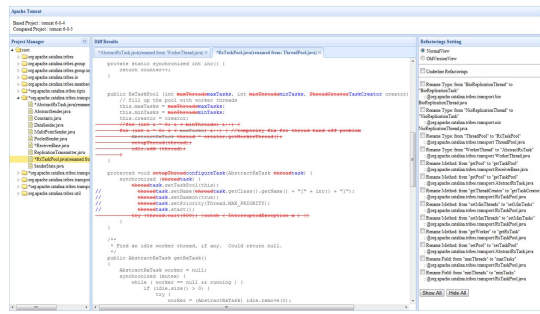


図4 ツール Rediffs の動作例

響を差分から除去可能とする。分析ツールの動作例を図4に示す。この図では、オープンソースソフトウェアApache Tomcatの版間差分から行われたリファクタリング操作を抽出し、その影響を除去している様子を示している。図右部のビューよりリファクタリング操作を選択することにより、該当のリファクタリング操作の影響の有無がリアルタイムに変更され、中央部のビューに表示された差分が更新される。このツールを用いることにより、差分全体の理解に先立って、リファクタリング操作の影響を除外した、振る舞いに関する小さな差分の理解を行うことができる。

(3) プログラムの高品質化

新しい機能等を既存プログラムのどのモジュールに割り当てればよいかを決定する責務割り当てに対しても、既存のパターンを用いた手法、制約充足ソルバを用いた手法を開発し、実際にその有用性を確認した。図5はパターンを用いた責務割り当て支援手法を実現した支援ツール RAST の動作例である。このツールは責務割り当て図のエディタを拡張することにより実現されており、開発者がツールを用いて責務割り当てを行っている最中に、より高品質な割り当ての候補があればそれを開発者に推薦する。推薦は、責務割り当ての指針である GRASP パターンに基づいて行われており、パターンの主旨に合わない箇所を不吉な臭いとして検出し、臭いを解

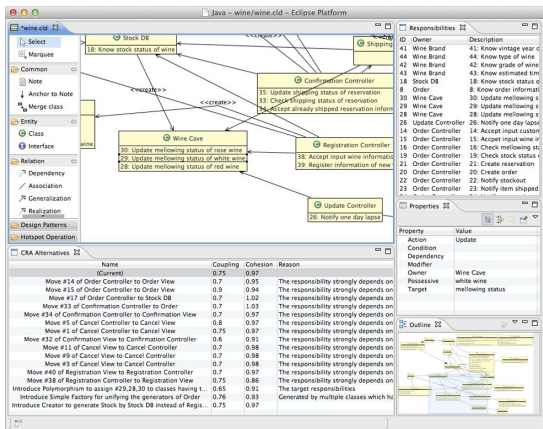


図5 ツール RAST の動作例

消するための変更を設計代替案として提示する。

また、機能実装による変更に伴うプログラムの品質低下を避け、また変更作業自体の効率向上を図るため、変更前にあらかじめリファクタリングを行うプレファクタリング技術について、機能検索とソースコード変換を用いた手法を提案し、オープンソースソフトウェア jEdit の過去の変更に対してその有効性を示した。

5. 主な発表論文等

〔雑誌論文〕(計 14 件)

大森 隆行, 林 晋平, 丸山 勝久: "統合開発環境における細粒度な操作履歴の収集および応用に関する調査". コンピュータソフトウェア, 査読有, 32(1) 60-80. 2015. DOI: 10.11309/jssst.32.1_60

崔 恩滯, 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, 査読有, 32(1) 47-59, 2015. DOI: 10.11309/jssst.32.1_47

星野 大樹, 林 晋平, 佐伯 元司: "ソースコード編集操作の自動グループ化". コンピュータソフトウェア, 査読有, 31(3) 277-283, 2014. DOI: 10.11309/jssst.31.3_277

大森 隆行, 丸山 勝久, 林 晋平, 沢田 篤史: "ソフトウェア進化研究の分類と動向". コンピュータソフトウェア, 査読有, 29(3) 3-28, 2012. DOI: 10.11309/jssst.29.3_3

林 晋平, 関根 克幸, 佐伯 元司: "Feature Location を用いたソフトウェア機能の対話的な実装理解支援". 情報処理学会論文誌, 査読有, 53(2) 578-589, 2012. <http://id.nii.ac.jp/1001/00080669/>

Shinpei Hayashi, Daiki Hoshino, Jumpei Matsuda, Motoshi Saeki, Takayuki Omori, Katsuhisa Maruyama: "Historef: A Tool for Edit History Refactoring". In Proc. 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2015), 査読有, 469-473, 2015. DOI: 10.1109/SANER.2015.7081858

Shinpei Hayashi, Takashi Ishio, Hiroshi Kazato, Tsuyoshi Oshima: "Toward Understanding How Developers Recognize Features in Source Code from Descriptions". In Proceedings of the 9th International Workshop on Advanced Modularization Techniques (AOAsia/Pacific 2014), 査読有, 1-3, 2014. DOI: 10.1145/2666358.2666578

Shinpei Hayashi, Sirinut Thangthumachit, Motoshi Saeki: "REdiffs: Re-

factoring-Aware Difference Viewer for Java". In Proceedings of the 20th Working Conference on Reverse Engineering (WCRE 2013), 査読有, 487-488, 2013. DOI: 10.1109/WCRE.2013.6671331

Takashi Ishio, Shinpei Hayashi, Hiroshi Kazato, Tsuyoshi Oshima: "On the Effectiveness of Accuracy of Automated Feature Location Technique". In Proceedings of the 20th Working Conference on Reverse Engineering (WCRE 2013), 査読有, 381-390, 2013. DOI: 10.1109/WCRE.2013.6671313

Tepei Kato, Shinpei Hayashi, Motoshi Saeki: "Cutting a Method Call Graph for Supporting Feature Location". In Proceedings of the 4th International Workshop on Empirical Software Engineering in Practice (IWESEP 2012), 査読有, 55-57, 2012. DOI: 10.1109/IWESEP.2012.17

Shinpei Hayashi, Takayuki Omori, Teruyoshi Zenmyo, Katsuhisa Maruyama, Motoshi Saeki: "Refactoring Edit History of Source Code". In Proceedings of the 28th IEEE International Conference on Software Maintenance (ICSM 2012), 査読有, 617-620, 2012. DOI: 10.1109/ICSM.2012.6405336

Katsuhisa Maruyama, Eijiro Kitsuo, Takayuki Omori, Shinpei Hayashi: "Slicing and Replaying Code Change History". In Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012), 査読有, 246-249, 2012. DOI: 10.1145/2351676.2351713

Sirinut Thangthumachit, Shinpei Hayashi, Motoshi Saeki: "Understanding Source Code Differences by Separating Refactoring Effects". In Proceedings of the 18th Asia Pacific Software Engineering Conference (APSEC 2011), 査読有, 339-347, 2011. DOI: 10.1109/APSEC.2011.47

Motohiro Akiyama, Shinpei Hayashi, Takashi Kobayashi, Motoshi Saeki: "Supporting Design Model Refactoring for Improving Class Responsibility Assignment". In Proceedings of the ACM/IEEE 14th International Conference on Model Driven Engineering Languages and Systems (MODELS 2011), 査読有, LNCS 6981, 455-469, 2011. DOI: 10.1007/978-3-642-24485-8_33

〔学会発表〕(計 17 件)

星野 大樹, 林 晋平, 佐伯 元司: "ソースコード編集履歴の不吉な臭いの検出". 情報処理学会第 187 回ソフトウェア工学研究発表会, 2015 年 3 月 12 日, 化学会館

(東京).
松田 淳平, 林 晋平, 佐伯 元司: "編集操作履歴の階層的なグループ化を用いたポリシー準拠のコミットの構築". 情報処理学会ソフトウェアエンジニアリングシンポジウム 2014, 2014年9月2日, 芝浦工業大学芝浦キャンパス(東京).

小松田 卓也, 林 晋平, 佐伯 元司: "機能検索結果を用いたプレファクタリング支援". 電子情報通信学会ソフトウェアサイエンス研究会, 2014年7月11日, 富良野文化会館(北海道).

加藤 哲平, 林 晋平, 佐伯 元司: "下位概念間の依存関係を用いた機能検索". 情報処理学会第183回ソフトウェア工学研究発表会, 2014年3月20日, 化学会館(東京).

林 晋平, 丁 斌, 加藤 哲平, 佐伯 元司: "ソースコード差分の対話的修正法". 第20回ソフトウェア工学の基礎ワークショップ, 2013年11月29日, ゆのくに天祥(石川).

星野 大樹, 林 晋平, 佐伯 元司: "ソースコード編集操作の自動グループ化". 第20回ソフトウェア工学の基礎ワークショップ, 2013年11月29日, ゆのくに天祥(石川).

林 晋平, 大森 隆行, 善明 晃由, 丸山 勝久, 佐伯 元司: "ソースコード編集履歴のリファクタリング手法". 第18回ソフトウェア工学の基礎ワークショップ, 2011年11月25日, 海扇閣(青森).

丸山 勝久, 木津 栄二郎, 大森 隆行, 林 晋平: "プログラム理解支援を目的とした編集操作スライスとその再生". 第18回ソフトウェア工学の基礎ワークショップ, 2011年11月25日, 海扇閣(青森).

[その他]

ホームページ等

<http://www.se.cs.titech.ac.jp/~hayashi/>

6. 研究組織

(1) 研究代表者

林 晋平 (HAYASHI SHINPEI)

東京工業大学・大学院情報理工学研究科・助教

研究者番号: 40541975

(2) 研究協力者

佐伯 元司 (SAEKI MOTOSHI)

東京工業大学・大学院情報理工学研究科・教授

研究者番号: 80162254

丸山 勝久 (MARUYAMA KATSUHISA)

立命館大学・情報理工学部・教授

研究者番号: 30330012

大森 隆行 (OMORI TAKAYUKI)
立命館大学・情報理工学部・講師
研究者番号: 90532903