

平成 30 年 5 月 30 日現在

機関番号：34315

研究種目：若手研究(B)

研究期間：2014～2017

課題番号：26730042

研究課題名(和文) 開発者の操作履歴の抽象化に基づくソフトウェア開発支援

研究課題名(英文) Supporting Software Development Based on Abstraction of Developers' Operation Histories

研究代表者

大森 隆行 (Omori, Takayuki)

立命館大学・情報理工学部・任期制講師

研究者番号：90532903

交付決定額(研究期間全体)：(直接経費) 2,100,000円

研究成果の概要(和文)：開発者が統合開発環境(IDE)上で行った操作の履歴を利用した開発支援について研究を行った。代表的な成果を以下に示す。

(1) IDEにおけるコード補完操作の繰り返しに関する実験および既存コード補完ツールの改善、(2) IDEにおける操作履歴の記録と応用に関する手法の調査の実施、(3) コード編集履歴および改版履歴を用いたリファクタリング理解に関する比較実験の実施、(4) 操作履歴に基づく開発タスクの推測手法の考案、(5) 操作履歴付加情報(アノテーション)に基づく操作履歴再生器の機能拡張、(6) コメントを用いたプログラム理解支援手法の検討、(7) 操作履歴に含まれるプライバシー情報のマスキング手法の考案。

研究成果の概要(英文)：This study was carried out to development novel technologies, methods, and tools to support software development based on histories of operations that software developers conducted on integrated development environments. Representative outcomes include:

(1) Experiment on repetitive code completion operations on IDEs and improvement of an existing code completion tool, (2) Survey on methods of recording operation histories and their applications, (3) Comparative experiment of understanding past refactorings with code edit histories and revision histories, (4) A method for estimating development tasks using operation histories, (5) Enhancement of an operation replayer based on operation history annotations, (6) Supporting program comprehension with code comments, and (7) Masking privacy-sensitive information within operation histories.

研究分野：ソフトウェア工学

キーワード：ソフトウェア開発支援 ソフトウェア開発環境 ソフトウェア進化 プログラム理解

## 1. 研究開始当初の背景

社会において広くソフトウェアが使用される現状において、ソフトウェアが欠陥を含まないことは重要である。ソフトウェアの欠陥は、保守作業においてプログラムを作り変える際に埋め込まれることが多い。その主な原因として挙げられるのが、既存のプログラムを十分に理解しないまま変更を行うことである。例えば、ソフトウェア変更を行う際、これまでの変更内容を把握せずに変更を行うと、誤りを埋め込む可能性が高い。

本研究では、これまでに、統合開発環境 (IDE) 上で行われた開発者の操作を元に、開発支援を行う手法について研究を進めてきた。その中で、開発者が行った操作を記録する OperationRecorder、記録された操作を再生し、過去の開発作業を分析可能とする OperationReplayer を構築した。記録される操作履歴は膨大であり、履歴中の操作を一つ一つ再生することは現実的ではない。このため、操作履歴の理解性を高めるための手法についても提案してきた。

現在までに実現した操作履歴の理解性改善手法の多くは、連続する操作をまとめたり、特定のクラスやメソッドで行われた変更を抽出したりするといったように、操作履歴やプログラムの構造に基づく手法であった。しかしながら、これは必ずしも開発者の直感と一致しない。より理解性を高めるため、開発者の直感に合った適切な形で履歴を表現することが求められている。

開発者が認識している個々の開発作業は「タスク」と呼ばれる。操作履歴とタスクを正しく関連づけることができれば、より素早く履歴の概要を把握できると考えられる。一方、開発作業中断後の再開の際、これまでの作業内容に復帰するのに時間がかかるというタスク復帰の問題が指摘されている。この問題の解決のための開発支援ツールはいまだ実現されていないものの、操作履歴等の時系列情報が有効だとされており、操作履歴を用いたプログラム理解支援手法の適用事例として有望である。以上のように、操作履歴に基づく新たなソフトウェア理解支援手法およびツールの実現は重要な研究テーマとなっている。

## 2. 研究の目的

本課題開始当初の研究目的を述べる。

(1) 操作履歴を用いたタスク情報の自動検出およびその有効性検証

本研究では、記録された操作履歴の特性から、タスクの境界や内容を推測可能とすることを旨とする。この情報を再生器上で利用することで、利用者は見る必要のない操作を飛ばすなどして、効率的に操作履歴の理解を行うことができるようになると考えられる。本研究では、効果的なタスク推測のためには、操

作履歴に関するどのような情報をどのように利用すれば良いかを模索する。また、このような抽象的な情報を利用した履歴再生が過去のソフトウェア変更の理解をどの程度支援できるかを明らかにする。

(2) 操作理解のための履歴表現手法およびその有効性検証

開発者がより効率的かつ容易に過去のソフトウェア変更に関する情報を得られるよう、より優れた履歴表現手法を模索する。操作履歴自体や、それに関する情報をグラフ構造として表現したり、要約を行ったりすることを考えている。これらの表現の適切な生成方法、および、これらがどのようにソフトウェア開発に利用できるかを明らかにする。

## 3. 研究の方法

ここでは、研究方法の具体的内容について述べる。

(1) 操作履歴に基づくタスク推測ツールの実現および評価

タスク推測を自動的に行うツールの構築を目指す。開発者が実際にどのようにタスクを認識しているかを明らかにするため、被験者へのインタビューを行う。

(2) 様々な抽象情報を取り扱うことができる操作履歴再生器の実現

操作履歴再生の際、開発タスクの他、様々な情報に基づき操作をフィルタリング、グループ化するため、OperationReplayer の機能拡張を行う。

(3) 抽象情報を利用した操作履歴再生器の評価

抽象情報を付加した操作履歴の再生の有効性を調査するため、従来手法との比較を行い、ソフトウェア理解の正確さや所要時間に差が生じるかを調べる。

(4) 操作履歴のグラフ表現への変換と評価

本研究でこれまでに実現した操作履歴のグラフ表現を拡張し、さらに多彩な情報を利用可能とする。それらがソフトウェア理解に関してどの程度有効かを明らかにする。

(5) 操作履歴の要約ツールの実現および評価

操作履歴の一部を要約することで、より直接的なソフトウェア理解支援を目指す。この際、上記のタスク情報、履歴のグラフ表現を応用する。

## 4. 研究成果

本研究の研究成果のうち主要なものについて述べる。

(1) IDE におけるコード補完操作の繰り返しに関する実験および既存コード補完ツールの改善

操作履歴がソースコード変更を支援する事例として、コード補完ツールに着目した。我々が収集した操作履歴データを分析することで、同じコード補完が短い時間のうちに

繰り返されることが多いという実験結果を得た。この結果を受けて、Java ソフトウェア開発において広く使用されている IDE である Eclipse のコード補完機能を改善し、最近入力された候補を上位に表示するツール RCC Candidate Sorter を開発した。評価の結果、提案ツールが既存ツールより補完時の候補提示順位の観点で優れていることが示された[雑誌論文 ]。

#### (2) IDE における操作履歴の記録と応用に関する手法の調査の実施

近年、IDE 上で行われた開発者の操作履歴を扱う手法が数多く提案されている。そこで、本研究では、主としてソフトウェア進化の研究における引用を想定し、細粒度なソースコード変更を記録する手法、および、それらを応用する手法について文献調査を行った。その成果をサーベイ論文として発表した[雑誌論文 ]。

本サーベイでは、対象となる操作指向の手法を記録手法と応用手法に分け、それぞれが基盤とする IDE や記録手法をまとめた。また、操作指向の手法に共通する特性を整理したり、応用手法を即時性、支援手段、支援対象という観点から分類したりする等、系統的な調査を行った。本サーベイの結果に関しては、海外の研究者からの要望を受け、英語に翻訳し、その内容を公開した[雑誌論文 ]。

#### (3) コード編集履歴および改版履歴を用いたリファクタリング理解に関する比較実験の実施

過去にどのようなリファクタリング操作が行われたかを知ることは、現実のソフトウェア保守において非常に重要である。本研究では、従来から行われてきた改版履歴(版管理システムに格納されたソースコードの差分)と、開発者が行った操作履歴を用いる場合で、リファクタリング操作の理解にどのような違いが生じるかの比較実験を行った[雑誌論文 ]。

結果として、それぞれの場合に特有の、操作の誤解や見落としを生じる要因があることが分かった。例えば、改版履歴を用いる場合、複数の変更が1つのコード変更に含まれることや、ある変更が他の変更を上書きしてしまうこと、変更の順番に関する情報が失われることが操作の誤解を招く要因となり得る。一方で、操作履歴を用いた場合でも、1つの意図を持った操作が編集の編集操作に分かれてしまうことや、IDE の自動編集機能等によって大規模かつ分散したコード変更が1つの操作に集約されてしまうことが誤解の要因となることが判明した。全体的には、操作履歴を再生する場合の方が、変更の見落としを防げる事例が多かった。

#### (4) 操作履歴に基づく開発タスクの推測手法の考案

開発タスクの推測を支援するツールとして、開発セッションごとに開発者が行った操作の要約を提示するツールを構築した。ここで、セッションとは、一定時間以上の無操作時間により区切られた期間のことである。被験者実験の結果、要約の提示がタスクの推測や思い出しに有用なケースがあることや、開発者によって認識するタスクの粒度に大きな差異があることがわかった。要約の提示方法については、表示手法に慣れていない開発者でも即座に使えるよう今後改善が必要である。

#### (5) 操作履歴付加情報(アノテーション)に基づく操作履歴再生器の機能拡張

操作履歴記録ツール OperationRecorder が記録する内容は、開発者が行ったコード変更やツール利用の内容そのものであり、それに付随する静的解析の情報等は保持しない。そこで、操作履歴グラフへの付加情報をアノテーションノードとして履歴に後から付加することができる仕組みを構築した。さらに、操作履歴再生器 OperationReplayer を拡張し、それらの情報を再生時に表示したり、それに基づいて履歴をフィルタリングしたりする機能を実装した。今後、コード変更の理解等においてより有用な情報を抽出し、それらを付加した操作履歴を用いて有用性を確認する必要がある。

#### (6) コメントを用いたプログラム理解支援手法の検討

ソフトウェア開発において、ソースコード中のコメントは様々な使われ方をする。本研究では、特に削除されたコメントに着目し、それらがコード内容との整合性を維持しているにも関わらず削除されている事例を発見した[学会発表 ]。また、コメントが説明している対象(構文要素)の規模が大きいほど、コメントの内容にはより抽象的な情報が含まれ、仮に説明対象のコードが変更されても、コメントの有用性が完全に失われる事例は少ないことが観測された。

コメントの記述内容は人間にとってソースコードよりも直感的に分かりやすいことが多いため、削除されたコメントを含めて、コメントの情報に基づくプログラム理解が有用であると考えられる。現時点までに我々が収集したデータでは、コメントの記述件数が十分ではないと判断されたため、今後、より大規模なデータセットを用いて実験を行い、その結果に基づいて支援手法を構築することが有望であると考えられる。

#### (7) 操作履歴に含まれるプライバシー情報のマスキング手法の考案

開発者が IDE 上で行った操作履歴の情報には、開発者やその知人を特定可能とするプライバシー情報が含まれることがある。操作履歴情報は、操作履歴に基づく支援ツールを開

発チーム内で使用する場合は、履歴を用いた研究における実験結果の根拠とする場合に、複数人で共有、あるいは一般に公開する必要がある。しかしながら、プライバシー情報が含まれる場合、情報の共有や公開が、開発者に予期しない被害をもたらす恐れがある。そこで、本研究では、操作履歴に実際にプライバシー情報が含まれることを確認し、それらの漏洩を防ぐためのマスキング(置換)手法を考案・実装した[学会発表]。現時点では、指定された文字列を履歴から消去するのみにとどまっているため、履歴の利用目的等に応じてより適切な方法でプライバシーを守れるよう、手法を改善する必要がある。

## 5 . 主な発表論文等

[雑誌論文](計6件)

Takayuki Omori, Katsuhisa Maruyama, "Comparative Study between Two Approaches Using Edit Operations and Code Differences to Detect Past Refactorings," IEICE Transactions on Information and Systems, 査読有, Vol.E101-D, No.3, 2018, pp.644-658, DOI: 10.1587/transinf.2017EDP7160

Takayuki Omori, Shinpei Hayashi, Katsuhisa Maruyama, "A Survey on Methods of Recording Fine-grained Operations on Integrated Development Environments and their Applications," Computer Software, 査読有, Vol.32, No.1, 2016, 19 pages, <http://www.ritsumei.ac.jp/~tomori/publication/omori-survey16.pdf>

Katsuhisa Maruyama, Takayuki Omori, Shinpei Hayashi, "Slicing Fine-Grained Code Change History," IEICE Transactions on Information and Systems, 査読有, Vol.E99-D, No.3, 2016, pp.671-687, DOI: 10.1587/transinf.2015EDP7282

Takayuki Omori, Hiroaki Kuwabara, Katsuhisa Maruyama, "Improving code completion based on repetitive code completion operations," Computer Software, 査読有, Vol.32, No.1, 2015, pp.120-135, DOI: 10.11309/jssst.32.1\_120

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、Vol.32, No.1, 2015, pp.60-80, DOI: 10.11309/jssst.32.1\_60

木津 栄二郎、大森 隆行、丸山 勝久、コードの編集履歴を用いたプログラム変更の検出、情報処理学会論文誌、査読有、Vol.56, No.2, 2015, pp.611-626, [https://www.jstage.jst.go.jp/article/jssst/29/2/29\\_2\\_168/\\_pdf/](https://www.jstage.jst.go.jp/article/jssst/29/2/29_2_168/_pdf/)

[学会発表](計7件)

ジョ ヘヨン、大森 隆行、糸賀 裕弥、大西 淳、データフロー要求の変更に伴う波及効果解析、電子情報通信学会知能ソフトウェア工学研究会(KBSE), 2018

Katsuhisa Maruyama, Shinpei Hayashi, Takayuki Omori, "ChangeMacroRecorder: Recording Fine-Grained Textual Changes of Source Code," The 25th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER'18), 2018

Takayuki Omori, "Privacy Preservation in Interaction History on Integrated Development Environments," The 1st International Workshop on Mining and Analyzing Interaction Histories (MAINT'18), 2018

大森 隆行、ソースコード編集操作履歴中のセンシティブな情報のマスキング、ソフトウェア工学の基礎 XXIV 日本ソフトウェア科学会 FOSE2017、2017

Takayuki Omori, "Lost Comments Support Program Comprehension," The 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER'17), 2017

大森 隆行、失われたコメントが語るもの、削除されたコメントに関する調査、ソフトウェア工学の基礎 XXIII 日本ソフトウェア科学会 FOSE2016、2016

Shinpei Hayashi, Daiki Hoshino, Jumpei Matsuda, Motoshi Saeki, Takayuki Omori, Katsuhisa Maruyama, "Historef: A Tool for Edit History Refactoring," The 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER'15), 2015

[その他]

(1)ホームページ

<http://www.ritsumei.ac.jp/~tomori/>

(2)立命館大学研究者学術情報データベース

<http://research-db.ritsumei.ac.jp/Profiles/77/0007601/profile.html>

## 6 . 研究組織

(1)研究代表者

大森 隆行 (OMORI, Takayuki)

立命館大学・情報理工学部・任期制講師  
研究者番号: 90532903

(2)研究協力者

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

林 晋平 (HAYASHI, Shinpei)

東京工業大学・情報理工学院・助教  
研究者番号：40541975