

# ソフトウェア工学講座

## ソフトウェアの高品質化を目指して



研究室紹介

肥後芳樹\*

The Software Engineering Laboratory:  
Towards Improved Quality of Software Systems

Key Words : Software engineering, Source code analysis, Code clone

### はじめに

筆者は2022年8月に大阪大学大学院情報科学研究科教授に着任し、ソフトウェア工学講座という研究室を主宰する立場となった。同年にはOpenAI社がChatGPTを公開しており、その影響は筆者の研究分野であるソフトウェア工学領域においても非常に大きく、本稿の執筆は、自身の研究の方向性を改めて見つめ直す良い機会となった。

著者は学部生時代、決して優等生といえる成績ではなかったが、プログラミングの演習は喜々として取り組んでいた。大学に入学するまでにほとんどプログラミングをした経験が無く、演習ではプログラミングによって自分の書いたコードが完全に自動で実行されるのが面白かった。しかしながら、プログラミングの能力が高かったわけではなく、演習課題ではいつもバグに悩まされていた。年次が上がり配属研究室を選ぶにあたり、著者を魅了したのはソフトウェア工学の研究を行っている井上研究室だった。当時の井上研究室ではデバッグ支援やプログラムの部品検索、重複コードの検出等の研究が行われていた。井上研究室に配属後は重複コード検出の研究に取り組んだ。先生方から頂いた研究テーマが自分にあっており、幸いにも一年目にして国際会議で発表する機会を得て、研究が面白いと感じ博士後期課程に進学した。学位取得後は、同研究科の楠本研究室で助教として採用していただき、その後准教授とな

り合計で約15年間大変お世話になった。その後は上記の井上先生の後を継ぐ形でソフトウェア工学講座の教授として着任した。

大学院生時代から続けていた重複コードに関する研究は助教以降も継続しており、研究の幅を広げるためそれに加えてソフトウェアリポジトリマイニングの研究についても取り組んだ。ソフトウェアリポジトリとはソフトウェアの開発履歴が蓄積されている貯蔵庫のことであり、誰がいつどの部分をどのように変更したのかという記録が全て残っている。この貯蔵庫（ソフトウェアの開発履歴）からお宝となるようなデータを発掘するのがソフトウェアリポジトリマイニングの研究である。また、2014年頃から自動プログラム修正の研究にも取り組みはじめた。自動プログラム修正とはその字句のとおり人間がまったく介在することなくバグの原因箇所の特定およびその修正作業を機械が行うことを意味する。2009年に技術的ブレイクスルーとなった遺伝的プログラミングを利用した自動プログラム修正の論文がソフトウェア工学分野のトップカンファレンスICSEで発表され、自動プログラム修正技術が徐々に注目を集めていた時期であった。

上記のように筆者は重複コード、ソフトウェアリポジトリマイニング、自動プログラム修正の三本柱で研究を行ってきた。研究室を主宰する立場となってからは、同研究室の松下准教授および神田助教と協力し、競技プログラミング関連の研究やソフトウェア開発環境に関する研究も行うようになった。本稿では、近年特に注力している機能等価メソッドデータの構築とその応用について紹介する。

### 機能等価メソッドデータの構築

高品質なソースコードを効率良く実装するための技術の開発および知見を得ることがこの研究の目的である。その目的を達成するために、下記の問いに



\* Yoshiki HIGO

1980年4月生まれ  
大阪大学大学院情報科学研究科コンピュータサイエンス専攻 博士後期課程  
(2007年)

現在、大阪大学大学院情報科学研究科  
コンピュータサイエンス専攻 教授  
博士(情報工学)  
専門/ソフトウェア工学  
TEL : 06-6879-4105  
E-mail : higo@ist.osaka-u.ac.jp

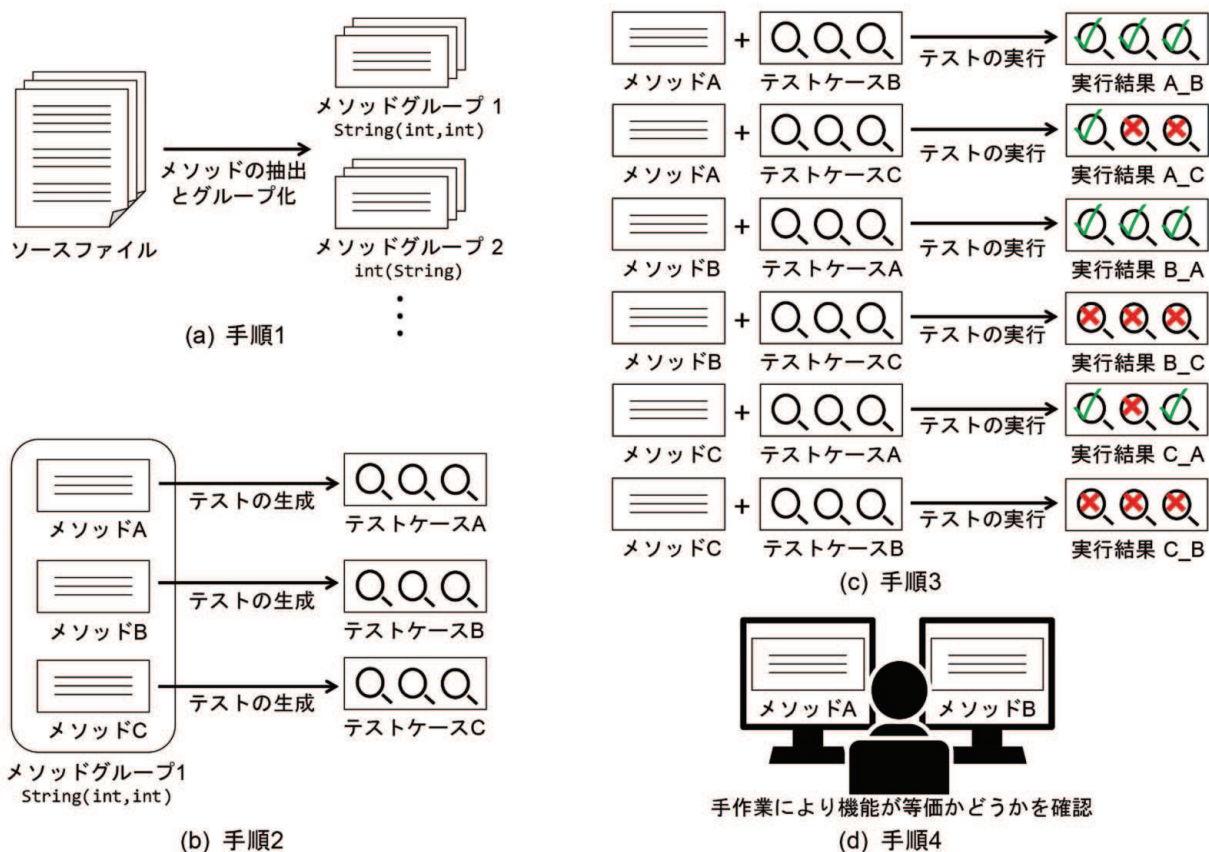


図1 機能等価なJavaメソッドのペアを得るための手順

対する答えを実証的に明らかにしようとするこの研究を行っている。

- どのような実装がソフトウェア品質や実行時性能の面から優れているのか
- 既存の数多くのソフトウェア工学手法はどのような実装に対して効果的に働くのか
- 大規模言語モデルを用いることで、ソフトウェア工学タスクの高度化は可能か

上記の問いを明らかにするため、この研究では同機能異実装ソースコード（同じ機能を持つが実装方法が異なるソースコード）に着目している。例えば、同機能異実装コードはコードクローン検出ツールの評価に利用できる。同じ機能を持つソースコードはコードクローンとして検出されることが望ましいので、同機能異実装ソースコードがどの程度コードクローンとして検出されるのかを調査することで、検出ツールの性能を評価できる。また、同機能異実装ソースコードを用いることで、メモリ使用量や実装速度などの性能面でどのような実装が優れているのか、ISO/IEC25010のようなソフトウェア品質面で

どのような実装が優れているのかを調査できる。

しかし、オープンソースソフトウェアにおいて同機能異実装ソースコードは大量に存在しているが、それらの収集は容易ではない。既存のコードクローン検出ツールを使うと、そのツールで検出可能な同機能異実装ソースコードのみが検出されてしまう。そのため、機能等価であるがコードクローンとして検出されなかったソースコードはコードクローン検出ツールの評価に使うことができず、コードクローン検出技術の評価を適切に行えない。

そこでこの研究では、収集対象とする同機能異実装ソースコードを「同じ入力（引数）を与えたときに同じ出力（返値）を返すメソッド（以降、機能等価メソッド）のペア」に限定する。同機能異実装ソースコードの検出を、機能等価メソッドのペアの検出に限定することにより、テスト自動生成技術を用いて機能等価メソッドのペアの候補を自動的に取得することが、この研究において大規模なデータセットを効率良く構築するためのキーアイデアである。取得した機能等価メソッドのペアの候補は手作業に

より真に機能等価であるかを確認する。

現時点では Java に対してはデータセットの構築は終わっている (将来的にはもう少し規模を拡大するかもしれない)。また, Python については構築中である。データセットの構築手順についても簡単に紹介する。本研究では下記の手順でデータセットの構築を行っている。

- 手順1: 対象ソースコードからメソッドの抽出および分類
- 手順2: 分類した各メソッドからテストケースを生成
- 手順3: 生成したテストを相互実行することにより機能等価メソッドペアの候補を取得
- 手順4: 機能等価メソッドペアの各候補のソースコードを閲覧し, 真に機能等価であるかを判定

図1は上記の手順の概要を表している。各手順について簡単に説明する。

**【手順1】** 対象ソースコードを解析してメソッドを抽出する。抽出したメソッドは, 返値の型と引数の型に基づいて分類する。それらの型が同じメソッドが同じグループに分類される。

**【手順2】** 分類した各メソッドからテストケースを自動生成する。現在のテスト自動生成技術では十分にテストを自動生成できないこともあるため, 生成されたテストの個数やカバレッジ等を利用して十分にテストが生成されたと判断できる場合のみ, そのメソッドを手順3で利用する。

**【手順3】** 同じグループに属し, テストケースを十分に生成できた各メソッドに対して, 相互にテストを実行する。図1(c)では, Method-A, Method-B, Method-C の3つのメソッドに対してテストを相互に実行している。

Method-A は Method-B から生成された全てのテストに成功し, Method-B も Method-A から生成された全てのテストに成功している。したがって, Method-A と Method-B は機能等価なメソッドペアの候補となる。Method-C は Method-A から生成されたテストの1つを成功しておらず, Method-B から生成された全てのテストに成功していない。したがって, Method-A と Method-C のペア, および Method-B と Method-C のペアは, 機能等価なメソッドペアの候補にならない。手順3では, テスト

自動生成技術は対象のメソッドに対して必ず成功するテストケースのみを生成するという性質を用いている。つまり, 図1の例であれば, Method-A は Method-A から生成された全てのテストケースに必ず成功し, Method-B と Method-C も同様である。相互のテスト実行によりある程度のメソッドの機能等価性を自動判定できる。

**【手順4】** 手順3で得られた機能等価なメソッドペアのソースコードを目視で調査し, 真に機能等価メソッドのペアかどうかを判断する。手順4はデータセットの質を担保するため (誤判定をできる限り避けるため), 複数人で行う。

構築した Java のデータセットは [1][2] で公開している。また構築したデータセットを大規模言語モデルを利用したコードクローン検出ツールのファインチューニング用データとして利用することにより, 検出の精度が上がることを確認済みである [3]。

## おわりに

本稿では, 現在行っている主要な研究テーマである機能等価メソッドデータの構築について触れた。本研究は 2024 年度から科学研究費 基盤研究 (A) の助成を受けることになっており, 実際に社会で役立つ技術を創出すべく, 力強く推進していく。また当研究室にはこれ以外にもさまざまな研究を行っており, 我々の研究室のホームページ [4] でその研究成果の論文を公開しているので, 興味のある方は是非ご覧いただきたい。

## 参考文献

- 1) Yoshiki Higo : Dataset of Functionally Equivalent Java Methods and Its Application to Evaluating Clone Detection Tools, accepted by IEICE Transactions on Information and Systems, Vol.E107-D, No.6 (2024)
- 2) FEMPDataset, <https://github.com/YoshikiHigo/FEMPDataset>
- 3) 井上龍太郎, 肥後芳樹: 機能等価メソッドデータセットを利用した LLM によるコードクローン検出の精度向上, 電子情報通信学会技術研究報告, Vol. 123, No. 414 (2024)
- 4) 大阪大学 大学院情報科学研究科 肥後研究室, <https://sel.ist.osaka-u.ac.jp/>