

# 修士論文

## ソースコードの類似性に基づいた テストコード自動推薦ツール

倉地 亮介

2020 年 1 月 28 日

奈良先端科学技術大学院大学  
先端科学技術研究科 情報科学領域

本論文は奈良先端科学技術大学院大学先端科学技術研究科情報科学領域に  
修士(工学) 授与の要件として提出した修士論文である。

倉地 亮介

審査委員：

飯田 元 教授	(主指導教員)
井上 美智子 教授	(副指導教員)
市川 晃平 准教授	(副指導教員)
崔 恩瀨 准教授	(京都工芸繊維大学)

# ソースコードの類似性に基づいた テストコード自動推薦ツール\*

倉地 亮介

## 内容梗概

ソフトウェアの品質確保の要と言えるソフトウェアテストを支援することは重要である。これまでに、テスト作成コストを削減するために様々な自動生成技術が提案されてきた。しかし、自動生成されたテストコードはテスト対象コードの作成経緯や意図に基づいて生成されていないという性質から後のメンテナンス活動を困難にさせる課題がある。この課題の解決方法として、既存テストの再利用が有効であると考えられる。そこで、本研究ではOSSプロジェクト上に存在する既存の品質の高いテストコード推薦するツール SuiteRec を提案する。SuiteRec は、類似コード検索ツールを用いてクローンペア間でのテスト再利用を考える。開発者からの入力コードに対して類似コードを検出し、その類似コードに対応するテストスイートを開発者に推薦する。さらに、テストコードの良くない実装を表す指標を示すテストスメルを開発者に提示し、より品質の高いテストスイートを推薦できるように推薦順位を並び替える。提案ツールの評価では、被験者によって SuiteRec の使用した場合とそうでない場合でテストコードの作成してもらい、テスト作成をどの程度支援できるかを定量的および定性的に評価した。その結果、SuiteRec を利用した場合、(1) 条件分岐が多いプログラムのテストコードを作成する際にコードカバレッジの向上に効果的であること、(2) 作成したテストコードはテストスメルの数が少なく品質が高いこと、(3) 開発者はテストの作成を容易だと認識し、自身で作成したテストコードに自信が持てることが分かった。

---

\*奈良先端科学技術大学院大学 先端科学技術研究科 情報科学領域 修士論文, 2020 年 1 月 28 日.

キーワード

類似コード検出, 推薦システム, ソフトウェアテスト, 単体テスト

# Automatic Test Suite Recommendation System based on Code Clone Detection\*

Ryosuke Kurachi

## Abstract

Automatically generated tests tend to be less read-able and maintainable since they often do not consider the latent objective of the target code. Reusing existing tests might help address this problem. To this end, we present SuiteRec, a system that recommends reusable test suites based on code clone detection. Given a Java method, SuiteRec searches for its code clones from a code base collected from open-source projects, and then recommends test suites of the clones. It also provides the ranking of the recommended test suites computed based on the similarity between the input code and the cloned code. We evaluate SuiteRec with a human study of ten students. The results indicate that SuiteRec successfully recommends reusable test suites.

## Keywords:

clone detection, recommendation system, software testing, unit test

---

\*Master's Thesis, Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, January 28, 2020.

# 目次

1. はじめに	1
2. 背景	3
2.1 ソフトウェアテスト . . . . .	3
2.2 テストコード自動生成技術 . . . . .	4
2.3 過去における研究 . . . . .	4
2.4 研究の目的と意義 . . . . .	5
3. 現状と今後の課題	10
謝辞	11
参考文献	12
付録	13
A. おまけその1	13
B. おまけその2	13

## 図 目 次

1	テストにおけるタスク . . . . .	3
2	これは図の例 . . . . .	5
3	おまけの図 . . . . .	13

## 表 目 次

1	Six Relevant Test Smells for Reusing Test Suites. . . . .	4
2	これは表の例 . . . . .	5

## 1. はじめに

近年，ソフトウェアに求められる要件が高度化・多様化する一方，ユーザからはソフトウェアの品質確保やコスト削減に対する要求も増加している [1]．その中でもソフトウェア開発全体のコストに占める割合が大きく，品質確保の要ともいえるソフトウェアテストを支援する技術への関心が高まっている [?]．しかし，現状ではテスト作成作業の大部分が人手で行われており，多くのテストを作成しようとするするとそれに比例してコストも増加してしまう．このような背景から，ソフトウェアの品質を確保しつつコスト削減を達成するために，様々な自動化技術が提案されている [?],[?],[?],[?],[?]．

既存研究で提案されている EvoSuite[?] は，単体テスト自動生成における最先端のツールである．EvoSuite は，対象コードを静的解析しプログラムを記号値で表現する．そして，対象コードの制御パスを通るような条件を集め，条件を満たす具体値を生成する．単体テストを自動生成することで，開発者は手作業での作成時間が自動生成によって節約することができ，またコードカバレッジを向上することができる．しかし，既存ツールによって自動生成されるテストコードは対象のコードの作成経緯や意図に基づいて生成されていないという性質から可読性が低く開発者に信用されていないことや後の保守作業を困難にするという課題がある [?],[?],[?]．このことは、自動生成ツールの実用的な利用の価値に疑問を提示させる．テストが失敗するたびに，開発者はテスト対象のプログラム内での不具合を原因を特定するまたは，テスト自体を更新する必要があるかどうかを判断する必要がある．自動生成されたテストは，自動生成によって得られる時間の節約よりも読みづらく，保守作業に助けになるというよりかむしろ邪魔するという結果が報告されている [?]．

我々は，この課題の解決するために既存テストの再利用が有効であると考ええる．本研究では，OSS に存在する既存の品質の高いテストコード推薦するツール SuiteRec を提案する．推薦手法の基本となるアイデアは類似コード間でのテストコード再利用である．SuiteRec は，入力コードに対して類似コードを検出し，その類似コードに対応するテストスイートを開発者に推薦する．さらに，テストコードの良くない実装を表す指標であるテストスメルを開発者に提示し，より品



質の高いテストスイートを推薦できるように推薦順位がランキングされる。

提案ツールの評価では、被験者によって SuiteRec の使用した場合とそうでない場合でテストコードの作成してもらい、テスト作成をどの程度支援できるかを定量的および定性的に評価した。その結果、SuiteRec の利用は条件分岐が多く複雑なプログラムのテストコードを作成する際にコードカバレッジの向上に効果的であること、作成したテストコードの内のテストスメル数が少なく品質が高いことが分かった。また、実験後のアンケートによる定性的な評価では、SuiteRec を使用した場合被験者はテストコードの作成が容易になると認識し、また自分の作成したコードに自信が持てることが分かった。

## 2. 背景

### 2.1 ソフトウェアテスト

ソフトウェアテスト(以下, テスト)とは, ソフトウェア開発プロセスの後半に実施される工程であり, 品質確保における最後の砦である。テストは, ソフトウェアが仕様書通りに動作することを確認すること, また不具合を検出し修正することでソフトウェアの品質を向上させることを目的として行われる。テストは図 [1] で示すように, テスト計画, テスト設計, テスト実行, テスト管理という大きく 4 つのタスクで構成される。テスト計画タスクでは, 開発全体の計画に基づき, テスト対象, スケジュール, 各タスクの実施体制・リソース配分等の策定を行う。テスト設計タスクでは, 設計書などソフトウェアの仕様が記述されたドキュメント等を基に, テストケースを作成する。テスト実行タスクでは, ソフトウェアを動作させ, それぞれのテストケースにおいてソフトウェアが期待通りの振る舞いをするかどうかを確認する。テスト管理タスクでは, テストの消化状況やソフトウェアの品質状況の確認を随時行い, テスト優先度やリソース見直しなどのアクションを行う。テスト工程のコスト削減のため, テスト実行タスクにおいて, 単体テストでは JUnit, 結合テスト Selenium, Appium 等のテスト自動実行ツールの利用が進んでいる。しかし, テスト設計タスクは未だ手動で行うことが多く, 自動化技術の実用化および普及が期待されている。



図 1 テストにおけるタスク

表 1 テストの種類

テスト粒度	説明
単体テスト	プログラムを構成する比較的小さな単位の部品が個々の機能を正しく果たしているかどうかを検証するテスト
結合テスト	個々の機能を果たすプログラムの部品(単体)を組み合わせて、仕様通り正しく動作するかを検証するテスト
システムテスト	個々の機能や仕組みを総合した全体像のシステムとして、仕様通り正しく動作するかを検証するテスト

## 2.2 テストコード自動生成技術

テストコード自動生成

## 2.3 過去における研究

過去における研究としては [1] などがある。

過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究

過去における研究 過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究 過去における研究

過去における研究 過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究過去における研究 過去における研究 過去における研究 過去における研究

図 2 これは図の例

表 2 これは表の例



目的と意義研究の目的と意義研究の目的と意義

研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の  
目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意  
義研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の  
目的と意義研究の目的と意義研究の目的と意義

研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の  
目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意  
義研究の目的と意義研究の目的と意義研究の目的と意義研究の目的と意義研究の  
目的と意義研究の目的と意義研究の目的と意義







### 3. 現状と今後の課題

現状と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状  
と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後  
の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後の課題  
現状と今後の課題 現状と今後の課題 現状と今後の課題

現状と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状  
と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後  
の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後の課題  
現状と今後の課題 現状と今後の課題 現状と今後の課題

現状と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状  
と今後の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後  
の課題 現状と今後の課題 現状と今後の課題 現状と今後の課題現状と今後の課題  
現状と今後の課題 現状と今後の課題 現状と今後の課題

## 謝辭

Thank you. Thank you.

## 参考文献

- [1] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25(NIPS'12)*, pages 1097–1105, 2012.

これはおまけの図です。

図 3 おまけの図

## 付録

### A. おまけその 1

これはおまけです。これはおまけです。これはおまけです。これはおまけです。  
これはおまけです。これはおまけです。これはおまけです。これはおまけです。  
これはおまけです。これはおまけです。これはおまけです。これはおまけです。  
これはおまけです。これはおまけです。これはおまけです。これはおまけです。

### B. おまけその 2

これもおまけです。これもおまけです。これもおまけです。これもおまけです。  
これもおまけです。これもおまけです。これもおまけです。これもおまけです。  
これもおまけです。これもおまけです。これもおまけです。これもおまけです。  
これもおまけです。これもおまけです。これもおまけです。これもおまけです。