社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

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 SuiteRec

倉地　亮介† 　崔　　恩瀞†† 　飯田　　元†

† 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　〒630–0192 奈良県生駒市高山町 8916–5
†† 京都工芸繊維大学情報工学課程　〒606–8585 京都府京都市左京区松ケ崎橋上町
E-mail: †kurachi.ryosuke.kp0@is.naist.jp, ††echoi@kit.ac.jp, †††iida@itc.naist.jp

あらまし　テスト工程において，テスト作成コストを削減するために様々なテストコード自動生成ツールが提案されてきた．しかし，既存のツールによって生成される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いないという性質から開発者の保守作業を困難にする課題がある．この課題の解決方法として，本研究では OSS プロジェクト上に存在する既存の品質が高いテストコードを推薦するツール SuiteRec を提案する．また，被験者実験を行い SuiteRec の有用性を確認した．

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，テストスメル，単体テスト

SuiteRec: Automatic Test Suite Recommendation System based on Code Clone Detection

Ryosuke KURACHI†, Eunjong CHOI††, and Hajimu IIDA†

† Graduate School of Information Science, Nara Institute of Science and Technology　8916–5, Takayama, Ikoma, Nara, 630–0192,Japan
†† Department of Information Science　Matsugasaki, Sakyo-ku, Kyoto, 606–8585 Japan
E-mail: †kurachi.ryosuke.kp0@is.naist.jp, ††echoi@kit.ac.jp, †††iida@itc.naist.jp

Abstract　Automatically generated tests tend to be less read-able and maintainable since they often do not consider thelatent objective of the target code. Reusing existing tests might help address this problem. To this end, we present SuiteRec, asystem that recommends reusable test suites based on code clonedetection. 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 providesthe 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.

Key words　clone detection, recommendation system, software testing, test smell, unit test

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 [?]．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テスト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大幅に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後の保守作業を困難にするという課題がある [?], [?], [?]．これは，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コード内

で不具合の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コードは，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，開発者の助けになるというよりかむしろ困難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ィアはクローンペア間でのテストコード再利用である．SuiteRec は，入力コード片に対して類似コード片を検出し，その類似コード片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．

評価実験では，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に含まれるテストスメルの数が少なく品質が高いことが分かった．また，SuiteRec は開発者が参考にしたいテストコードを上位に推薦できることを確認した．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．

以降，2 章では，本研究に関わるコードクローン及びソフトウェアテストについての背景を説明する．3 章では，本研究で提案するテストコード自動推薦ツール SuiteRec ついて説明する．4 章では，被験者による SuiteRec の評価実験について説明する．5 章では，SuiteRec の有効性と妥当性への脅威について議論する．最後に，6 章でまとめと今後の課題について説明する．

2. 背　　景

2.1 テストスメル

本節では，テストコードの品質を定量的に測定するための基準であるテストスメルについて説明する．

テストスメルとは，テストコードの良くない実装を示す指標である．プロダクションコードの設計だけでなく，テストコードを適切に設計することの重要性は元々 Beck ら [?] によって提唱された．さらに，Deursen ら [?] は 11 種類のテストスメルのカタログ，すなわちテストコードの良くない設計を表す実装とそれらを除去するためのリファクタリング手法を定義した．このカタログはそれ以降，19 個の新しいテストスメルを定義した Meszaros [?] によってより拡張された．

以下に本研究で扱う 6 種類のテストスメルを説明する．

Assetion Roulette: 1 つのテストメソッド内に複数の assert 文が存在するテストコード. 各 assert 文は異なる条件をテストするが，テストが失敗した場合開発者へ各 assert 文のエラーメッセージは提供されないので，失敗を特定することが困難になる．

Conditional Test Logic: テストメソッド内に複数の制御文が含まれているテストスイート．テスト成功・失敗は制御フロー内にある assert 文に基づくの予測するのが難しい．

Default Test: JUnit などのテスティングフレームワークを使用したテストコードの内，テストクラスやテストメソッドの名前がデフォルトの状態であるテストコード．テストコードの可読性の向上ために適切な名前に変更する必要がある．

Eager Test: テスト対象クラス内の複数のメソッドを呼び出しているテストコード．1 つのテストメソッド内で複数のメソッド呼び出しを行うと，正確に何をテストしているかについて混乱が生じる．

Exception Handling: テストメソッド内で例外処理が含まれているまたは例外を投げるテストコード．例外処理はプロダクションコードに記述し，テストコード内で例外処理が正しく行われるかどうかを確かめるようにリファクタリングする必要がある．

Mystery Guest: テストメソッド内で，外部リソースを利用するテストコード．テストメソッド内だけで完結せず外部のファイルなど，外部リソースを利用すると外部との依存関係が生じ，外部リソースが壊れた場合テストも失敗してしまう．

2.2 テストコード自動生成技術

3. SuiteRec: テストコード自動推薦ツール

本章では，本研究で提案するコードクローン検出技術を用いたテストコード自動推薦ツール SuiteRec について述べる．本研究では，コードクローン検出技術を利用し，オープンソースソフトウェア (以下，OSS) 上に存在する既存の高品質なテストコード推薦することで，開発者のテストコード作成を支援することが目的である．SuiteRec の基となるアイディアは，クローンペア間でのテストコード再利用である．SuiteRec は，開発者からの関数単位の入力コード片に対してその類似コード片を検出する．そして，類似コード片に対応するテストスイートを開発者に推薦する．さらに，推薦されるテストスイート内に含まれるテストスメルを提示し，より品質が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．

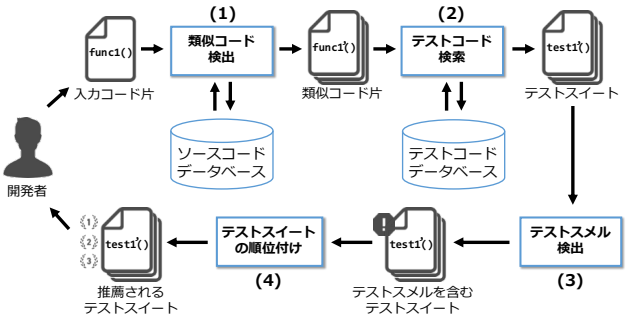

図 1　Overview of SuiteRec.

(1)　SuiteRec は，入力されたコード片を受け取ると，その

コード片をコードクローン検索ツールにかけ入力コード片の類似コードを検出する．

(2) 複数の類似コード片が検出されると，次にその類似コード片に対応するテストスイートをテストコードリポジトリ内から検索する．

(3) 各類似コード片のテストスイートが検出されると，次にそれらをテストスメル検出ツールにかけ各テストスイートに含まれるテストスメルを検出する．

(4) 最後に，(1) で得られた類似コードと入力コードの類似度と (3) で検出されたテストスメルの数を基に出力されるテストスイートの順番がランキングされる．

3.1 Step1: 類似コード片の検出

3.2 Step2: テストコードの検索

3.3 Step3: テストスメルの検出

3.4 Step4: 推薦されるテストスイートの順位付け

4. 評 価 実 験

5. 議　　論

6. まとめと今後の課題

文　　献

[1] D.E. クヌース，改訂新版 TEX ブック，アスキー出版局，東京，1992.
[2] 磯崎秀樹，LATEX 自由自在，サイエンス社，東京，1992.
[3] S. von Bechtolsheim, TEX in Practice, Springer-Verlag, New York, 1993.
[4] 藤田眞作，化学者・生化学者のための LATEX—パソコンによる論文作成の手引，東京化学同人，東京，1993.
[5] 阿瀬はる美，てくてく TEX，アスキー出版局，東京，1994.
[6] N. Walsh, Making TEX Work, O'Reilly & Associates, Sebastopol, 1994.
[7] D. Salomon, The Advanced TEX book, Springer-Verlag, New York, 1995.
[8] 藤田眞作，LATEX マクロの八衢，アジソン・ウェスレイ・パブリッシャーズ・ジャパン，東京，1995.
[9] 中野賢，日本語 LATEX 2ε ブック，アスキー出版局，東京，1996.
[10] 藤田眞作，LATEX 2ε 階梯，アジソン・ウェスレイ・パブリッシャーズ・ジャパン，東京，1996.
[11] 乙部巌己, 江口庄英, pLATEX 2ε for Windows Another Manual, ソフトバンク パブリッシング，東京，1996–1997.
[12] ポール W. エイブラハム，明快 TEX，アジソン・ウェスレイ・パブリッシャーズ・ジャパン，東京，1997.
[13] 江口庄英，Ghostscript Another Manual，ソフトバンク パブリッシング，東京，1997.
[14] マイケル・グーセンス，フランク・ミッテルバッハ，アレキサンダー・サマリン，LATEX コンパニオン，アスキー出版局，東京，1998.
[15] ビクター・エイコー，TEX by Topic—TEX をよく深く知るための 39 章，アスキー出版局，東京，1999.
[16] レスリー・ランポート，文書処理システム LATEX 2ε，ピアソンエデュケーション，東京，1999.
[17] 奥村晴彦，[改訂版] LATEX 2ε 美文書作成入門，技術評論社，東京，2000.
[18] マイケル・グーセンス，セバスチャン・ラッツ，フランク・ミッテルバッハ，LATEX グラフィックスコンパニオン，アスキー出版局，東京，2000.
[19] マイケル・グーセンス，セバスチャン・ラッツ，LATEX Web コンパニオン—TEX と HTML/XML の統合，アスキー出版局，東京，2001.
[20] ページ・エンタープライゼス(株)，LATEX 2ε マクロ & クラスプログラミング基礎解説，技術評論社，東京，2002.
[21] 藤田眞作，LATEX 2ε コマンドブック，ソフトバンク パブリッシング，東京，2003.
[22] 吉永徹美，LATEX 2ε マクロ & クラスプログラミング実践解説，技術評論社，東京，2003.
[23] https://oku.edu.mie-u.ac.jp/~okumura/texwiki/

付　　録

1. PDF の作成方法と A4 用紙への出力

- PDF に書き出すには二通りの方法があります．

（1） dvipdfmx を使って PDF に変換する．

```
dvipdfmx -p a4 -x 1in -y 1in -o file.pdf file.dvi
```

オプションの `-p a4 -x 1in -y 1in` は省略できます．

（2） まず，dvips を使用して，ps に書き出します（以下では段幅の関係で折り返します）．

```
dvips -Pprinter -t a4 -O 0in,0in
 -o file.ps file.dvi
```

`printer` には，使用するプリンタ名を記述します．オプションの `-t a4 -O 0in,0in` は省略できます．

次に Acrobat Distiller で PDF に変換します．

- `dvips` を使用して A4 用紙に出力する場合のパラメータはおおよそ以下のような設定になります．

```
dvips -Pprinter -t a4 -O 0in,0in file.dvi
```

`printer` には使用するプリンタ名を記述します．オプションの `-t a4 -O 0in,0in` は省略できます．

2. 削除したコマンド

本誌の体裁に必要のないコマンドは削除しています．削除したコマンドは，`\part`，`\theindex`，`\tableofcontents`，`\titlepage`，ページスタイルを変更するオプション（`headings`，`myheadings`）などです．