類似コード検出ツールを用いたテストコード再利用に向けた調査

Investigation for Test Code Reuse Using Similar Code Detection Tool

倉地 亮介 * 崔 恩瀞 † 飯田 元 ‡

あらまし 本研究では，既存のテストコードの再利用によるテストコード生成環境の実現に向けて，類似するプロダクトコードのペアを分類し，類似コードペア間とテストコードペア間の類似度の関係を調査した．

1 はじめに

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という課題がある [1]．この課題の解決方法として既存テストの再利用によるテストコード生成を行う環境が有効であると考えられる．

提案環境では，類似コード検出ツールを用いてテスト生成対象のコード片 a に対して類似した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コードを再利用することでコード片のテストコードを生成する．

既存テストの再利用は，命名規則に従った保守性の高いテストコードの生成が期待できる．一方で，類似コード間でのテスト再利用は，適用対象の類似コードペアが存在する必要があることや，テスト対象となる類似コードペア間の関係に依存するので困難な作業である．そのため, どのようなテストコードが類似コード間で再利用できるのかを明らかにすることがテストコードの再利用支援において重要である.

本研究は提案環境の開発に向けてプロジェクト内の類似コードペアをテストコードの有無によって分類し,「両方のコード片にテストコードが存在する類似コードペア」を対象に類似コードペア間とテストコードペア間の類似度の関係を調査した．

2 調査概要

2.1 調査方法

類似コードペアをテストコードの有無によって分類し，類似コードペア間とテストコードペア間の類似度の関係を調査する．調査手法はまず，OSS 上の有名な 3 つの Java プロジェクト内のテストコードとプロダクトコードをそれぞれ抽出し，類似コード検出ツール NICAD を用いて類似コードペアを検出した [2]．次に，類似コードペアのテストコードを特定し，テストコードの有無によって類似コードペアを分類した．そして「両方のコード片にテストコードが存在する類似コードペア」を対象にソースコードの類似度を差異の度合いを表すタイプ [3] によって計算した.

表 1 既存 Java プロジェクト中の類似コードペアの分類結果

両方のコード片にテストコードが存在する類似コードペア		どちらか片方のコード片にテストコードが存在する類似コードペア		テストコードが存在しない類似コードペア		合計
数	割合 (%)	数	割合 (%)	数	割合 (%)	数
62	4.9	334	26.2	879	68.9	1275

*Ryosuke Kurachi, 奈良先端科学技術大学院大学

†Eunjong Choi, 京都工芸繊維大学

‡Hajimu Iida, 奈良先端科学技術大学院大学

表 2　類似コードペア間の類似度とテストコードペア間の類似度の関係

	テストコードペア間の類似度			
類似コードペア間の類似度		タイプ 2	タイプ 3	Not Similar
	タイプ 2	77	10	6
	タイプ 3	28	5	10
	Not Similar	0	0	17

2.2　調査結果

3 つの Java プロジェクトを対象にテストコードの有無によって類似コードペアの分類を行った結果を表 1 に示す．この表からプロジェクト中のテスト対象となる類似コードペアの内，26.2%が既存テストを再利用できる可能性があることが分かる．すなわち，類似コードペア内で片方のコード片にはテストコードがあるにもかかわらず，もう片方のコード片はテストされていない類似コードペアが全体の 4 分の 1 以上を占めており，提案環境の実現によって多くのコード片にテストコードを再利用できる可能性を示している．　類似度の関係調査では，類似コードペアの分類によって得られた「両方のコード片にテストコードが存在する類似コードペア」62 個とそれに対応するテストコードのペア 153 個の類似度の関係を表 2 に示した．調査の結果，テスト対象となる類似コードペアが類似していない (Not Similar) 場合は，類似するテストコードペアが存在しないことが分かる．また，類似コードペアの類似度がタイプ 2，3 の場合は，テストコードの類似度もタイプ 2，3 が多い結果となった．この結果は，テストコードペアの間の類似度と対象の類似コードペア間の類似度には相関関係があり，類似コードペア間の類似度が高いほどテストコードを再利用できる可能性が高いことを示している．　一方で，類似コードペア間の類似度がタイプ 2 と高いにもかかわらず，テストコードペアが類似しない組み合わせが 6 件検出された．これらの類似コードペアのメソッドを確認したところ，同じ制御構造を持つが，最後に出力する数値の選択だけが異なっていることが分かった．検出された例として，Apache kafka では，同一の制御構造で特定のオブジェクトを取得した後，get メソッドでデータを取得する処理と，delete メソッドでデータを削除する処理が類似コードペアとなっていた．このように共通のデータを使用し，互いに関係した処理であっても，異なる処理を実行している場合は，テストコードを再利用することは難しいと考えられる．そのため，類似コードペア間の振る舞いに着目して分類を行い更なる調査をする必要がある．

3　まとめ

本研究では，プロジェクト内の類似コードペアを分類し，類似コードペア間とテストコードペア間の類似度の関係を調査した．その結果，類似度の関係には相関があることを明らかにした．また，調査結果から類似コードペア間のタイプによる類似度だけでなく，振る舞いにも着目する必要があることが分かった．今後は，既存のメソッド呼び出しの差異に基づく類似コードペアの分類手法を用いて，類似コードペア間の振る舞いと対応するテストコードの関係を調査をする予定である．

謝辞　本研究は JSPS 科研費 JP18H04094, 19K20240 の助成を受けた．

参考文献

[1] S. Shamshiri, et al. How Do Automatically Generated Unit Tests Influence Software Maintenance?. *Proc. of ICST*, pp.239–249, 2018.
[2] C. K. Roy, et al. NICAD: Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization. *Proc. of ICPC*, pp.172–181, 2008.
[3] C. K. Roy, et al. Comparison and evaluation of code clone detection techniques and tools:

An example of use for fose2019.cls

a qualitative approach. Science of Computer Programming, Vol. 74, No. 7, pp. 470–495, 2009.