コピーアンドペーストを対象としたソフトウェア再利用動向の抽出・評価手法の検討

井垣　宏[†1]　大田崇史[†1]　楠本真二[†1]

ソフトウェア開発においてコピーアンドペーストによるソースコードの再利用がよく行われている．開発者がソースコードの再利用を行うためには，既存のソースコードを良く理解したうえで，再利用可能なコード片を特定し，開発中のソースコードに適用する必要がある．本研究では，開発者が行ったコピーアンドペーストによるソースコードの再利用動向抽出・分析手法を提案する．

Record and Analysis for Copy and Paste based Software Reuse

Hiroshi Igaki,[†1] Takafumi Ota[†1] and Shinji Kusumoto [†1]

This paper presents the extraction and analysis method for developers' copy and paste behavior.

1. はじめに

ソフトウェアの再利用はソフトウェア開発の生産性や品質の改善に寄与すると言われている．一方で，再利用の実施には多様なスキルが必要であり，特に本稿で対象とするソースコードの不適切な再利用は保守性に関する問題を引き起こす可能性があるとされている．そのため既存研究において，開発者の再利用動向の調査が行われてきた．例えば，Kim ら[1)] は開発者のコピーアンドペーストを記録・分類し，どのような状況でソースコードの再利用が行われるかを分析している．本研究では，開発者によるコピーアンドペースト履歴だけでなく，再利用が可能であったソースコードとあわせて分析を行うことで，開発者によるソースコード再利用を評価する枠組みを提案する．

2. リアルコピーアンドペーストと潜在的コピーアンドペースト

本研究では，開発者が行ったコピーアンドペーストをリアル C&P と呼ぶ．コピーアンドペーストは開発者がソフトウェア開発中のある時刻にあるコード片をコピーし，ペーストすることで行われる．開発者はコピーアンドペーストにおいて，ペースト直後のコード片を変更し，利用することが多い．そこで我々は開発者が開発者がコピーしたコード片のファイルパスと開始終了行を “real copied code”，ペーストした直後のコード片を含むファイルパスと開始終了行を “temporal real pasted code” とする．さらに，ペースト後に同じ開発者によって行われた直近のコミットに含まれるコード片とそのファイルパス，開始終了行を “real pasted code” と定義する．以上の定義をふまえて，リアル C&P をコピペを行った開発者名，コピーした時刻，real copied code ，ペーストした時刻，real pasted code 及び real copied code と real pasted code 間の類似度より構成されるものと定義する．

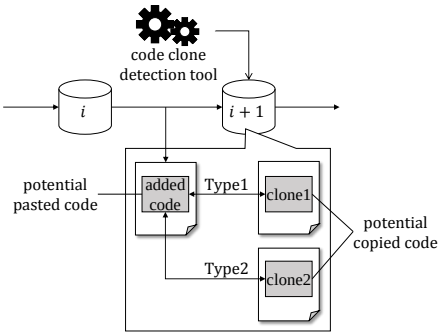

図 1　潜在的コピーアンドペーストの抽出

開発者によって行われたリポジトリ中の各コミットに含まれるコード片には，開発者が一から実装したものやコピーアンドペーストによって実装したものが含まれる．そこで我々は各コミットに含まれるコード片のうち，同一リポジトリ中に含まれるソースコードを再利用することで “潜在的に”C&P によって実装が可能であったコード片を特定し，潜在的 C&P として抽

†1 大阪大学
Osaka University

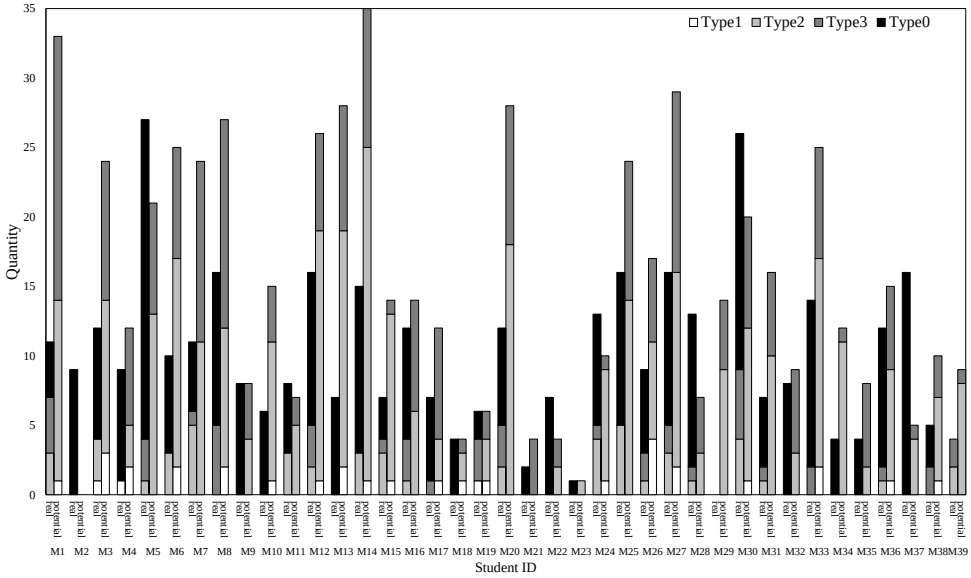

図 2 Extracted real and potential copy and paste

出する手法を提案する．潜在的 C&P の抽出には，図 1 に示すようにコードクローン検出ツールを用いる．Type1∼Type3 のコードクローンを検出することで，そのコミットで開発者が実装したコード片を含むクローンペアを特定する．ここで，検出されたクローンペアに含まれる 2 つのコード片のうち，そのコミットで開発者が実装したコード片に含まれるものを “potential pasted code”，含まれないものを “potential copied code” と呼ぶ．潜在的 C&P は potential pasted code, potential copied code それぞれの開発者，開発日時及びコード間の類似度より構成されるものとする．

リアル C&P 及び潜在的 C&P の持つ類似度はコードクローンの Type1∼Type3 の分類に従う．すなわち，同一のコード片であれば Type1，ユーザ定義名等の正規化後同一と判断される場合は Type2，一定基準値以下の Gap を含む場合は Type3 と分類される．ここで，リアル C&P については，Type1∼Type3 の範囲にないもの（類似度が低い）については，Type0 と分類する．

3. ケーススタディ

大学院生向け講義におけるソフトウェア開発プロジェクトを対象にケーススタディを実施した．プロジェクトは 5,6 人のチームで 4 日間にわたって実施されており，各チーム 5000 行程度実装を行っている．Java, JavaScript 及び Html で開発されており，今回は Java のソースコードのみを対象として抽出・分析を行った．図 2 に全学生のリアル/潜在的 C&P の抽出結果を示す．本ケーススタディにより，Type1 で実装が可能であったにも関わらず，Type2,3 以降，すなわち C&P 後にコード片の不必要な修正を実施している事例が見て取れる．既存のコード片を十分に理解していれば，より適切なコード片の選択が可能であった可能性が高いと考えられる．

4. おわりに

ソースコード再利用の評価を目的として，開発者のリアル C&P と潜在的 C&P を特定し，比較する手法を提案し，ケーススタディを実施した．ケーススタディにより，コードクローン検出ツールで検出不可能なコピーアンドペーストがリアル C&P の半分を占めることやほとんどの学生が不必要な修正を伴うコピーアンドペーストを少なからず行っていることなどが明らかになった．今後は開発者スキルと再利用傾向の比較や authorship 情報を用いた分析等，より詳細な分析を行っていくことを検討している．

参 考 文 献

1) Kim, M., Bergman, L., Lau, T. and Notkin, D.: An ethnographic study of copy and paste programming practices in OOPL, *Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on*, pp.83–92 (2004).