ソフトウェア保守に影響を与えるコードクローン

佐野　由希子[†]

ソフトウェア保守に影響を与える要因の１つとしてコードクローンが挙げられる．コードクローンとは，コピーアンドペーストなどによって生成される類似したコード片のことである．このコードクローンの保守コストが，そうでないコードの保守コストに比べて高いかどうか調査している．

1. はじめに

近年，ソフトウェア保守に影響を与える要因の１つとしてコードクローンが注目されている．コードクローンとは，コピーアンドペーストなどによって生成される類似コード片のことである．一般に，コードクローンはソフトウェアの品質を下げるといわれており，検出手法や集約支援手法の研究が盛んに行われている．しかし，ソフトウェア保守においてコードクローンへ修正が加えられる頻度が高いかどうかを，コード片単位で調べた研究はない．そこで，コードクローンとそうでない部分のどちらに対してより頻繁に修正が行われているかを調査することによって，コードクローンがソフトウェア保守に与える影響を評価しようと研究を行っている．このような立場からアンケートに回答した．

また，ソフトウェア開発経験年数は 3 年ほどである．そして，Java に関わった経験については，大学で 3 か月ほど Java の授業を受けたほか，研究に用いるツールを作成するため，4 か月ほど Java のコーディングを行った．

2. アンケート回答

アンケートへの回答を表 1 に示す．

観点 A　２つのコード片は類似しているので，１つのコードにまとめたほうがよい．

観点 B　２つのコード片は類似しているので，一方にバグがあれば，他方にも同様のバグがあるか調べたほうがよい．

観点 C　２つのコード片は類似しているので，一方のコードを再利用できそうな場所では，かわりに他方を使ってもよい．

観点 D　２つのコード片は類似しているので，類似しているという事実を記録して管理したほうがよい．

観点 X　２つのコード片は，ソフトウェア保守に影響を与えるコードクローンであるといえる．ここで，ソフトウェア保守に影響を与えるコードクローンとしては，観点 B の条件「２つのコード片は類似しているので，一方にバグがあれば，他方にも同様のバグがあるか調べたほうがよい.」を満たすコードクローンを想定している．また，どのような類似コード片をコードクローンと見なすかは検出手法によって異なるが，ここではトークンベースのコードクローン検出手法で検出可能な類似コード片をコードクローンと見なすことにする．この検出手法では，空白やコメント，変数や定数や型などの差異を無視した上で，構造の一致するコード片をコードクローンとして検出する．

表 1　設問への回答

ソース No.	A	B	C	D	X
1	yes	yes	no	yes	yes
2	no	yes	no	yes	yes
3	no	no	no	no	no
4	no	yes	no	yes	yes
5	no	yes	no	yes	yes
6	no	yes	no	yes	yes
7	yes	yes	yes	yes	no
8	no	no	no	no	no
9	no	yes	no	yes	yes
10	yes	yes	yes	yes	no
11	no	yes	no	yes	yes

2.3 ソース No.3 についての回答理由

単なるメソッド宣言の羅列なので，ソフトウェア保守の観点から注目する必要性は低い．また，このよう

† 大阪大学
Osaka University

な類似コード片は偶然の一致で発生する可能性も高い．

2.7　ソース No.7 についての回答理由

意味的にはまったく同じコードであるが，このように一部を別の関数に分けられてしまうと，コードクローンとして検出するのは難しいので，観点 X は no とした．

2.8　ソース No.8 についての回答理由

どちらを使っても最終的な出力結果は同じになるが，計算時間やメモリ使用量が異なるので，用途に合わせて使い分けるべきであろう．

2.10　ソース No.10 についての回答理由

ソース No.7 と同じ理由から，観点 X を no とした．

3. 議　　論

現在，コードクローンがソフトウェア保守に与える影響について研究している．この影響を評価するため，コードクローン部分に対して修正が行われる頻度と，コードクローンでない部分に対して修正が行われる頻度を比較しようと考えている．もしコードクローン部分に対して行われる修正の頻度が高いならば，コードクローンはソフトウェア保守に悪い影響を与えているといえる．逆に，コードクローン部分に対する修正の頻度が低い場合は，コードクローンはソフトウェア保守に良い影響を与えていることになる．また，修正頻度の計測には，バージョン管理ツールとコードクローン検出ツールを用いる計画だ．それらを用いて修正の行われた位置とコードクローンの存在する位置を特定し，修正された箇所がコードクローンであったか否か調べることによって，修正頻度を計測する．その際，どのような類似コード片をコードクローンとして検出するか検討中である．コードクローン検出ツールには，トークン単位で構造の一致するコードクローンを検出するものや，プログラム依存グラフを用いることで対応行の順序が入れ替わっているようなコードクローンも検出できるものなどがある．ただし，プログラム依存グラフを用いたものは実行時間が長くなるという欠点もある．コードクローン検出ツールは，検出の精度と実行時間の差を考慮して選択すべきだろう．また，コードクローンの検出をトークンベースで行った場合と，プログラム依存グラフを用いて行った場合とで，修正頻度についての評価結果を比較するのも興味深いかもしれない．

コードクローンの検出にトークンベースの検出ツールを使用した場合には，再現率が高い代わりに適合率は低くなってしまう．そのため，コードクローンとして検出されたコード片から，どのようにして false positive を除去するかという問題についても検討中である．ここでいう false positive とは，トークン単位の構造が偶然同じだったためにコードクローンとして検出されたもののことである．例えば，アンケートのソース No.3 のような宣言文の羅列も false positive である．その他，関数や代入文などの羅列や，switch 文のように構造が同じになりやすい文なども，false positive として検出されやすい．また，短いコード片については，コピーアンドペーストなどは使わず，独立に書かれたものであっても偶然似てしまうことが多い．宣言文などの羅列や，switch 文が原因となっている false positive の除去に関しては，メトリクス RNR を用いる手法が提案されている[1)]．偶然に似てしまう短いコード片についても，検出するコードクローンの最小サイズを大きくすることで除去することができる．しかし，そうすると検出したいコードクローンまで除去されてしまうので，このようなコード片のフィルタリングは難しいと感じている．そこで，コードクローンの検出結果から，いかに false positive を除去するかについて議論したい．

参　考　文　献

1) 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎. 産学連携に基づいたコードクローン可視化手法の改良と実装. 情報処理学会論文誌, Vol. 48, No. 2, pp. 811–822, Feb 2007.