実現する機能に基づいたソースコードの類似性

田　中　健　介[†]

筆者は，コードクローンを用いてソフトウェアがどの程度重複しているか調査し，ソースコード盗用検出，同様の機能を持つ関数を抽出・グループ化する研究を行っている．ソースコードの類似性とは，そのソースコードが実現している機能の類似性であると考える．

1. はじめに

筆者は現在修士 2 年生で，修士論文に向けて，大量のソースコードからクローンセットを生成する研究を行っている．互いにコードクローンとなっているコード片は同様の機能を持つと考え，これらを収集・整理することにより，ライブラリ開発支援やその他システム開発現場での工数削減が可能であると考えている．

ソフトウェアの開発経験年数は約 3 年，Java に関わった経験は約 2 年，Java との関わり方は研究用のツール作成に用いるのが主である．現在は C 言語で作成されたソフトウェアを対象に研究を行っているが，今後他のプログラミング言語，特に Java で作成されたソフトウェアを適用対象に含めることを考えている．

筆者は，ソースコードの類似とは，そのソースコードが実現する機能の類似であると考え，アンケートに回答する．

2. アンケート回答

筆者は，２つのコード片が同様の機能を実現する場合，あるいは，一方のコード片が実現する機能が他方のコード片が実現する機能に含まれている場合，ライブラリとして再利用可能な形式で１つにまとめるべきではないかと考えている．

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．

表 1　設問への回答

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

2.1 ソース No.1 についての回答理由

クラス名が異なるだけで，機能に差はない．よって，１つのコードにまとめる，あるいは類似したコードであることを記録して管理する必要があると考える．

同じ機能であるからといってどちらを利用しても構わないとは考えない．１つにまとめることが望ましいが，まとめない場合でも一方のみを使用するほうがよいと考える．以降，観点 C に関しては，この考えに従い no とする．

2.2 ソース No.2 についての回答理由

可視性キーワードが異なる場合，カプセル化を意図している場合があると考えられる．そのため，観点 A を no とした．

2.3 ソース No.3 についての回答理由

あるメソッドを定義しているかどうかの違いであり，コード片 3-2 の機能はコード片 3-1 の機能に完全に含まれるので，１つにまとめることが望ましい．

† 大阪大学
Osaka University

2.4　ソース No.4 についての回答理由

コード片 4-2 の機能はコード片 4-1 の機能に完全に含まれるので，１つにまとめることが望ましい．

2.5　ソース No.5 についての回答理由

データ型が異なるが使用方法はまったく同様である．データ型をどちらかに統一し，１つにまとめるのが望ましい．

2.6　ソース No.6 についての回答理由

型を指定しているかどうかの違いのみであり，コード片 6-2 の機能はコード片 6-1 の機能に完全に含まれるので，１つにまとめることが望ましい．

2.7　ソース No.7 についての回答理由

一部の処理がメソッドとして抽出されているのみの変化である．よって，機能はまったく同じである．

2.8　ソース No.8 についての回答理由

入力と出力に関しては同じ扱いをすることができる．しかし，内部のアルゴリズム，ソースコードが異なるので，すべて no とする．

2.9　ソース No.9 についての回答理由

処理内容は同じであるが，扱うデータが異なっている．よって，１つにまとめることには抵抗がある．

2.10　ソース No.10 についての回答理由

No.7 に同じ．

2.11　ソース No.11 についての回答理由

No.9 に同じ．

3. 議　　論

ソースコードはソフトウェアの機能を記述しており，若干異なるソースコードでも同じ機能を実現することができる．つまり，ソースコードの些細な違いを吸収し，同じ機能を持つコード片をいかに検出することができるかが課題である．

筆者は大量のソフトウェアを対象としたクローンセットの生成を目指す研究を行っている．既存のクローン検出ツールを用いることにより自動的にクローンセットを検出することができる．しかし，現在のクローン検出ツールでは一度に検出対象にすることができるソースコードの量に限界があり，また，一度の検出対象範囲内でのみクローンセットを得ることができるため，大量のソースコード全体を対象としたクローンセットを得ることができない．そこで，大量のソースコードを複数のグループに分割し，コードクローンの検出結果を組み合わせることによって，クローンセットの生成を試みている．具体的な方法は，ある検出でコード片 A とコード片 B が，他の検出でコード片 B とコード片 C がクローンペアとして得られた場合，コード片 A，B，C は全て同じクローンセット内のコードクローンとなることを利用した方法である．しかし，コード片 A とコード片 B が，コード片 B’ とコード片 C がコードクローンになっている場合，B と B’ が完全に一致せず僅かに異なっている場合，正しいクローンセットが得られない．そこで関数単位のクローンを検出することでこの問題の回避を試みている．しかし，この方法は２つの関数のソースコード間でコードクローンになっている行の割合に閾値を設けて検出しているため，どの程度機能が類似した関数クローンが得られるのか不安に思っている．

そこで，筆者はソースコードが若干異なっていても，同じ機能を実現しているコード片を検出することが可能であるかどうか，またその方法について議論したい．