

Identifying Code Clones having High Possibilities of Containing Bugs

Author :
Manishankar Mondal
Chanchal K. Roy
Kevin A. Schneider

ICPC '17 Proceedings of the 25th
International Conference on
Program Comprehension
Pages 99-109

発表者:野口耕二郎

目次

- 背景, 目的
- 実験準備
- 実験結果
- まとめ, 所感

背景, 目的

背景

- すべてのクローンにリファクタリングや追跡を行うのは現実的でない
- 優先的にリファクタリングや追跡を行うべきクローンを特定したい

目的

- クローンの変更がクローンのバグの混入のしやすさにどのように関係しているかを調べる
- その関係を調べてどのようなクローンに対して優先的にリファクタリングや追跡を行えばよいかを知る

実験準備

リサーチクエスト

- RQ1: 将来バグ修正される可能性があるのは過去に変更されたクローンとそうでないクローンのどちらか
- RQ2: 過去に頻繁に変更されたクローンは将来バグ修正される可能性が高いか
- RQ3: より最近変更されたクローンが将来バグ修正される可能性は高いか
- RQ4: 過去に変更されていないクローンの内将来バグ修正される可能性が高いのはどのようなものか

データセット

TABLE II. SUBJECT SYSTEMS

Systems	Lang.	Domains	LLR	Revs
jEdit	Java	Text Editor	191,804	4000
Freecol	Java	Game	91,626	1950
Carol	Java	Game	25,091	1700
Jabref	Java	Reference Management	45,515	1545
LLR = LOC in the Last Revision		Revs = No. of Revisions		

実験の手順

- リビジョンの抽出
- クローンの検出
- クローン系譜の検出
- クローンに含まれるバグを修正したリビジョンの検出

以降で用いる変数

- BFC: bug-fix commitの略. クローンに対してバグ修正がされたコミット
- BR: BFCのリビジョンの集合
- r : BRの要素

実験結果

RQ1の実験結果

- 過去に変更されたクローンはそうでないクローンよりバグ修正が行われる可能性が高いかどうかを分析
- 用いるメトリクス
 - NCC_r : r に存在し, 過去に変更されたクローン
 - NCN_r : r に存在し, 過去に変更されなかったクローン
 - $NCCB_r$: r に存在し, 過去に変更されてBFCで変更されたクローン
 - $NCNB_r$: r に存在し, 過去に変更されずにBFCで変更されたクローン

RQ1の実験結果

- 以下の割合を右図のように比較

- $Probability1 = \frac{100 \times \sum_{r \in BR} NCCB_r}{\sum_{r \in BR} NCC_r}$

- 過去に変更されたクローンの内BFCで変更された割合

- $Probability2 = \frac{100 \times \sum_{r \in BR} NCNB_r}{\sum_{r \in BR} NCN_r}$

- 過去に変更されなかったクローンの内BFCで変更された割合

- *Probability1*の方が大きいことが分かり、過去に変更されたクローンの方が将来バグ修正が行われる可能性が高いことが分かった

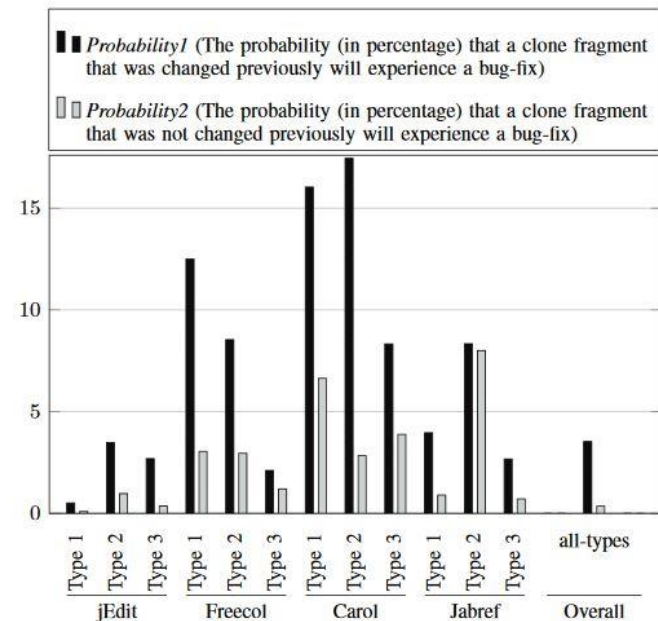


Fig. 2. Comparison of the probabilities that a clone fragment that was changed or not changed previously will experience a bug-fix change.

RQ2の実験結果

- 変更の頻度がバグの発生の一やすさに関係しているかどうかを分析
- 用いるメトリクス
 - $SCCB_r$: r に存在し、過去に変更されてBFCで変更されたクローンの集合
 - $SCCN_r$: r に存在し、過去に変更されてBFCで変更されなかったクローンの集合
 - $AOS-SCCB$: $SCCB_r$ の中でクローンが変更された頻度の平均
 - $AOS-SCCN$: $SCCN_r$ の中でクローンが変更された頻度の平均

RQ2の実験結果

- 右図はACF-SCCBとACF-SCCNを比較したもの

- ACF-SCCBとACF-SCCN のどちらかがどの場合でも大きいという結果が得られなかったことから、クローンの変更の頻度がバグの発生しやすいに関係してないことが分かる

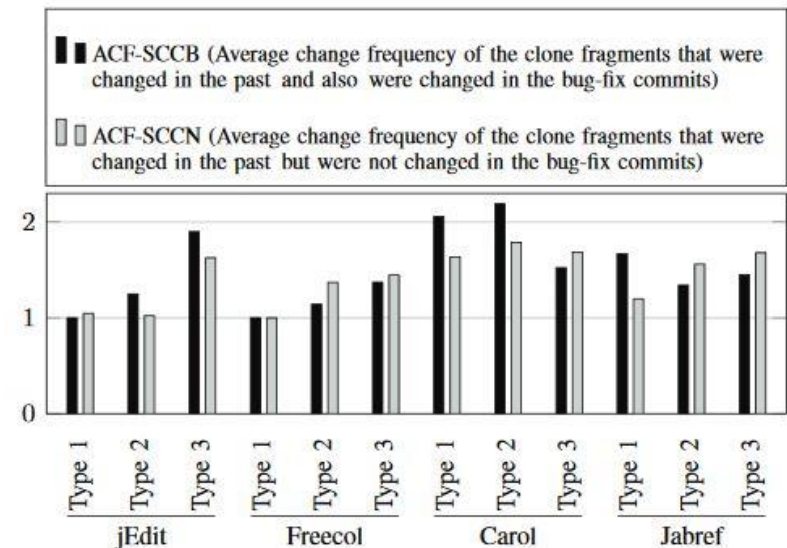


Fig. 3. Comparing the change frequencies of the clone fragments that experienced or did not experience bug-fix changes.

RQ3の実験結果

- クローンがどれだけ最近変更されたかがバグの発生しやすい
さに関係するかどうかを分析する
- 用いるメトリクス
 - $SCCB_r$ と $SCCN_r$ をそのまま用いる
 - $r_{previous}$: r より過去の一歩最近の変更があったリビジョン
 - $OC_{CF} = r - r_{previous}$
 - リビジョン r からクローン片 CF がどれほどの期間変更がなかったかを示す値
 - $AOC-SCCB$: $SCCB_r$ の OC_{CF} の平均
 - $AOC-SCCN$: $SCCN_r$ の OC_{CF} の平均

RQ3の実験結果

- 右図はAOC-SCCBとAOC-SCCNを比較したもの
- 棒グラフの高さが小さいほどクローンがより最近変更されたことになる
- AOC-SCCBがほとんどの場合小さいことが分かり、より最近変更されたクローンがバグ修正される可能性が高いことが分かる

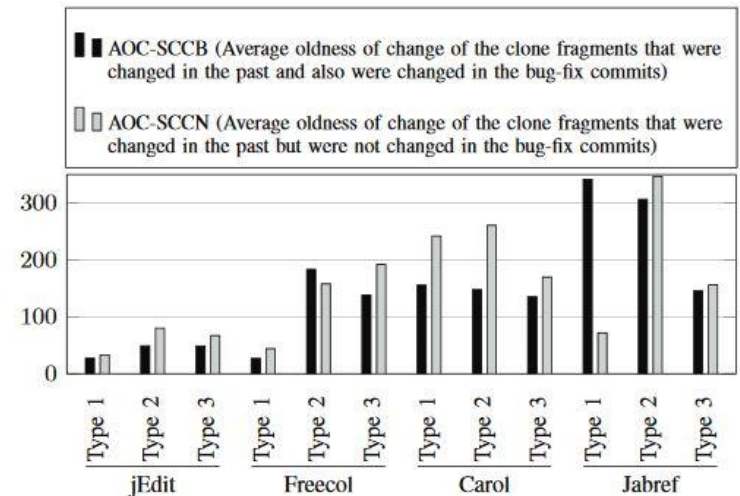


Fig. 4. Comparing the change recency of the clone fragments that experienced or did not experience bug-fix changes.

RQ4の実験結果

- 過去に変更されていないクローンの内最近作成されたクローンがバグ修正される可能性が高いかどうかを分析
- 用いるメトリクス
 - $SCNB_r$: r に存在し, 過去に変更されずにBFCで変更されたクローンの集合
 - $SCNN_r$: r に存在し, 過去にもBFCでも変更されなかったクローンの集合
 - $r_{created}$: クローン片が作成されたリビジョン
 - $OCR_{CF} = r - r_{created}$
 - リビジョン r よりどれだけ以前にクローン片 CF が作成されたかを示す値
 - $AOCR-SCNB$: $SCNB_r$ の OCR_{CF} の平均
 - $AOCR-SCNN$: $SCNN_r$ の OCR_{CF} の平均

RQ4の実験結果

- 右図はAOCR-SCNBとAOCR-SCNNを比較したもの
- 棒グラフの高さが小さいほどクローンがより最近作成されたことになる
- ほぼ全ての場合でAOCR-SCNBが小さいことがわかり、過去に変更がなかったクローンから最近作成したクローンはバグ修正がされやすいことが分かる

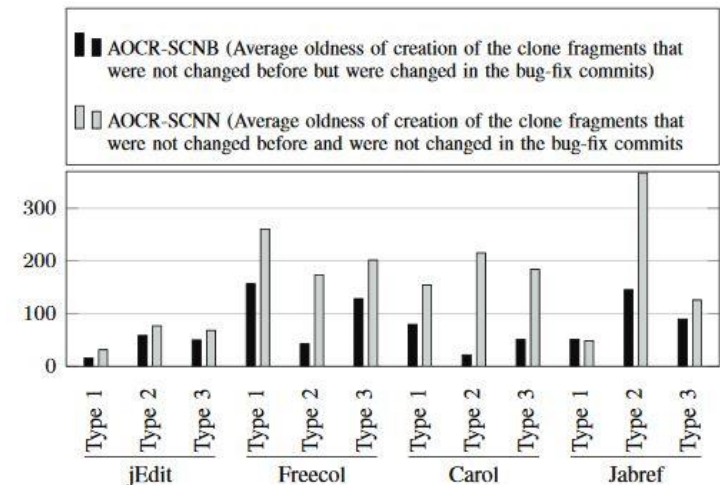


Fig. 5. Comparison of the average oldness of creation of the clone fragments that were not changed in the past.

まとめ, 所感

まとめ

- クローンの変更の頻度はクローンのバグの発生のしやすさに関係がない
- クローンがどれだけ最近変更されたかによってクローンのバグの発生のしやすさが左右される
- 過去に変更されていないクローンから新たに作成したクローンはバグが発生しやすい

所感

- クローン系譜の使い方が参考になった
- RQ3の結果の図を見るとAOC-SCCBが大きい部分がいくつかある。どのような要因でそうなっているのか気になった
- RQ2の結果は今後の研究に結構大きな影響を与えるのではないかと思った