

Title	コードクローンのメトリクス値と開発者の相関の調査
Author(s)	東, 誠; 肥後, 芳樹; 早瀬, 康裕 他
Citation	情報処理学会第70回全国大会講演論文集. 2008, 5, p. 415-416
Version Type	VoR
URL	https://hdl.handle.net/11094/50227
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

コードクローンのメトリクス値と開発者の相関の調査

東 誠[†] 肥後 芳樹[‡] 早瀬 康裕[‡] 松下 誠[‡] 井上 克郎[‡]

大阪大学基礎工学部情報科学科[†] 大阪大学大学院情報科学研究科[‡]

1. はじめに

ソフトウェア保守性を悪化させる問題の一つとして、ソースコード中に同型のコードが存在するコード片(コードクローン[1].以下、クローンとする)がある。例えば、あるクローンの欠陥を修正した場合、他のクローンにも同様の修正を検討する必要がある。

しかし近年、必ずしも全てのクローンが生産性を悪化させるわけではないことが明らかになった[2]。このため、全てのクローンを取り除くことは労力に見合わず、クローンを適切に管理する必要がある。

クローンを管理のためには様々な情報を用いることが考えられるが、本研究ではクローンを編集した開発者に着目する。開発者は様々な役割や目的を持ってソースコードを編集するため、その編集内容には特徴があると考えられる。

本論文では、開発者による編集の傾向の調査手法を提案する。この手法は、まず、版管理システムの一つである CVS[3] から編集の履歴を取り出し、開発者がどのような変更をクローンに対して行なったかを、クローンに対するメトリクス値の変化として抽出する。そして、開発者毎のメトリクス値の変化を調査するために、統計的な手法で検定を行う。

クローンを編集した開発者の情報は版管理システムから容易に取得できるため、この調査手法は実際のソフトウェア開発の現場でも簡単に利用できる。また、この調査方法の応用としては、保守性を悪化させるようなメトリクス値の変化を事前に予測することなどが考えられる。

2. クローン編集傾向の調査手法

開発者によるクローン編集傾向の調査手法を

A method for evaluating correlation between developers and code clone metrics

[†]Makoto HIGASHI, [‡]Yoshiki HIGO, [‡]Yasuhiro HAYASE,

[‡]Makoto MATSUSHITA, [‡]Katsuro INOUE

[†]School of Engineering Science, Osaka University

[‡]Graduate School of Information Science and Technology, Osaka University

図 1 に示す。調査手法の詳細を以下に示す。

2.1 コミットトランザクション情報の抽出

まず、図 1 の(1)のように版管理システムのリポジトリの履歴から同一の開発者による一連のコミットであるコミットトランザクション[4]の情報を抽出する。コミットトランザクションの情報は以下の情報から成り立つ。

- 編集されたファイルのパス名
- 編集されたファイルの変更差分
- 編集後のリビジョン番号
- 編集後のファイルの状態
- コミットした開発者
- コミットされた日時
- コミットログ

コミットトランザクションはコミットされた日時の早い順に並べ、1 から順に通し番号を振る。その後、図 1 の(2)のように以下の時点におけるソースコード(以下、スナップショットとする)を取得し、日時の早い順に通し番号を振る。

- コミットトランザクション 1 の直前
- 各コミットトランザクションの直後

2.2 スナップショットにおけるクローンの検出

次に、2.1 節で取得したスナップショットにおけるクローンを検出する。クローンの検出には CCFinderX[5] を利用する。クローン検出の際に、各クローンのクローンメトリクス(LEN, POP, NIF, RAD, RNR, TKS, LOOP, COND)が CCFinderX によって計算される。

2.3 クローン対応関係の抽出

図 1 の(3)のように、スナップショット間のクローンの対応関係を抽出する。

過去の研究において、複数バージョン間のクローンの対応関係を求める手法が提案されている[6,7]。本研究においては、スナップショット t に存在するクローン a と、スナップショット $t+1$ に存在するクローン b が、以下のいずれかの条件を満たしていた場合、 a, b 間にクローン対応関係があるとする。なお、編集のあるクローンとは、コミットトランザクション t において編集されたファイルの変更差分と重複する箇所が存在するクローンを指す。これは CVS の差分情報とクローン

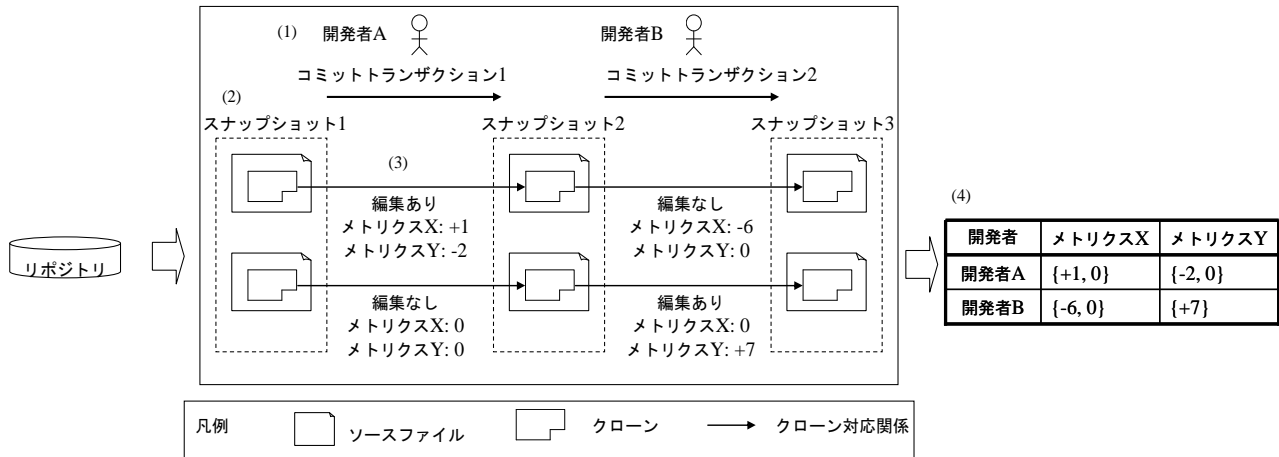


図1 クローン編集傾向の調査手法

の開始行，終了行を用いて特定する[8].

- (1) b が編集のあるクローンであり, b の開始行, 終了行を CVS の差分情報によって調整[8]した値が a の開始行, 終了行に一致する.
- (2) b が編集されたクローンではなく, 以下の全ての条件を満たす.
 - a と a', b と b'がそれぞれクローンペア[1]であるような a' と b'が存在し, a' と b'が条件(1)を満たす.
 - a の開始行 a_{begin} , 終了行 a_{end} , b の開始行 b_{begin} , 終了行 b_{end} において, $(a_{begin} \leq b_{begin} \wedge b_{end} \geq a_{end}) \vee (b_{begin} \leq a_{begin} \wedge a_{end} \leq b_{end})$ が成り立つ.

2.4 検定用データの取得

図1 の(4) のように, 開発者毎のメトリクス値の変化を調査するための検定用データを取得する. 検定用データは開発者を行, メトリクスを列とする表である. 行が開発者Aであり, 列がメトリクスXである昇目の要素は, 開発者Aの全てのコミットトランザクションにおいてクローン対応関係にあるクローン a, b のメトリクスXの値の差のうち, 以下のいずれかの条件を満たすものの集合とする.

- b が編集のあるクローンである.
- a と b のメトリクス X の値の差が 0 でない.

2.5 メトリクス値の変化の差を検定

検定用データのそれぞれのメトリクスについて, 開発者に対して, Kruskal-Wallis 検定[9]を行う. これにより, クローンのメトリクス値の変化と開発者の相関の有無を調査できる.

3. まとめと今後の課題

本研究では, 版管理システムから取り出した編集の履歴を用いて, クローンのメトリクス値の変化と開発者の相関を調査する方法を提案した. 今後の課題としては, 実際のソフトウェアの開発履

歴に対して調査を行うことや, より多くのメトリクスを用いることなどが挙げられる.

謝辞

本研究は一部, 文部科学省リーディングプロジェクト「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた.

参考文献

- [1] Chanchal, K.R. and James, C.: A Survey on Software Clone Detection Research, *Technical Report 2007-541* (2007).
- [2] Cory, K. and Michael, W.G.: “Cloning Considered Harmful” Considered Harmful, *Proc. 13th Working Conference on Reverse Engineering*, pp.19-28, IEEE Computer Society (2006).
- [3] CVS, available from <<http://www.nongnu.org/cvs/>> (accessed 2008-1-15)
- [4] Zimmermann, T., Weissgerber, P., Diehl, S., and Zeller, A.: Mining Version Histories to Guide Software Changes, *IEEE Trans.Softw.Eng.*, Vol.31, No.6, pp.429 – 445 (2005)
- [5] Toshihiro, K.: CCFinder Official Site, available from <<http://www.ccfinder.net/>> (accessed 2008-1-15)
- [6] 川口, 松下, 井上: 版管理システムを用いたクローン履歴分析手法の提案, 電子情報通信学会論文誌D, Vol.J89-D, No.10, pp.2279-2287 (2006).
- [7] M, Kim. and D, Notkin.: Using a clone genealogy extractor for understanding and supporting evolution of code clones, *Proc. 2nd Intl. Workshop on Mining Software Repositories*, pp. 17-21, ACM (2005).
- [8] 川口, 松下, 井上, 飯田: コードクローン履歴閲覧環境を用いたクローン評価の試み, 情報処理学会研究報告, Vol.2006, No.125, pp.49-56 (2006).
- [9] Kruskal, W.H. and Wallis, W.A.: Use of ranks in one-criterion variance analysis, *J.Am.Stat.Assoc.*, Vol.47, No.260, pp.583-621 (1952).