クローン履歴を利用したクローン分析環境

川口真司[†] 松下誠[†] 井上克郎[†]

ソフトウェアの保守工程における大きな問題の一つとしてクローンが指摘されており，これまでに様々なクローン検出手法が提案されている．これらの手法は，ある時点におけるソースコードを対象にしており，クローンがどのような過程で生まれ，どのような変遷を辿ったのかは一切考慮されない．これまでに我々はクローンの履歴を抽出するためのクローン履歴分析手法を提案している．本稿では，抽出したクローン履歴を利用してクローン理解支援を行うクローン分析環境について考察する．クローン履歴を用いることでかつてクローン関係にあったコード片を抽出するほか，クローンの行数やその割合の変化を時系列にそったグラフとして表現できる．このように，本環境を用いることによって，クローンを除去するべきかどうかの判定作業を支援することを目指す．

Clone Analyzing Environment Using Clone History

SHINJI KAWAGUCHI ,[†] MAKOTO MATSUSHITA [†] and KATSURO INOUE[†]

Code clones are seriouse problem in software maintenance process. To solve this problrem, many code clone detenction methods are proposed however, few researches focused on histories of code clones. Histories of code clone is useful for retrieving somewhile clone relationship. We present clone analysing environment using histories of clones. It supports making decision to remove clones or not showing clone sets used to be same clone set and amount of clones.

1. はじめに

保守工程におけるさまざまな問題のなかでも非常に大きな問題の一つとして，ソースコード中に含まれる重複コード (以下，クローン) が挙げられる [1]．これまでにクローンを検出するための様々な手法が提案されており，そのいくつかは実際に利用可能なシステムとして実用化されている．さらに，発見されたクローンを効率的に閲覧，除去するためのシステムも提案されている．これらのシステムを用いることによってソフトウェアに含まれているコードクローンを調査，除去するための労力が大幅に削減される．

これまでに提案されているクローン分析手法は，もしコピーされたコード片に多少の編集操作が加わっていてもこれらをクローンとして抽出するために様々な工夫がこらされている．しかし，それでも全てのクローンを漏れなく検出することは容易ではない．例えば，コード片に対して細かな編集が積みかさなった結果，本質的には依然として類似したコード片であるにもかかわらず，クローンとして検出されなくなる場合がある．

そこで，我々は最新の状態のみならず過去のコピーが行われた時点の状態を考慮して，このようなコード片もクローンとして抽出するための手法をクローン履歴分析として提案している [2]．本稿では我々が提案するクローン履歴分析がソフトウェア開発においてどのような形で活用できるのか，また実際に活用を行う際には，どのようなシステムが必要になるかについて考察する．

2. クローン履歴分析

過去の時点のコードクローン分析を最新版の分析結果に反映させるためには，過去の時点のクローンが現在，どの位置に存在しているかを分析しなければならない．

図 1 は，ソフトウェア中に存在するクローンの変化の様子の一例である．ここでは，当初二つのクローンセット A, B が 3 つのクローンセット A, B, C に変化している．このときに，我々が着目するのはクローンセット B と クローンセット C が元は同じクローンセットであって，関連性が高いということである．

このような関連を抽出するためには，過去の時点に存在するクローンが，現在に存在するクローンと対

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

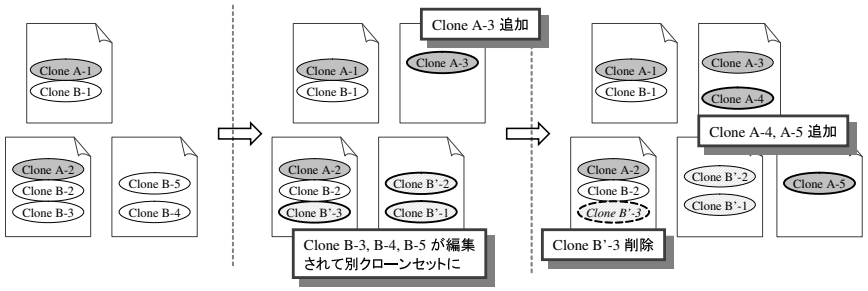

図 1 クローンの変更履歴

応しているのか，対応しているならばどのクローンと対応しているのかを分析する必要がある．そこで，文献 [2] において，過去のクローンから新しい時点のクローンへの対応関係をクローン履歴関係と定義し，その抽出を行う手法を提案している．なお，過去の時点のソースコードを得るために CVS などの版管理システムを用いている．

3. クローン履歴分析環境

これまでの研究において，検出されたクローン全てが排除すべき対象ではなく，クローンの除去を行う際にはその前にクローンを調査して，除去するべきかどうかを吟味する必要があることが知られている．

そこで，クローン履歴そのものをユーザに提示することによって，そのクローンが修正対象か否かの判定を支援することを考えている．例えば，クローンが編集された場合 (図 1 の Clone B-3 が Clone C-1 になった場合) にどのような変更が行われたかを直接確認することができる環境があれば，クローンそのものについての理解がより深められる．

このとき，クローンを理解し，クローン除去を行う際の作業モデルは以下のようになる．

- 改善を要する箇所の絞りこみ
  まず，具体的にどの部分が問題になっているのかを知ることが必要である．この工程においては，どこでクローンの増加が顕著になっているか等，クローンの変化を全体的に捉えることが重要である．
- 改善を要する箇所の特定
  次に，絞りこまれたいくつかのクローンセットが除去する必要があるのかどうかを判断する．そのためには，クローンセットの量や分散具合，変化の活発度などを総合的に判断しなければならない．
- 改善作業
  最後に，改善する必要があると判定されたクローンについて，ソースコードを修正してコードクローンの除去を行う．

これまで改善すべきクローンを発見するために，クローンセットの特徴を表すメトリクスが提案されている．そのうちの一つとしてクローン履歴の活用を考えている．例えば，クローンセットが発生した日時や，クローンセット内に含まれるクローン量の一日あたりの増加量の変化など，さまざまな指標がメトリクスとして活用できると考えられる．

また，発見したクローンを除去するべきかどうかを検討する上でもクローン履歴は有用である．例えばクローン履歴を確認することで，そのクローンセットのクローンがいつ増えたのか，その時点でどのような変更がされたのかを確認することができる．さらに，その変更を行ったのが誰かも版管理システムによって管理されているため，クローンがどのような意図で作成されたのかを理解するために，そのクローンを作った開発者に直接インタビューするのも有効な手段である．誰がクローンを作ったのかが判明していることで，安易なクローンの作成を抑止する効果も期待できる．

目下，上記のモデルを念頭においてクローン履歴関係閲覧システムを Eclipse プラグインとして実装を行っている．現時点では主に開発者の利用を想定して本システムの実装を行っているが，より大局的な視点からソフトウェア中に含まれるクローンを把握することもできるよう実装を進めていきたい．

参 考 文 献

1) Kamiya, T., Kusumoto, S. and Inoue, K.: CCFinder: A Multi-Linguistic Token-based Code Clone Detection System for Large Scale Source Code, *IEEE Trans. Software Engineering*, Vol. 28, No. 7, pp. 654–670 (2002).
2) 川口真司, 松下誠, 井上克郎: 版管理システムを用いたコードクローン履歴分析, 電子情報通信学会技術研究報告 SS2005-31, Vol. 105, No. 208, 小樽商科大学, pp. 43–48 (2005 年 8 月).