

コードクローン情報を用いたソフトウェアプロセス・プロダクト改善

肥後 芳樹†, 楠本 真二†, 井上克郎†

筆者はコードクローン情報を用いてソフトウェア保守支援などを行ってきた。本稿では、コードクローン情報を用いたソフトウェアプロセス・プロダクト改善について議論を行う。

Improving SW Process/Products with Code Clone Information

Yoshiki Higo†, Shinji Kusumoto†, Katsuro Inoue†

We have been developing a code clone analysis system for effective software maintenance. In this paper, we discuss how to apply code clone analysis system to improve actual software processes and products.

1. はじめに

本稿では著者のコードクローンに対する取り組みを紹介し、実際の開発現場での適用に関して議論する。

2. 筆者の取り組み

2.1. コードクローン

コードクローンとは、ソースコード中に存在する互いに一致・類似するコード片を指し、「重複コード」とも呼ばれる。コードクローンは、ソフトウェアの保守作業を困難にする一つの要因である、と指摘されている。

2.2. 支援ツール

著者はコードクローンに対する保守支援を行うため、コードクローン検出ツール CCFinder[1]やコードクローン分析ツール Gemini[2]を開発してきている。CCFinderや Gemini を用いることで大規模ソフトウェアから短時間でコードクローンを検出し、特徴(長さや同形コード片の数など)を指定することで該当するコードクローンのソースコードをシームレスに確認できる。現在では、開発したツールを Gemini/CCFinder パッケージとして、国内外の個人・組織に配布している、また著者自らが企業まで赴いて分析作業を行う場合もある。

2.3. コードクローン分析の目的

現在、コードクローン分析の目的は大まかに三つに分けられる。(1) ソースコードと設計情報が与えられたときに、実装と設計の対応がとれているかの調査を行う。具体的には、設計情報で重複している部分はソースコードでも重複しており、それ以外の部分には重複をし

ている部分がないかを確認する。(2) 長い間保守されているシステムのソースコードとそのバグ情報が与えられたときに、コードクローンとバグの関係を調べることにより、今後の保守作業に役立てる。(3) 複数バージョンのソースコードが与えられたときに、バージョン間でのコードクローンの遷移を調査し、保守作業の効率化を計る。例えば、繰り返し修正が加えられているコードクローンは、集約などを行い将来的な修正コストを抑えるべきであるかもしれない。一方、安定しているコードクローンはシステムの保守コストを悪化させていないので、リファクタリングは不要であると思われる。

3. 議論

現在のところ、コードクローン情報を用いた保守支援として上述の三つを考えている。しかしコードクローン情報はソフトウェアのドメインにあまり左右されない情報であり他の利用方法もあるのではないかと著者は考えている。ソフトウェア開発プロセス・プロダクト改善のための有益なコードクローン利用方法について議論を行いたい。

参考文献

- [1] T. Kamiya, S. Kusumoto, and K. Inoue, CCFinder: A multi-linguistic token-based code clone detection system for large scale source code, IEEE Transactions on Software Engineering, Vol.28, no.7, pp.654-670, Jul. 2002.
- [2] 植田泰士, 神谷年洋, 楠本真二, 井上克郎, "開発保守支援を目的としたコードクローン分析環境", 電子情報通信学会論文誌, Vol.86-D-I, No.12, pp.863-871, Dec. 2003.

†大阪大学 大学院情報科学研究科 コンピュータサイエンス
専攻 〒560-8531 豊中市待兼山町 1-3