

バースマークを用いたソフトウェア拡張開発での流用度合いの測定

玉田 春昭[†] 森崎 修司[†] 門田 暁人[†] 松本 健一[†][†]奈良先端科学技術大学院大学 情報科学研究科

概要

本稿では、既存システムの改造、また、既存システムへの機能追加を行うプロジェクトにおいて、モジュールの流用の度合いを測定する一手法を示す。この度合いを測定することで、テスト工数をモジュールごとに配分する基準を与えることができる。また、開発を外部委託したとき、依頼したモジュール以外に修正が行われていないかを確認することができる。その測定のために、本稿では、バースマークと呼ばれる、プログラム（モジュール）の実行に不可欠な部分をそのプログラムの特徴として抽出し、それを手掛かりにプログラムを比較する技術を用いる。

対象プロジェクトとして、オープンソースで開発されている Apache Struts を選択し、バージョン 1.0.2, 1.1, 1.2.2, 1.2.4, 1.2.7, 1.3.8, 1.3.9 間をバースマークを用いて比較した。その結果、各バージョン間で流用の度合いの大小をバースマークにより相対的に測定できることを確認した。

1 はじめに

近年のソフトウェア開発では、スクラッチからプロダクトを開発することは稀であり、あるプロダクトに対する機能追加や、あるプロダクトを基準モデルとし、新たなプロダクトを開発する場合は非常に多くなってきている。そのとき、基準となったプロダクトと作成されたプロダクトの差分を取り、実際に修正された箇所を特定することで、実際の修正箇所のみをテスト項目として挙げることができ、テスト工程の短縮に繋がる。また、実開発においては、全ての修正箇所は予め選出されており、その場所以外を修正することは保守性を下げることに繋がる。そのため、外部へ開発を委託したとき、必要な箇所以外が修正されていないかを確認する必要がある。

これらの問題を解決するため、本稿ではバースマーク [1, 3, 5] に着目する。バースマークはプログラムの実行に不可欠な情報を特徴として抽出し、それを比較

することで、プログラム同士の類似性を計測する技術である。

2 アプローチ

2.1 バースマーク

バースマークとはプログラム（モジュール）の実行に不可欠な部分をプログラムの特徴として抽出したものであり、抽出したバースマークを比較することで、プログラムの類似性を計測することができる [1, 3, 5]。従来、バースマークは盗用されたプログラムを発見するための技術として用いられてきた。本稿では、基準となったプロダクトとの差をバースマークを用いて計測し、テスト行程以降の投入工数の調整の一基準として用いる。

バースマークはバイナリから直接抽出されるため、ソースコード上の冗長な情報を無視することができる。また、プログラム中のどの情報に着目するかにより異なる種類のバースマークが提案されており、本稿ではバースマークツールとして jbirth[2] を選択した。本稿で用いたバースマークは、変数初期値バースマーク (CVFV)、メソッド呼出系列バースマーク (SMC)、継承関係バースマーク (IS)、使用クラスバースマーク (UC) の 4 種類である。

2.2 分析対象と分析方法

分析対象はオープンソースとして開発されている Web アプリケーションフレームワーク Apache Struts[4] を採用した。そして、Struts のバージョン 1.0.2, 1.1, 1.2.2, 1.2.4, 1.2.7, 1.3.8, 1.3.9 を、バースマークを用いて全ての組み合わせで比較した。各バージョンは 2 番目の数値が大きくなる時、多数の機能が追加されており、大規模な修正が行われている。また、3 番目の数値が大きくなる時、バグ修正を始めた小規模な修正が行われている。

本稿では、一つのクラスを小さなモジュールとしてみなす。そして、異なるバージョン間で同じ名前を持

Measurement of Program Reuse-ratio in Enhancement Projects using Software Birthmarks. Haruaki Tamada[†], Morisaki Shuji[†], Akito Monden[†], Ken-ichi Matsumoto[†]
[†]Graduate School of Information and Science, Nara Institute of Science and Technology.

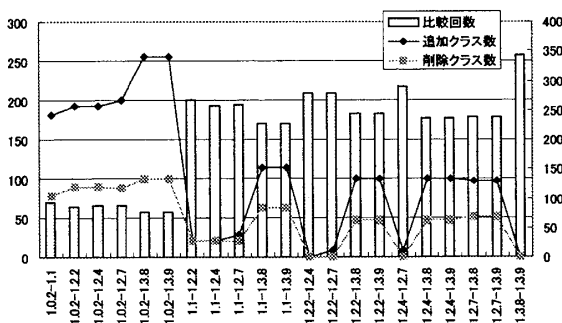


図 1: 各バージョンごとのクラス数の変化

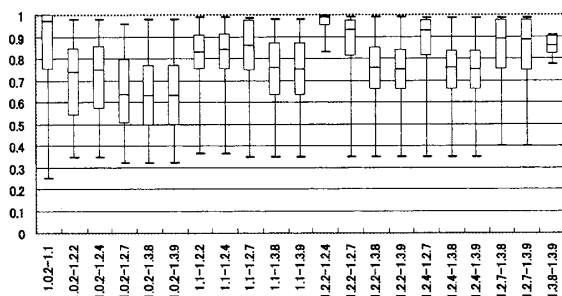


図 2: バースマークによるバージョン間の比較

つクラスをバースマークで比較し、各ペアの類似度の分布に着目する。

各バージョン間を比較したとき、追加されたクラス、削除されたクラスの数と同じ名前のクラスの数分布を図 1 に示す。横軸はバージョンの各組み合わせを表し、左縦軸は折れ線グラフで示されているクラス数を表しており、右縦軸は棒グラフで示されている比較回数を表している。

図からわかるように 3 番目の数値のみが異なっている場合は追加、削除クラス数が少なくなっており、比較回数が多くなっている。逆に 2 番目の数値が異なっている場合は、追加、削除クラス数が多くなっており、比較回数が少ない。また、全体として、削除クラス数は追加クラス数より少ない。

3 結果

各バージョン間をバースマークを用いて比較した結果を箱ひげ図で図 2 に示す。横軸はバージョンの各組み合わせを表し、縦軸は当該バージョン間のクラス名が一致した各ペアをバースマークで比較したときの類似度の分布を表している。

図から 1.2.2 と 1.2.4 の比較が一番差が少なく、次いで 1.3.8 と 1.3.9 の比較が差が少なくなっている。また、

バージョンの差が大きい場合は箱は、バージョン差が小さい場合の箱よりも大きくなっており、バージョン間の変化量をバースマークにより相対的に推し量ることが可能であることがわかる。

一方、箱は 0.5% より上部に存在している。これは、既存のクラスのほとんどはバースマークの類似度が 0.5 より大きくなっていることを表しており、一部のクラスのみが大幅に変更されていることがわかる。

4 おわりに

Apache Struts の 7 バージョンをバースマークを用いて相互に比較することでバージョン間の変化量を測定した。その結果相対的に変化量を推量することができ、変化量が少なければ、流用率が高いと判断することができる。今後はこの測定に最適なバースマークの策定や実開発プロジェクトに用いられたプロダクトに対する調査を行う。

謝辞

本研究の一部は、文部科学省「eSociety 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] Ginger Myles and Christian Collberg. Detecting software theft via whole program path birthmarks. In *Proc. Information Security 7th International Conference, ISC 2004*, volume 3225, pages 404–415. Springer-Verlag GmbH, September 2004. Palo Alto, CA, USA.
- [2] Haruaki Tamada. jbirth: A tool for extracting birthmarks from java class files, 2003. <http://se.aist-nara.ac.jp/jbirth/>.
- [3] Haruaki Tamada, Masahide Nakamura, Akito Monden, and Ken ichi Matsumoto. Java birthmarks —detecting the software theft—. *IEICE Transactions on Information and Systems*, E88-D(9):2148–2158, September 2005.
- [4] The Apache Software Foundation. Apache struts, 2000. <http://struts.apache.org/>.
- [5] 岡本 圭司, 玉田 春昭, 中村 匡秀, 門田 暁人, and 松本 健一. Api 呼び出しを用いた動的バースマーク. *電子情報通信学会論文誌*, J89-D(8):1751–1763, August 2006.