

科学研究費助成事業（基盤研究（S））公表用資料
〔平成28年度研究進捗評価用〕

平成25年度採択分
平成28年3月22日現在

多様なソフトウェア資産の収集・分析・評価と効果的な利活用の研究

Collecting, Analyzing, and Evaluating Software Assets for
Effective Reuse

課題番号：25220003

井上 克郎（INOUE KATSURO）

大阪大学・大学院情報科学研究科・教授



研究の概要

ソフトウェア資産の利活用支援フレームワーク SARF を開発する。SARF では①多様なソフトウェア資産を収集・分析し、②得られた資産の定量的な価値を評価し、③開発環境を通じ資産の利活用支援を行う。コード検索、クローン分析、メトリクス、プログラム解析や再利用等の技術をもとに、実運用に耐えるシステムに発展させ、広く公開して利用の普及を促進する。

研究分野：情報学

キーワード：ソフトウェア工学

1. 研究開始当初の背景

近年のソフトウェア開発では、オープンソースソフトウェア（OSS）や過去に開発したソフトウェアなど多様な資産を効率的に利活用することが必須である。しかし、今までのソフトウェア開発支援環境では、ソフトウェア資産の蓄積、分析・評価、そして再利用のための作業は、それぞれの開発者による各種ツールの利用の方法に任されており、積極的な再利用を支援するための定量的、定性的な基盤ができていなかった。

2. 研究の目的

ソフトウェア資産の利活用支援フレームワーク SARF（Software Analysis and Reuse Framework）について検討を行い、その基盤となるモデルを構築する。また、SARF の枠組みの元で、プログラムの評価、コードの検索、クローンの分析、各種メトリクス、プログラム解析や再利用等の技術の開発を行い、プロトタイプを作成し、実用に供するシステムやツールの開発を行う。

3. 研究の方法

SARF の基盤となるソフトウェアエコシステム全体をモデル化する方法の検討を行う。また、次の3つのサブテーマを行う。

①多様なソフトウェア資産を収集・分析する方法を開発する。

②得られた資産の定量的な価値を評価する方法を開発する。

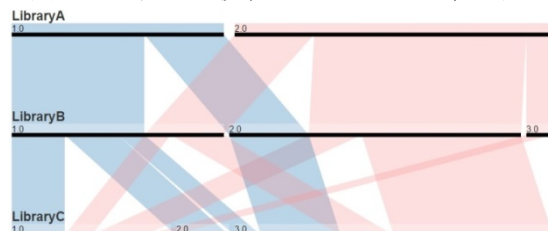
③開発環境を通じ資産の利活用支援を行う方法を開発し、最終的に、実運用に耐えるシステムとして公開する。

4. これまでの成果

これら①～③を統一的に行えるようにするための基盤として、ソフトウェアエコシステムのモデル化方法を開発し、それに基づいて収集、分析、評価が効果的に行えることを示した。

このモデル化では、ソフトウェアエコシステムに対して、個々のソフトウェアシステムの各版を頂点、その間の依存関係を有向辺とした Software Universe Graph（SUG）と呼ぶグラフを構築し、実際にどのようなソフトウェアが別のソフトウェアを利用しているか、そのような利用関係が時間とともにどのように変化するかを可視化できるようにした。

このモデルを用いて、160万あまりの Java プログラムを収集し分析を行った。被利用数（popularity）や多様性（variety、プログラムを利用するシステム種類数）という評価基準を用いて、利用者数は多くないが、利用することが推奨されるソフトウェアの版や、多数が利用しているが危険な版を見つける方法について提案し、開発者が再利用する際の利用指針を提供できることを示した。また、SUG を用いて多数のライブラリの同時使用の様子を直感的に理解できるようにするため、下図のような可視化システムの構築を行った。



た。

例えば、ライブラリ A、B、C が同時に利用されているようなシステムを分析し、A の 1.0 版を使うシステムは、同時にライブラリ B の 1.0 を使っているのが約 7 割、2.0 を使っているのが約 3 割であることを示している。また、最頻の組合せは、ライブラリ A が 2.0、B が 2.0、そして C が 3.0 の組合せであることが視覚的にわかる。

このようなソフトウェアエコシステム全体に対する取り組みの他、①、②、③のそれぞれのサブテーマに関して、活発な研究を行い、トップランクの論文誌や国際会議に発表するなど、大きな成果を挙げている。

例えば、①に関しては、OSS を集め、その集合に対して、効果的にライセンスの違反の可能性のあるファイルを見つけるための分析手法を開発している。ソフトウェアの再利用の際には、そのライセンスを調べ、その指示を厳密に守る必要があるが、ライセンスの変更が行われる可能性もあり、その正当性の検証を行うことは重要である。この手法では、同一内容のソースファイル集合に対して、ファイルのヘッダー部分の文章を理解することによって個々のライセンスを調べ、複数のライセンスを持つ集合を不整合集合として警告を出力する。Linux Debian 7.5 に対して分析したところ、6,763 個の不整合集合を発見した。これはたとえ広く普及しているパッケージであっても、ライセンスに問題を含む多くのものが含まれていることを示しており、再利用者は、注意深く、オリジナルのライセンスは何かを調べる必要があることを示唆している。

また、②に関する重要な成果としては、Web サービス API が提供するインターフェース群の評価を行い、より良いモジュール化するための手法を開発している。この研究では、インターフェースのアンチパターンを検出し、それらが解消されるモジュール構造を、多目的最適化法と呼ぶ探索的ソフトウェア工学手法を用いて実現している。実際に Amazon や Yahoo の Web サービスインターフェースのモジュール化において他の手法より優れた構造が得られることを示した。

③に関する研究成果としては、ソフトウェアを再利用する際に、開発者により良いソフトウェアを提供するために、開発履歴を用いて、最適なりファクタリング手法を提示するシステムを開発した。このシステムでは、Java プログラムの不吉な匂い (Bad Smell) を検出し、探索的ソフトウェア工学手法を利用し、いろいろなりファクタリングの中で、不吉な匂いを小さくし、できるだけ過去と同様な変更で、意味の変更が起こらないものを効率よく探索し、提示する。また、探索には、いくつかの機械学習手法を試し、non-dominated sorting genetic algorithm

(NSGA-II) が優れていることを示した。

5. 今後の計画

今までの各サブテーマで開発を行ってきたいろいろなツール群を基礎として、当初の目的とする開発者の支援を効率良く行うためのプロセスについて、何が十分であり、何が不足しているかの検討を行い、不足部分に関しては、その方法論やツールの開発に注力を行う。例えば、プログラムの来歴 (Provenance) を分析するツールの開発を考える。

さらに、本プロジェクトで開発した技術を統合したプロセスを用いることによって、どの程度、開発者の負担が軽減し、作業が効率化し、また、作成されるシステムの品質、性能が良くなるかの定量的な評価方法について研究を行う。

6. これまでの発表論文等(受賞等も含む)

[1] A. Qunji, M. Kessentini, K. Inoue, M. Ó Cinnéide, "Search-Based Web Service Antipatterns Detection", IEEE Trans. Services Computing (TSC), Accepted.

[2] A. Qunji, M. Kessentini, H. Sahraoui, K. Inoue, K. Deb, "Multi-criteria Code Refactoring Suggestions: An Industrial Case Study", ACM Trans Software Eng. and Methodology (TOSEM), Accepted.

[3] J. Yang, K. Hotta, Y. Higo, H. Igaki, S. Kusumoto, "Classification Model for Code Clones Based on Machine Learning," Empirical Software Engineering (ESE), 20-4, pp.1095-1125, 2015.

[4] Y. Wu, Y. Manabe, T. Kanda, D. German, K. Inoue, "A Method to Detect License Inconsistencies in Large-Scale Open Source Projects", 12th Working Conf. on Mining Software Repositories (MSR), pp.324-333, Firenze, Italy, 2015.

[5] R. Gaikovina Kula, D. German, T. Ishio, K. Inoue, "Trusting a Library: A Study of the Latency to Adopt the Latest Maven Release", 22nd Int. Conf. Software Analysis, Evolution, and Reengineering (SANER), pp. 520-524, Montreal, Canada, 2015.

[6] 井上克郎, 楠本真二, 神谷年洋: "類似コード辺検出に関する研究", 平成 27 年度科学技術分野の文部科学大臣表彰、科学技術賞研究分野、2015.

[7] Y. Higo, S. Kusumoto: "How Should We Measure Functional Sameness from Program Source Code? an Exploratory Study on Java Methods," 22nd Int. Sym. Foundations of Software Engineering (FSE), pp. 294-305, Hong Kong, 2014.

SARF プロジェクトホームページ
<http://sel.ist.osaka-u.ac.jp/SARF/>