

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 25 年 8 月 22 日現在

機関番号：14401

研究種目：基盤研究（C）

研究期間：2010～2012

課題番号：22500026

研究課題名（和文） 類似ソフトウェア分析基盤の構築に関する研究

研究課題名（英文） Developing a Similar Software Analysis Framework

研究代表者

松下 誠 (Makoto Matsushita)

大阪大学・大学院情報科学研究科・准教授

研究者番号：60304028

研究成果の概要（和文）：

ソースコードが互いに類似するソフトウェアを分析する環境を構築するための基盤技術について種々の成果を得た。主な成果として、Android アプリケーションを対象にした部品グラフがべき乗則に従うことを発見し、Java アクセス修飾子のアクセス過剰性がアプリケーションの進化発展に際してどのように遷移するか分析し、ひとたび過剰性を持ったアクセス修飾子は、将来の開発において修正されにくいことを明らかにした。

研究成果の概要（英文）：

Establishing some technologies to develop an environment for analyzing similar source code. Major contributions are as follows: I found that a component graph of Android applications obeys the power law. I investigated Accessibility Excessiveness of Java access modifier from certain Java application and illustrated the evolution of modifiers. Also it is cleared that once a modifier have Accessibility Excessiveness it is hard to fix through the development since emerged.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2010 年度	1,200,000	360,000	1,560,000
2011 年度	1,000,000	300,000	1,300,000
2012 年度	1,100,000	330,000	1,430,000
総計	3,300,000	990,000	4,290,000

研究分野：ソフトウェア工学

科研費の分科・細目：情報学・ソフトウェア

キーワード：類似ソフトウェア、静的ソフトウェア解析

1. 研究開始当初の背景

ソフトウェアの分析技術に関する研究はこれまで数多く行われている。その中でも、ソフトウェアの類似性に着目した分析技術が近年注目を集めている。たとえば、ソースコード中の重複した部分を発見するソースコードクローン検出技術、類似ソフトウェアのクラスタリング技術などがある。これらの技術は、ソフトウェアの再利用、ソフトウェ

アの欠陥検出など、ソフトウェア開発時において多様な支援を行うものとして利用されている。

しかしながら、これらの分析手法で用いられる、分析対象の抽象化、類似部分の検出アルゴリズム、検出結果の可視化といった分析技術中で用いられている要素技術は、分析手法の目的に応じて個別に設計、実装されているために、利用者が必要とする分析手法が既

存手法に存在しない場合、分析手法全体を再構築しなければならないため、多大な労力が必要となる。

また、個別の要素技術については、離散数学の分野や情報可視化の分野など、ソフトウェア工学外で研究が進められている研究成果を応用することが行われているが、他分野での新たな研究成果を用いてより良い分析技術を構築しようとしても、各要素技術が不可分の形で設計されているが故に、困難なものとなっている。

2. 研究の目的

本研究は、これまで行ってきたソフトウェアの類似性に着目した分析技術に関する研究の成果、および研究を行ってきた上で得られた知見に基づき、類似ソフトウェア分析技術を構築するための基盤環境を構築することを目指している。

そのために、類似ソフトウェア分析技術に内包される要素技術を定義したうえで、各要素技術間の相互関係を定義することによって、可搬性の高い要素技術の組み合わせによって類似ソフトウェア分析技術を構築するための手法を明らかにする。その結果をもとに、各要素技術を実装するために必要な基盤環境を整備し、類似ソフトウェア分析技術が要素技術の実装の組み合わせによって実装できるようにする。

3. 研究の方法

類似ソフトウェア分析を行うための基盤技術を開発するにあたり、本研究ではいくつかの対象・分析内容を設定し、それらについて個別に研究を行い、最終的に分析環境を構築するための要素技術を幅広く開発する。本課題にて取り組む研究課題は以下の通りである。

- 現在広く利用されている携帯端末の OS である Android を対象として、それらが相互にどう利用され、結果どのようなソフトウェアがどの程度同じ、あるいは異なるかを判断できるようにする。また、この結果をもとに、ソフトウェア固有の機能を同定して、当該部分のコードを抽出することにより、再利用を行うための基礎情報を提供する方法を考案する。
- 同一ソフトウェア内で互いに似た部分を、リファクタリングで集約する際、ソ

ースコード中の「どこからどこまで」を似ていると判断して集約するか、を判断するための方法を開発する。

- コードクローン分析は従来行われているが、その結果と欠陥との相関について分析し、クローンの存在から欠陥を類推するための技術について開発を行う。
- Java ソフトウェアのアクセス修飾子がどう用いられているかを、複数の似たソフトウェアに対して横断的に分析することにより、アクセス可能範囲がどの程度正しく設定され、また、誤った範囲設定がどのように修正されていくか、自動的に判定する方法について開発する。

4. 研究成果

4.1 平成 22 年度の成果

本年度は主に以下の 3 つの課題について研究に取り組み、それぞれ以下のような成果を得た。

Android は携帯ソフトウェアの基盤であり、Android 向けに開発された多数のソフトウェアが公開されている。それらのソフトウェアがどのように利用されているかをグラフとして分析し、べき乗則が成り立っているかを調査、分析した。Java アプリケーション一般については分析がなされており、べき乗則に従うことがわかっていたが、今回は部品グラフを構築する対象を Android 用アプリケーションに限定しても同じことが言えるかを調査している。

その結果、Android 用アプリケーションの部品グラフも、Java 一般のアプリケーションの部品グラフと同様に、入次数は全体が、出次数も次数の大きな部分がべき乗則に従うことを確認した。このことから、一般のソフトウェア集合と同様に Android のソフトウェア集合においても、少数の一部の部品が何度となく繰り返し利用されていることがわかった。

類似のソフトウェアをまとめて集約する方法の 1 つとして、テンプレートメソッドの形成リファクタリングがある。しかしながら、このリファクタリングを行うには類似メソッド対間の差分を特定し、類似メソッド対間の差分が共通化されるようにメソッド抽出を行う必要がある。この作業は、各メソッドを対比させながら行う必要があり、人手では時間がかかる。

そこで、このようなリファクタリングを行

うことを想定して、固有処理として切りだすコード片の候補を提示する手法を考案した。本手法は、類似メソッド対間の構文上の差分を検出し、その後差分を含みかつメソッド抽出に適したコード片を固有処理として切りだす候補として検出する。候補の検出では、まず差分と同一の範囲に対してメソッド抽出に適しているか判定し、その後段階的に範囲を拡大させながら、同様の判定を行う。さらに、考案した手法を実現するプロトタイプシステムを開発し、実際のソースコードに対して適用した。

適用実験の結果、システムが候補とするコード辺はリファクタリング候補として妥当であること、また、これらの候補を内容によって分類することができ、想定したリファクタリングを支援できることがわかった。

ソフトウェアの類似度としてコードクローンは重要な手がかりを示す。コードクローンは、欠陥を引き起こすと言われているが、先行調査ではコードの生存期間について考慮していないという問題があり、定量的な評価は行われていなかった。

そこで、先行研究で用いられた手法に加えコードの生存期間を考慮した上で欠陥修正がコードクローンを含む割合の調査を行った。調査の結果、生存期間が短いコードクローンほど欠陥修正に含まれる割合が高く、逆に生存期間が長いコードクローンは欠陥修正に含まれる割合が低くなることが分かった。

4.2 平成 23 年度の成果

本年度は、昨年度から継続して Android を対象とした分析技術について研究を行い、以下のような成果を得た。

ソフトウェアの開発者・利用者双方にとって、ある機能を持つ一般に複数のソフトウェアの中から、現在利用したいと考えているコンテキストに一番よく沿っているソフトウェアを選択することは、その後の作業を円滑に進める点において、たいへん重要な問題である。既存研究においても、複数の似たソフトウェアを対象として、その機能の違いを分析の結果として得た上で、利用者等に提供する方法が提案されている。

しかしながら、既存のソフトウェアの違いを分析する方法は、一般的に対象ソフトウェアに対する深い理解が前提となっている。従って、これからソフトウェアを利用しようとしている場合、未だソフトウェアに対する知

識が乏しいため、与えられたソフトウェアの機能の違いを、既存手法を用いて分析することは非常に困難である。

そこで今回は、ソフトウェアの機能の違いを自動的に比較するためのツールを試作した。本ツールでは、ソフトウェアが利用している API を利用することにより、機能の自動抽出と比較を行うこととした。これにより、ソフトウェアに対する理解が浅い場合であっても、ツールが自動的に各ソフトウェアの機能を抽出し、それらの違いを表示することができるようになった。作成したツールを用いて実験を行うことによって、ツールが本来想定する機能を実現できていることがわかった。

4.3 平成 24 年度の成果

本年度は、Java アクセス修飾子に関する研究を行い、以下のような成果を得た。

Java プログラムの各フィールドやメソッドのアクセス修飾子の宣言とその利用状況との差をアクセス過剰性 (AE: Accessibility Excessiveness) と呼び、AE を静的解析するツール ModiChecker が開発されている。ModiChecker をソフトウェアにへ起用することにより、過剰となっているアクセス修飾子が多数存在していることが確認された。一方、プログラムの開発履歴において、過剰なアクセス修飾子がどのタイミングで修正されるのか、あるいは修正されずに残り続けるのかということについては、過去に分析が行われていない。

そこで、OSS である Ant の複数のバージョン間における AE の数の変化を調べ、AE の数とプログラムが成熟しているかどうかの関連について考察した。Java プログラムの開発履歴における、過剰なアクセス修飾子に対する修正作業の実行頻度に関する分析を行った。分析対象としたのは既存の 7 つの Java プロジェクトであり、これらに対して機能拡張された ModiChecker を実行して得られたデータを利用した。分析を行うに当たり、宣言されているアクセス修飾子と実際の利用範囲に基づき、フィールドおよびメソッドを 3 状態に分類した。さらに、バージョン間における状態遷移について、遷移の性質ごとに 6 つのグループへと分類した。

その結果、アクセス修飾子が作成される場合、フィールドとメソッドでは傾向が異なることや、いったん作成されたアクセス修飾子は、それが AE であるかどうかにかかわらず

あまり修正されず、したがって AE が発生するとなかなか解消されず残ってしまうことなどがわかった。また一方で、一部の種類の過剰なアクセス修飾子については、7 つの Java プロジェクト全てにおいて修正が行われていることを確認した。

また、本事業において平成 23 年度に行われた研究成果が、電子情報通信学会において電子情報通信学会ソフトウェアサイエンス研究会研究奨励賞を受賞した。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕 (計 1 件)

小堀一雄, 石居達也, 松下誠, 井上克郎: "Java プログラムのアクセス修飾子過剰性分析ツール ModiChecker の機能拡張とその応用例", SEC Journal Vol.9 (2013, 採録決定) .

〔学会発表〕 (計 5 件)

政井智雄, 吉田則裕, 松下誠, 井上克郎: "類似メソッドの集約のための差分抽出支援", 電子情報通信学会技術研究報告, SS2010-8, Vol.110, No.60, pp.45-50 (2010).

政井智雄, 吉田則裕, 松下誠, 井上克郎: "テンプレートメソッドの形成に基づく類似メソッドの集約支援", 日本ソフトウェア科学会 FOSE2010 ソフトウェア工学の基礎 XVII, pp125~130 (2010).

齋藤晃, 吉田則裕, 松下誠, 井上克郎: "コードの生存期間を考慮したコードクローンと欠陥修正の関係調査", 信学技報, vol.110, no.227, SS2010-28 - SS2010-38, pp.19-24 (2010).

神田哲也, 真鍋雄貴, 松下誠, 井上克郎: "Android 用アプリケーションの部品グラフを対象としたべき乗則の調査", 第 73 回情報処理学会全国大会講演論文集(1) pp.491-492 (2011).

Tetsuya Kanda, Yuki Manabe, Takashi Ishio, Makoto Matsushita, Katsuro Inoue: "A Prototype of Comparison Tool for Android Applications Based on Difference of API Calling Sequences", IEICE Technical Report, vol.111, no.107, SS2011-10, pp.35-40 (2011).

6. 研究組織

(1)研究代表者

松下 誠 (Makoto Matsushita)

大阪大学・大学院情報科学研究科・准教授

研究者番号: 60304028