

平成 26 年 5 月 29 日現在

機関番号：14401

研究種目：若手研究(A)

研究期間：2011～2013

課題番号：23680001

研究課題名(和文) ソフトウェア部品の振舞い特性を用いた欠陥検査の効率化

研究課題名(英文) Efficient Inspection Using Behavioral Properties of Software Components

研究代表者

石尾 隆 (Ishio, Takashi)

大阪大学・情報科学研究科・助教

研究者番号：60452413

交付決定額(研究期間全体)：(直接経費) 5,500,000円、(間接経費) 1,650,000円

研究成果の概要(和文)：本研究では、ソフトウェアシステム内あるいはソフトウェアシステム間で類似した処理の記述を、開発者が効率よく目視検査できるようにする技術の研究を行った。
具体的には、ソフトウェアの振舞いの共通点を自動抽出する手法、ソフトウェア間のソースコードの利用関係を可視化する手法によって検査すべき対象を絞り込むことを可能とし、また、高速なデータフロー解析とその可視化手法によって迅速なソースコード読解を可能とした。

研究成果の概要(英文)：This research investigated techniques for inspecting similar source code fragments in software systems efficiently.

In particular, we have developed techniques to extract common behavior among components and to visualize source code reuse relationships among systems for selecting components to be inspected. We have also developed techniques to extract and visualize data-flow information from a system for efficient source code reading.

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェア工学 データフロー解析 ソースコード比較

1. 研究開始当初の背景

ソフトウェアは、計算機に実行させたい処理を記述した文書の集合である。ソフトウェアを作る際には、類似した機能を持つソフトウェア間で、あるいは単一ソフトウェアの中でも、既に作成した処理を記述した「手続き」と呼ばれる単位を複製し、その内容をわずかに変更することで、新しい処理の実現を行うことが多い。

処理の複製は、開発効率の向上には大きな効果がある一方で、誤りを含んだ手続きを複製した場合には、その誤りも同時に複製してしまうことになる。そのため、開発者は、1つの誤りを発見したとき、同様の誤りが類似した他のソフトウェアや、同一ソフトウェアの他の場所に存在する可能性を調べなくてはならない。

発見された欠陥の複製がソフトウェアに存在するかどうかを調査するためには、該当する処理をコピーしたと記録されている部分だけでなく、誤りに関連したキーワードでソフトウェア全体の検索を行い、網羅的に目視検査を行うといった取り組みが行われている。しかし、大規模なソフトウェアほど複製された処理の量が多く、開発者にとってこのような調査は大きな負担となっている。

2. 研究の目的

本研究では、ソフトウェアの効率的な検査を可能とするため、次の2つの手法の実現を目的とした。

(1) 検査対象の手続きを、その振舞いの特徴に基づいて自動的に分類する手法。同時に検査すべきソフトウェアの処理を効果的に選択することを可能とする。

(2) 検査対象の手続きを理解するために有効な情報を自動的に収集し、提示する手法。開発者の多くはソフトウェアのすべての処理を把握しているわけではないため、開発者が迅速に検査対象の手続きを理解することを可能とする。

3. 研究の方法

本研究は、オブジェクト指向言語として企業システムの開発に広く活用されているJavaを主な対象として研究を行った。

(1) 手続き単位の振舞いを特徴づける手法については、3つの観点から調査を行った。第一に、Javaプログラムに対する解析手法である、既存のパターンマイニング手法およびコードクローン検出手法の応用可能性について調査を行った。第二に、Javaプログラムの実行時情報に対する解析技術を構築し、実行に関する特徴を調査した。第三に、プログラミング言語を問わず活用できる方法として、ソフトウェアのソースコードの字句的な特徴のみを用いる手法について研究を行った。

(2) 検査対象の手続きの理解を支援する手法に関しては、まず、Javaプログラムに対するデータフロー解析技術を開発者が容易に利用できるように、高速な解析手法の構築とその結果の可視化を試みた。また、開発者が、初めて見るソースコードを読解して正しい判断が下せるのか、被験者実験による調査を行った。

4. 研究成果

(1) 検査対象の手続きを分類する手法に関して、以下の成果を得た。

コードクローン検出によって得られる情報の有用性を明らかにした。オープンソースソフトウェアから検出されたコードクローンに対してその内容の調査を行い、開発者が手続き単位を複製するときにはあまり処理内容自体を変更しておらず、手続き内で重要な処理を複製しているときには他方でも同様の機能を実装しているものと期待できる。この結果から、コードクローン検出手法は、欠陥の検査においても有用であると考えられる。

手続きの中に含まれる共通処理を抽出するコーディングパターンマイニング技術に対して、その特性の分析を行った。その結果、従来のコーディングパターン検出で得られていたパターン情報はソースコードの編集に対応して非常に変化しやすいという特徴があり、ソフトウェアの開発が活発に進行している段階では類似性の基準としては適していないことを明らかにした。

パターンマイニングで用いる情報を、ソフトウェアが実行される環境(OS)と相互作用するためのAPIの利用方法のみに限定することで、ソフトウェアが持つ機能の概要を抽出できることを明らかにした。この結果は、ソフトウェア間で共通の機能を持ち、共通の欠陥も含む可能性があるかどうかを判断する際の基準として用いることができる。

実行時情報を用いて、複数の手続きで使用される入力データが共通であるかどうかを自動的に抽出する手法を構築した。具体的には、プログラムの実行を分析することで、1つのデータを構築するために使われた他のデータを列挙し、図1に示すようなグラフの形で可視化する。この図からは、解析範囲外(UNKNOWN)のデータから1つのデータが作られ、それが3つのデータを作成するために使われていること、さらにそれらのデータ間で何らかのデータ更新が行われていることが矢印によって示されている。このような図を用いることで、手続き単位で記述が似ているというだけでなく、使用するデータに実際の共通点があることを開発者が確認できるよ

うにした。

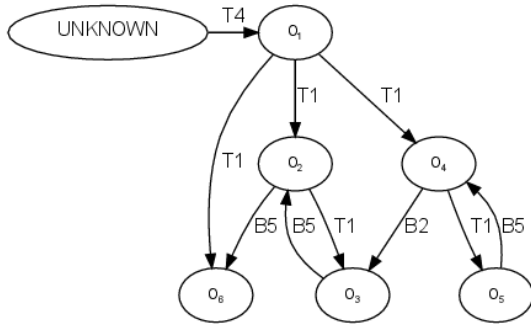


図 1 データの作成関係の可視化

企業において、一部のソフトウェアは、既存ソフトウェア全体を複製してから必要な範囲だけを書き換えることで開発されている。そのようにして開発されたソフトウェアは、個々の部品単位で比較すると内容が大きく変更されている場合もあるが、ソースコード全体では元になったソフトウェアとの類似度が非常に高い。このことを利用して、複数のソフトウェアのソースコードから、ソフトウェア間の再利用（派生）関係を可視化する手法を構築した。図 2 は、オープンソースソフトウェア PostgreSQL のソースコードのみから復元されたバージョン間の関係図である、8.3.x と名付けられたバージョンと、8.4.x と名付けられたバージョンが並行して開発されており、8.4.1 は 8.3.4 や 8.3.12 に比べると 8.3.8 元になっている可能性が高いことを示している。

実験の結果、ソフトウェアの再利用関係をソースコードのみから 70% 以上復元することができることを確認した。正確さの向上については今後の改善が必要であるが、複数の類似したソフトウェアを並行開発しているような環境では、共通機能を持つ可能性の高いソフトウェアだけを迅速に特定し、開発者が必要な検査を実施することが可能となる。可視化ツール PRET-Extractor はインターネット上で公開している。

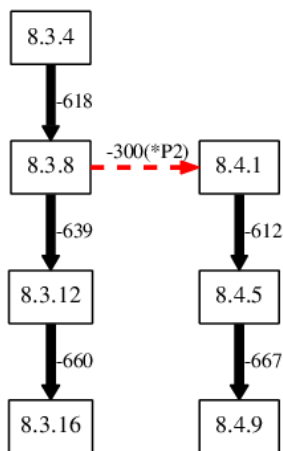


図 2 ソフトウェア間の派生関係

(2) 検査対象の手続きの理解を支援する手法として、以下の成果を得た。

開発者が注目するソフトウェアの一部だけを高速に解析することのできるデータフロー解析を実現した。従来のデータフロー解析技術は正確さを重視して設計されているが、開発者がソースコードを閲覧する作業では正確さよりも速度が重視されることが多い。そこで、データフロー情報を高速に抽出し、開発者が注目しているソースコード位置から遠いデータフロー情報を可視化対象から自動的に除外する手法を構築した。図 3 のように、データフローをグラフ形式で可視化するツールを構築し、統合開発環境に組み込んだ。

欠陥の検査を行う際に必要な作業の 1 つとして、特定の処理が実行される条件を確認する作業を想定し、被験者実験を行った。従来の統合開発環境と比較して、データフローの可視化によって、作業が迅速に行えるようになることを確認した。

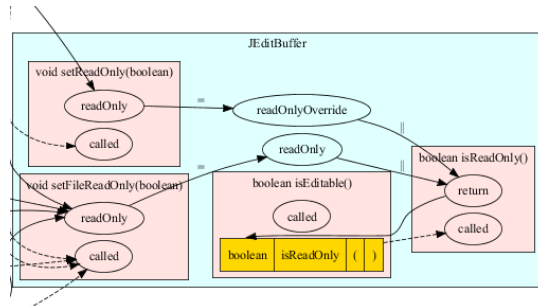


図 3 可視化されたデータフロー

様々なプログラムでデータフロー解析が共通して有効であるかどうかを確認するため、データフロー情報の統計的な特徴に関する分析を行った。特に、データフロー解析によって手続きの動作に影響すると判定される命令の量を調査した。図 4 は、7 つのプログラムに対して解析結果を比較したグラフである。異なるプログラムに対して類似した結果が得られたことから、データフロー解析は多くのプログラムに対して有効であると期待できる。

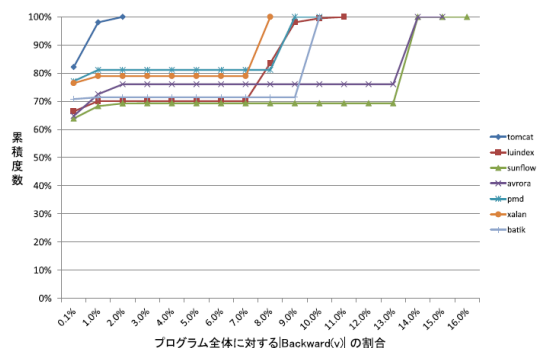


図 4 データフローの影響範囲の分布

開発者が未知のソフトウェア部品の機能を正しく認識できるのか、企業の開発者の協力を得て実験を実施した。その結果、確認すべき手続きの一覧と機能の簡単な説明を与えられた開発者は、手続きと機能との関係の有無をある程度の正確さで判断できることを確認した。このことから、異なる製品、手続きが与えられたとしても、開発者がそれらを同時に検査することが可能であると期待できる。

被験者実験に使用した設問および回答のデータは、他の研究者も利用できるように、インターネット上で公開している。

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4件)

秦野 智臣, 鹿島 悠, 石尾 隆, 井上 克郎: Thin Slice のサイズに関する統計的評価. 情報処理学会論文誌, 査読有, Vol.55, No.2, 2014, pp.971-980.

<http://id.nii.ac.jp/1001/00081300/>

Tetsuya Kanda, Yuki Manabe, Takashi Ishio, Makoto Matsushita, Katsuro Inoue: Semi-automatically Extracting Features from Source Code of Android Applications. IEICE Transactions on Information and Systems, 査読有, Vol.E96-D, No.12, 2013, pp.2857-2859.

DOI: 10.1587/transinf.E96.D.2857

Takashi Ishio, Hiroki Wakisaka, Yuki Manabe, Katsuro Inoue: Towards Logging Optimization for Dynamic Object Process Graph Construction. IEICE Transactions on Information and Systems, 査読有, Vol.E96-D, No.11, 2013, pp.2470-2472.

DOI: 10.1587/transinf.E96.D.2470

中野 佑紀, 伊達 浩典, 渡邊 結, 石尾 隆, 井上 克郎: プログラム実行履歴を用いたオブジェクト生成関係の可視化, 査読有, 情報処理学会論文誌, 2012, Vol.53, No.3, pp.1166-1176.

<http://id.nii.ac.jp/1001/00081300/>

[学会発表](計16件)

Yuki Kashiwabara, Yuya Onizuka, Takashi Ishio, Yasuhiro Hayase, Tetsuo Yamamoto, Katsuro Inoue: Recommending Verbs for Rename Method using Association Rule Mining. 1st IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering. 2014年2月6日, Antwerp, Belgium.

Takashi Ishio, Shinpei Hayashi, Hiroshi Kazato, Tsuyoshi Oshima: On the Effectiveness of Accuracy of Automated Feature Location Techniques. 20th IEEE

Working Conference on Reverse Engineering, 2013年10月14日, Koblenz, Germany.

Tetsuya Kanda, Takashi Ishio, Katsuro Inoue: Extraction of Product Evolution Tree from Source Code of Product Variants. 17th Software Product Line Conference, 2013年8月30日, Tokyo, Japan.

Hironori Date, Takashi Ishio, Katsuro Inoue: Investigation of Coding Patterns over Version History. 4th IEEE International Workshop on Empirical Software Engineering in Practice. 2012年10月26日, Osaka, Japan.

Takashi Ishio, Shogo Etsuda, Katsuro Inoue: A Lightweight Visualization of Interprocedural Data-Flow Paths for Source Code Reading. 20th IEEE International Conference on Program Comprehension. 2012年6月11日, Passau, Germany.

[図書](計0件)

[産業財産権]
出願状況(計0件)

取得状況(計0件)

[その他]

ホームページ等

PRET-Extractor 公開ページ
<http://sel.ist.osaka-u.ac.jp/pret/>
Feature Location Research Dataset 公開ページ <http://sel.ist.osaka-u.ac.jp/FL/>

6. 研究組織

(1)研究代表者

石尾 隆 (ISHIO, Takashi)
大阪大学・大学院情報科学研究科・助教
研究者番号: 60452413

(2)研究分担者

なし

(3)連携研究者

なし