

科学研究費助成事業 研究成果報告書

平成 28 年 10 月 23 日現在

機関番号：17102

研究種目：挑戦的萌芽研究

研究期間：2013～2015

課題番号：25540026

研究課題名(和文)クラッシュログからのソースコード修正箇所の推定に向けた挑戦

研究課題名(英文)Mining Crash Report Repositories for Bug Localization

研究代表者

亀井 靖高 (Kamei, Yasutaka)

九州大学・システム情報科学研究科(研究院・准教授)

研究者番号：10610222

交付決定額(研究期間全体)：(直接経費) 2,800,000円

研究成果の概要(和文)：本研究では、クラッシュログリポジトリに着目し、ユーザ環境下でソフトウェア障害が発生した際に、ソースコードのどの箇所にバグが混入しているかを開発者にフィードバックするモデルを提案する。クラッシュログはソフトウェア障害が発生した際に自動的に生成され、開発プロジェクトに送信されるログのことである。本提案では、(a) リポジトリ間のリンキング技術の開発と学習データの作成、(b) ソフトウェアの修正箇所推定のためのモデリング技術、(c) プロトタイプツールの作成、を実施し、予備調査の結果、本ツールが出力する上位5件のファイルの中に、約50%の確率でクラッシュの欠陥ファイルが含まれることがわかった。

研究成果の概要(英文)：This research project proposes the model that suggests the location of software faults to developers when the software faults happen in user environments, by making use of crash repositories. Crash logs in the crash repositories are automatically generated at the time that software faults happen in user environments and sent to the projects that develop the software. This research project studies on (a) linking data across repositories (e.g., version control repositories, issue tracking repositories, and crash repositories), (b) developing fault localization techniques, and (c) developing a prototype tool. The case study shows that the prototype tool can identify the location of software faults with 50% of accuracy by showing the top 5 files that are likely to be defect-prone.

研究分野：ソフトウェア工学

キーワード：ソフトウェアリポジトリマイニング オープンソースソフトウェア ソフトウェア信頼性 クラッシュリポジトリ

1. 研究開始当初の背景

- (1) 昨今、情報システムは日常生活の中に浸透しており[1]、例えば、携帯端末で用いられる Android OS は、2011 年 9 月の時点で、世界で 6,000 万人以上ものユーザによって利用されている。そのため、たった 1 つのソフトウェアバグが引き起こす社会的影響は甚大であり、今日の社会ニーズとして、開発プロジェクトがソフトウェア障害の発生をいち早く認識して修正することが至上命題となっている。
- (2) 問題解決の一手段は、Bugzilla や JIRA といった障害報告を受け付けるシステムをオンライン上に設けて、ユーザの声をいち早く受け取ることである[2]。例えば、世界第 2 位のブラウザシェアを誇る Firefox では、年間に 6 万件もの障害報告がユーザからなされており、開発プロジェクトはユーザからの障害報告の内容を確認した後に、修正箇所を推定/特定し、修正パッチをリリースするというサイクルを繰り返すことで、安心/安全なシステムを長い期間にわたって提供しよう試みている。
- (3) しかしながら、障害の発生から修正が開始されるまでには平均して 180 日以上もの日数を要しているのが現状である[3]。

2. 研究の目的

- (1) 今までの研究課題の実施を通して得た問題の原因は次の 2 つに集約される。本研究の目的は、問題 1、および、問題 2 を解決し、障害の発生から修正までに要する日数の短縮を支援することである。
- (2) 問題 1: ソフトウェア障害を自発的に報告するユーザはごくわずかであるため、開発プロジェクトが障害の発生を認識するまでに時間を要する。
- (3) 問題 2: たとえ障害がユーザから報告されたとして、開発者が修正箇所を推定したり特定したりするために十分な情報が含まれていないことが多い。(多くの場合、ユーザは開発に関する知識を有していない)。

3. 研究の方法

- (1) 本研究では、クラッシュログリポジトリに着目し、ユーザ環境下でソフトウェア障害が発生した際に、直ちにソースコードのどの箇所にもバグが混入しているかを開発者にフィードバックするモデルを提案する。クラッシュログは、ソフトウェアの利用中、障害が発生した際に開発プロジェクトに送信されるユーザの実行ログであり、多数のユーザを有する開発プロジェクトでは導入され、蓄積されつつある。例えば、Firefox プロジェ

クトでは、2009 年からその本格的な運用が始まっている[4]。

- (2) 「クラッシュログからのソースコード修正箇所の推定」の実現を目指し、次の 3 つの研究項目を実施する。主に、(a) リポジトリ間のリンク技術の開発と学習データの作成、(b) ソフトウェアの修正箇所推定のためのモデリング技術、(c) プロトタイプツールの作成、を実施する。
- (3) 従来はユーザから主体的に行われた障害報告のデータにのみ着目しているが、本研究では自動収集されるクラッシュログに着目している点が特色である。クラッシュログは、ユーザが主体的に報告する意思がなくとも、ユーザ環境でソフトウェア障害が発生した際に自動的に障害内容が報告されるため、Bugzilla と比べて早い段階でプロジェクトに障害報告がなされる。

4. 研究成果

- (1) リポジトリ間のリンク技術の開発と学習データの作成: クラッシュレポートと不具合報告のリンクとして、関連づけられている否かを判断するための 2 つの項目を特定した。1 つ目は、Bugzilla の不具合票の Crash Signature という項目である。この項目内に各クラッシュレポートのクラッシュタイプ名が記述されている場合、そのレポートは不具合に関連付けられている。2 つ目は、クラッシュレポートの bug list という項目である。この項目内に不具合票の ID が記載されていた場合、このクラッシュレポートと同じクラッシュタイプのレポートは不具合に関連付けられている。

学習データに用いる特徴量として、クラッシュの発生時期、各メソッドの呼び出し回数、呼び出し順序、ソースコードの変更時期などが予測のための特徴量として役立つことがわかった。

学習データの特徴を調べるために、Firefox プロジェクトで 3 年間に蓄積された約 460,000,000 件のクラッシュレポート、および、Bugzilla で 15 年間に蓄積された 930,000 件の不具合レポートを対象として調査を行った。その結果、例えば、不具合に関連づけられているクラッシュタイプの 90% は、1,000 件のクラッシュレポートが送信されるまでの間に関連づけられていることを明らかにした。

本研究成果の詳細は、主に本報告書「5. 発表論文等」の[雑誌論文]の②等や[学会発表]の④、⑤や⑧等で報告されている。

- (2) ソフトウェアの修正箇所推定のためのモデリング技術: 研究成果(2) で作成し

た学習データを用いて、修正箇所推定のモデリング技術を開発した。クラッシュレポートのスタックトレースを解析し、呼び出されたメソッドのファイルを抽出する。スタックトレースに出現するファイル名の出現頻度やメソッドの出現頻度を元に、修正箇所の予測値を算出する。スコアの算出方法については、本報告書「5. 発表論文等」の[学会発表]の⑦等を参考にされたい。

モデリングの精度向上のために、クラッシュレポートによる不具合発見とテストによる不具合発見の分類を目的として、テストコードとプロダクトコード変更とのリンクを行った。本成果により、クラッシュレポートと不具合報告、変更コードとのミスリンクの削減（モデリング技術の精度向上に寄与）が期待できる。

その他に、クラッシュレポートのリンク先である **Bugzilla** に蓄積された不具合報告を修正するために、ソースコード中の修正すべき箇所の推定を試みた。関数単位とクラス単位で推定手法を比較した結果、細粒度である関数単位の方が 1/7 の作業量（開発者が目視でチェックするコード行数）で箇所の推定を達成できる可能性を示した。

本研究成果の詳細は、主に本報告書「5. 発表論文等」の[雑誌論文]の②や[学会発表]の⑥等で報告されている。

- (3) プロトタイプシステムの作成:スタックトレースの解析機能、ソースコードのメトリクス計測機能、欠陥箇所予測機能を実装した。欠陥箇所予測機能のコア部分（スコアリング）は **Ruby** で実装されており、規模は約 300 行程度である。メトリクス計測機能の一部（特にプロダクトメトリクス）は **TechMatrix** 社の **Understand** を用いて計測し、その出力結果を加工することで実現している。

本ツールは対象ソフトウェアの言語、構成等に依存することなく、適用することが可能である。予備調査の結果、ツールが出力する上位 5 件のファイルの中に、約 50%の確率でクラッシュの欠陥ファイルが含まれることがわかった。

本研究成果の詳細は、主に本報告書「5. 発表論文等」の[学会発表]の⑦等で報告されている。

参考文献

- [1] Paul Luo Li, James Herbsleb, Mary Shaw, Brian Robinson, ``Experiences and Results from Initiating Field Defect Prediction and Product Test Prioritization Efforts at ABB Inc. ,'' Proceedings of the International Conference on Software Engineering (ICSE), pp.413-422, 2006.

- [2] Pamela Bhattacharya, Iulian Neamtiu, ``Fine-grained incremental learning and multi-feature tossing graphs to improve bug triaging, '' Proceedings of the International Conference on Software Maintenance (ICSM), pp.1-10, 2010.
- [3] Akinori Ihara, Masao Ohira, Ken-ichi Matsumoto, ``An analysis method for improving a bug modification process in open source software development, '' Proceedings of the International workshops on Principles of Software Evolution (IWSE), pp.135-144, 2009.
- [4] Foutse Khomh, Tejinder Dhaliwal, Ying Zou, Bram Adams, ``Do faster releases improve software quality? An empirical case study of Mozilla Firefox, '' Proceedings of the International Conference on Mining Software Repositories (MSR), pp. 179-188, 2012.

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、全て査読有り）

- ① Yasutaka Kamei, Takafumi Fukushima, Shane McIntosh, Kazuhiro Yamashita, Naoyasu Ubayashi and Ahmed E. Hassan, ``Studying Just-In-Time Defect Prediction using Cross-Project Models, '' Journal of Empirical Software Engineering. (To appear)
- ② 小須田 光, 亀井 靖高, 鶴林 尚靖, ``クラッシュレポートの送信頻度と不具合との関連付けに関する実証的評価'', コンピュータソフトウェア, Vol. 32, No. 4, pp.131-140, December 2015.
- ③ Ayse Tosun Misirli, Emad Shihab and Yasutaka Kamei, ``Studying High Impact Fix-Inducing Changes, '' Journal of Empirical Software Engineering, Vol. 21, Issue. 2, pp. 605-641, April 2016.
- ④ Shane McIntosh, Yasutaka Kamei, Bram Adams and Ahmed E. Hassan, ``An Empirical Study of the Impact of Modern Code Review Practices on Software Quality, '' Journal of Empirical Software Engineering. (To appear).

[学会発表]（計15件、内査読有り7件、招待講演2件）筆頭著者が発表者。うち8件は下記の通り。

- ① Yasutaka Kamei, Emad Shihab, ``Defect Prediction: Accomplishments and Future Challenges, '' In Proceedings

- of Leaders of Tomorrow / Future of Software Engineering Track at International Conference on Software Analysis, Evolution, and Reengineering (SANER2016), Vol. 2, pp. 33-45, March 2016 (Osaka, Japan).
- ② Shane McIntosh, Yasutaka Kamei, Bram Adams and Ahmed E. Hassan, ``The Impact of Code Review Coverage and Code Review Participation on Software Quality: A Case Study of the Qt, VTK, and ITK Projects,`` In Proceedings of International Working Conference on Mining Software Repositories (MSR 2014), pp. 192-201, June 2014. (Hyderabad, India)
- ③ Chakkrit Tantithamthavorn, Akinori Ihara, Hideaki Hata, and Kenichi Matsumoto, ``Impact Analysis of Granularity Levels on Feature Location Technique,`` In Proceedings of the Asia Pacific Requirements Engineering Symposium (APRES), pp. 135-149, April 2014. (Auckland, New Zealand)
- ④ 小須田 光, 亀井 靖高, 鶴林 尚靖, ``クラッシュレポートの送信頻度が不具合との関連付けに与える影響``, ソフトウェア工学の基礎ワークショップ FOSE2014, pp. 25-34, December 2014.
- ⑤ 亀井 靖高, 長本 貴光, ラピュト シャシャンク, 小須田 光, 伊原 彰紀, 鶴林 尚靖, ``クラッシュリポジトリマイニング -ソースコード 欠陥箇所の特定向けて-``, ソフトウェア工学の基礎ワークショップ FOSE2013, pp. 173-184, November 2013.
- ⑥ 南智孝, 伊原彰紀, 坂口英司, 松本健一, ``プロダクトコード変更に伴い共進化するテストコード特定手法の提案,`` ウィンターワークショップ 2016・イン・逗子 論文集, pp. 51-52, January 2016.
- ⑦ 小須田 光, 亀井 靖高, 鶴林 尚靖, ``ユーザ障害情報によるソースコード欠陥箇所予測ツール,`` 情報処理学会研究報告, ソフトウェア工学研究会, December 2015.
- ⑧ 小須田光, 亀井靖高, 伊原彰紀, 鶴林尚靖, ``クラッシュレポートが不具合修正に与える影響の分析,`` 情報処理学会研究報告, ソフトウェア工学研究会, March 2014.

〔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

6. 研究組織

(1) 研究代表者

亀井 靖高 (KAMEI, Yasutaka)

九州大学・大学院システム情報科学研究
院・准教授

研究者番号 : 10610222

(2) 研究分担者

伊原 彰紀 (IHARA, Akinori)

奈良先端科学技術大学院大学・情報科学研
究科・助教

研究者番号 : 40638392