

操作履歴を利用した不具合票自動生成に向けて

池田 祥平¹ 坂口 英司¹ 伊原 彰紀¹ 松本 健一¹

概要：OSS (Open Source Software) で発見される不具合は、OSS プロジェクトが管理する不具合管理システムを通して報告される。不具合管理システムには、不具合発見者が修正に必要な情報を報告するため、不具合票の生成を支援する仕組みが実装されている。しかし、不具合の再現手順や不具合発見時のシステムの挙動などは、不具合発見者にとって報告することが容易ではない。本論文では、開発のために行われる操作履歴 (例：コマンド実行履歴やファイル変更履歴、メソッド実行履歴など) を自動取得することで、開発者が不具合修正に必要と考える項目を自動的に収集し、不具合票の半自動生成を目指す。

キーワード：不具合票, 自動生成, オープンソースソフトウェア

Toward Automatic Generation of Bug Reports Using Operation History

SHOHEI IKEDA¹ HIDESHI SAKAGUCHI¹ AKINORI IHARA¹ KENICHI MATSUMOTO¹

1. はじめに

OSS 開発では不具合を一元管理するために Bugzilla ^{*1}, Jira ^{*2}をはじめとする不具合追跡システムを使用している。OSS に不具合を発見した開発者、または、ユーザは、不具合が発見されたプロダクトやプラットフォーム、不具合の再現手順、発生した現象などを報告する。しかし、不具合報告は開発者にとって必ずしも容易ではなく、また、開発者が当該不具合の修正のために必要とする情報を理解することは難しい。

Bettenburg ら [1] は、不具合発見者が報告する不具合情報と修正担当者が求める情報に乖離があることを明らかにしている。具体的には、報告者はプロダクトやバージョンを報告するが、修正担当者はあまり必要と考慮しておらず、逆に修正担当者はスタックトレースやテストケースを重要と考慮しているが、報告者から提供されることは少ない。開発者が求める不具合情報を取得するために、不具合票の生成を支援する仕組みが不具合管理システムに実装され

ている。例えば Bugzilla では、不具合の登録時、不具合が発生したプロダクトのコンポーネントやバージョン、OS (Operating System) を選択式としている。加えて、「不具合が発生する手順を記述してください」という旨の注意書きを示している。また、Moran ら [3] は、アンドロイドアプリケーションにおける不具合再現手順の報告を、一部自動化するシステム FUSION を提案している。FUSION では、クリックやスワイプといった操作と対象となるボタンなどのコンポーネントが選択肢で示されるため、不具合報告者は選択肢から選ぶことで不具合報告が可能である。同時に、操作すべきコンポーネントを強調したスクリーンショットを生成し、修正担当者の理解を支援する。

本論文では、開発のために行われる操作履歴 (例：コマンド実行履歴やファイル変更履歴、メソッド実行履歴など) を自動取得することによって、開発者が不具合修正に必要と考える項目を自動的に収集し、不具合票の半自動生成を目指す。

¹ 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

*1 bugzilla.mozilla.org

*2 www.atlassian.com/software/jira

表 1 生成可能な不具合票の項目

項目	必要な情報
再現手順	コマンド実行履歴, ファイル変更履歴, コード変更履歴
発生した挙動	環境情報から仮想環境を構築し, 再現手順を実行して直接確認
期待される挙動	本システム対象外
スタックトレース	メソッド・変数呼出履歴
バージョン	環境構築時のクローン情報
コンポーネント	環境構築時のクローン情報
プロダクト	環境構築時のクローン情報
エラーレポート	エラー発生時のコンソールログ
オペレーティングシステム	環境構築時に設定した OS
ビルド情報	ビルドコマンド時のコンソールログ

2. 不具合票の自動生成に向けて

2.1 要件

(1) 操作履歴が取れること

不具合票を自動生成するために, 本論文では不具合発見者の操作履歴を用いる. 操作履歴は, 不具合発見者が OSS 利用中に実施した行動であり, 欠陥が発生したプロセスを示す. 従って, 不具合票に求められる再現方法, 及び, 発生した挙動を説明することができる.

(2) 環境の詳細が取れること

OSS が稼働する環境が異なると, 不具合を再現することができない場合がある. 環境変数をはじめとする, 個人特有の設定などを理解することにより不具合の再現を実現する.

2.2 不具合票の生成と利用シナリオ

提案手法では次のような状況における不具合報告を対象とする. (a) OSS の開発・テスト中に発生した不具合を, その OSS における他の開発者に報告する. (b) OSS ライブラリを利用した際に発生した不具合を, OSS ライブラリの開発者に報告する.

不具合発見者が使用する機器において, 本提案手法では, 機器が使用する OS, ライブラリの種類, 及び, 開発者が行ったコマンド実行履歴やファイル変更履歴, コード変更履歴, メソッド・変数呼出履歴といった操作履歴を適宜記録する. コードの自動整形など, 開発者の意図せず自動で行われた操作も同様に記録する. 不具合が発生した場合, 提案手法は (i) OS や利用ライブラリといった環境の情報と (ii) 不具合発生時点までの操作履歴に基づいて不具合票を生成する. 表 1 は, 開発者が不具合修正に必要とする項目と, 本提案手法が各項目を説明するために収集する情報を示す.

3. 関連研究

Bettenburg ら [1] は, Apache, Eclipse, Mozilla の修正担当者と不具合発生者間における不具合票について調査し, 不具合発見者が報告する不具合情報と修正担当者が求める情報に乖離があることを明らかにした. 例えば, 報告者はプロダクトやバージョンを報告するが, 修正担当者はスタックトレースやテストケースを求めている. また, 修正担当者にとって不具合修正に最も役立つ情報は, 再現手順や発生した挙動であることを示した. Davies ら [2] は, Eclipse, Firefox, Apache HTTP, Facebook API の不具合票から, 不具合発見者が報告する情報と, それが不具合修正に与える影響を調査した. 結果, 開発者が不具合修正に役立つと考えるスタックトレースやテストケースはほとんど含まれていないことを示した. 一方で, 彼らはそれらの情報を自動で収集する方法については調査していない.

Moran ら [3] は, アンドロイドアプリケーションにおける不具合再現手順の報告を, 一部自動化するシステム FUSION を提案した. FUSION では, 不具合報告者はアプリケーションに対する操作と対象となるコンポーネントを選択肢から選択することで, 操作するコンポーネントを強調したスクリーンショットとともに, 再現手順を報告できる. 本論文とは不具合の再現手順を容易化する点で関連するが, スクリーンショットを手動で組み合わせる点で, 操作履歴から自動で不具合票を生成する本論文とは異なる.

4. おわりに

本論文では, 不具合報告に必要な情報を自動収集することを目的として, 何を取得すべきかを検討した. 本論文により, 不具合が詳しく修正担当者に伝わるようになるため, 修正が容易となり, 開発の促進が期待される. ワークショップでは, 他の収集すべき情報, 本手法の評価方法について議論したいと考えている.

謝辞 本論文の一部は, 文部科学省科学研究補助費 (若手 B: 課題番号 16K16037) による助成を受けた.

参考文献

- [1] N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and T. Zimmermann: *What Makes a Good Bug Report?*, In Proceedings of the 16th International Symposium on Foundations of Software Engineering (2008).
- [2] S. Davies and M. Roper.: *What's in a bug report?*, In Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, pages 26:1-26:10, New York, NY, USA, (2014).
- [3] Kevin Moran, Mario Linares-Vasquez, Carlos Bernal-Cardenas, Denys Poshyvanyk: *Auto-Completing Bug Reports for Android Applications*, ESEC/FSE 2015 Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering Pages 673-686 (2015).