The University of Osaka
Institutional Knowledge Archive

Title	Study on Cost-Effective Debugging Methods under Restricted Resources
Author(s)	嶋利, 一真
Citation	大阪大学, 2022, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/88137
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

様式3

論 文 内 容 の 要 旨

氏 名 (嶋利 一真)	
論文題名	Study on Cost-Effective Debugging Methods under Restricted Resources （限られた資源を用いた効率的なデバッグ手法に関する研究）

論文内容の要旨

近年，ソフトウェアの開発コストは増加傾向にあり，それに伴いソフトウェア開発コストの半分を占めるデバッグコストも増加している．このコストを削減するために様々なデバッグ手法が提案されているが，必要なリソースを事前に予測できないデバッグ手法では，予想外のデバッグコストの増加を招き，開発コストの上昇につながる可能性がある．そのため，限られた資源を用いた効率的なデバッグ手法が必要となる．そこで，本論文ではデバッグに利用可能な資源が限られている場合において既存のデバッグ手法の問題を解決するような手法を提案する．

本論文の第一部では，各命令ごとに最新k件の観測値を記録することで限られた保存領域を用いてJavaプログラムの実行を記録するNear-Omniscient Debuggingを提案する．評価では，実行トレースのデータ依存関係の精度，実行トレース量，バグ関連命令が完全に記録されているバグの割合を測定し，Near-Omniscient Debuggingの有用性を確認した．また，記録した実行トレースを用いてソースコードに観測値を埋め込み可視化するツールを作成し，本ツールが不完全な実行トレースを用いたデバッグが可能であることを2つの例で示した．

本論文の第二部では，企業のソフトウェアにおけるテストケースの選択問題に取り組む．ライブラリの更新を行う際に，更新後のライブラリを用いたソフトウェアに対して回帰テストを行う必要がある．しかし，多くのテストケースが存在する場合，ライブラリの互換性の検証に多くの時間がかかる．この問題を解決するために，実行時情報の類似性に基づいて互換性検証に必要なテストを選択する手法を開発した．提案手法で選択したテストケースのカバレッジと実行時間の多様性を評価し，手法の有用性を検証した．

本論文の第三部では，類似コード片検索結果に対するフィルタリング手法を提案する．類似コード片検索ツールで類似バグのあるコード片を検索した場合に，検索結果内のすべてのコード片を検査すべきであるが，一般に開発リソースには制限がありすべてを検査することは難しい．この問題に対処するため，本フィルタリング手法では，検索結果に対してクラスタリングを適用してコード片をグループ化し，正解確率の低いコード断片を出力から除外しする．バグを含む類似のコード断片を集めたデータセットに対して，本フィルタリング手法を適用し，その有用性を確認した．

これらの提案したデバッグ手法を用いることで，開発者は自らが利用可能なデバッグ資源に応じて効率的なデバッグ作業を行うことが可能となる．

様式 7

論文審査の結果の要旨及び担当者

氏　名　（　嶋利　一真　）		
	（職）	氏　名
論文審査担当者	主 査　教授	井上　克郎
	副 査　教授	楠本　真二
	副 査　教授	伊野　文彦

論文審査の結果の要旨

本学位論文を審査担当者間で精査した結果、以下に述べる内容を確認した。

本学位論文は、限られた資源を用いた効率的なデバッグ手法に関して3つの想定状況に対して以下の3つの研究を行った。

1つ目は、学位論文の2章と3章で示すように、網羅的なデバッグ手法であるOmniscient Debuggingの記録量が膨大となるといった問題を解決するための実行トレース記録手法を研究した。具体的な手法として、命令ごとに最新k件の観測値を記録することで限られた保存領域を用いてJavaプログラムの実行を記録するNear-Omniscient Debuggingを提案した。提案手法における実行トレースのデータ依存関係の精度、実行トレース量、バグ関連命令が完全に記録されているバグの割合を測定し、Near-Omniscient Debuggingの有用性が確認された。また、記録した実行トレースを用いてソースコードに観測値を埋め込み可視化するツールを作成し、本ツールが不完全な実行トレースを用いたデバッグが可能であることが2つの例で示された。

2つ目は、4章で示すように、企業のソフトウェアにおけるテストケースの選択問題について研究した。ライブラリの更新を行う際に、更新後のライブラリを用いたソフトウェアに対して回帰テストを行う必要があるが、多くのテストケースが存在する場合、ライブラリの互換性の検証に多くの時間がかかる。この問題を解決するために、実行時情報の類似性に基づいて互換性検証に必要なテストを選択する手法を開発した。テストケースの実行時情報をクラスタリングすることで、多様な実行を示し、カバレッジの高いテストケース選択が可能となっている。提案手法で選択したテストケースのカバレッジと実行時間の多様性を現在企業で用いている手法と比較することで、提案手法の有用性が示された。

3つ目は、類似コード片検索結果に対するフィルタリング手法について研究した。類似コード片検索ツールで類似バグを含むコード片を検索した場合に、検索結果内のすべてのコード片を検査すべきであるが、一般に開発リソースには制限がありすべてを検査することは難しい。この問題に対処するため、検索結果に対してクラスタリングを適用してコード片をグループ化し、正解確率の低いコード断片を出力から除外するフィルタリング手法を提案した。この手法を用いることでバグを含む類似のコード断片を集めたデータセットに対して、既存のフィルタリングよりも効率的にバグを含む類似のコード片を検索可能である事が示された。

本論文で得られた知見は、限られた資源でデバッグを実施する必要がある実務者や、効率的なデバッグ手法についての研究者に非常に役立つものである。よって、博士（情報科学）の学位論文として価値のあるものと認める。