

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Supporting Reactive and Proactive Source Code Refactoring: A Context-Based Approach
著者(和文)	Sae-LimNatthawute
Author(English)	Natthawute Sae-Lim
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第11328号, 授与年月日:2019年9月20日, 学位の種別:課程博士, 審査員:佐伯 元司,権藤 克彦,渡部 卓雄,小林 隆志,林 晋平
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第11328号, Conferred date:2019/9/20, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	審査の要旨
Type(English)	Exam Summary

(博士課程)

論文審査の要旨及び審査員

報告番号	甲第		号	学位申請者氏名		Sae-Lim Natthawute	
論文審査 審査員		氏 名	職 名		氏 名	職 名	
	主査	佐伯元司	教授	審査員	林晋平	准教授	
	審査員	権藤克彦	教授				
		渡部卓雄	教授				
		小林隆志	准教授				

論文審査の要旨（2000 字程度）

本論文は「Supporting Reactive and Proactive Source Code Refactoring: A Context-Based Approach」と題し、英文 8 章より構成されている。

第 1 章「Introduction」では、本研究の背景と目的について述べている。プログラムのソースコード中に含まれる Code Smell が理解性や保守性を低下させ、バグの混入などを増加させる問題を指摘している。その上で、Code Smell の現在の検出技術は開発者にとって不要な多くの Code Smell 候補を検出していること、Code Smell の発生を事前に防止するためには発生予測技術が重要であるが、そのような技術がまだ開発されていないことを述べている。本論文は、検出した多くの Code Smell 候補に開発者の視点から順位づけする技術とソースコードの改版履歴から Code Smell の発生を予測する技術を開発することが目的であると述べている。

第 2 章「Background」では、Code Smell、Refactoring などの定義と、Code Smell の既存の検出技術、本論文で使用するソースコードの波及解析技術、情報検索技術を概説している。

第 3 章「A Study on How Developers Filter and Prioritize Code Smells」では、開発者が何を基準に対処すべき Code Smell を選択し、順位づけしているかを実験的手法によって明らかにしている。その結果、課題と関連の深いモジュール（開発者の Context）を重要視しており、これらから開発者が選択、順位づけを行っていることが判明したと述べている。

第 4 章「Context-Based Code Smells Prioritization」では、課題と関連の深いモジュールを検出するために既存の波及解析ツールを使用し関連度を求め、その値で既存の Code Smell 検出ツールが出力した候補を順位づけする手法を提案している。実験により、この手法を評価し、開発者の Context を考慮しない手法よりも提案手法の方が、実際に開発者が対処すべき Code Smell の順位をより正確に求めていることを示している。

第 5 章「Using Context-Based Code Smells Prioritization to Support Referred Context」では、3 章、4 章では修正したモジュールの推測を目指して、課題に関連するモジュールを利用した Code Smell 検出を行っていたが、修正のために参照すべきモジュールも考慮することの有効性を議論している。Myllyn を用いて記録された課題と参照されたモジュール情報を集め、各モジュールに対しそのモジュールを参照している課題数を分析し、その数値に基づいた Code Smell 候補の順位づけの有用性を実験によって確認できたとしている。

第 6 章「An Exploratory Study on Decaying Modules for Proactive Refactoring」では、Code Smell の発生を予測するために、Code Smell の種類ごとにクラスの凝集度などの既存のメトリックスを組み合わせ、その値の増加によって発生を予測する手法を提案している。4 つの開発プロジェクトのリポジトリより、各バージョンのソースコードを取り出し、メトリックス値の増加があるモジュールには、改版が進むと、将来さらなる増加が起これ、Code Smell により近づいていく傾向が高いことを知見として得られたとしている。さらに、この 4 つのプロジェクトに対し、ランダムフォレスト法を用いて Code Smell の予測モデルを学習させ、そのモデルの予測性能を評価することにより、メトリックスを使って Code Smell の発生を予測する手法が有効であることを示している。

第 7 章「Using Automated Impact Analysis Techniques to Improve Decaying Modules Prediction」では、3 章、4 章の成果である開発者の Context を考慮することにより、Code Smell の予測性能を向上できることを述べている。6 章で使ったメトリックスに加え、波及解析ツールで出力したモジュールの関連度を利用して得た予測モデルを用いて、6 章で行った評価実験と同じ実験を行った。その結果、実験対象の 4 つのプロジェクトのうち、3 つのプロジェクトにおいて有意な予測性能の改善が見られたとしている。

第 8 章「Conclusion」では、本論文の結論を総括し、今後の課題について述べている。

以上で述べたように、本論文では、ソースコードの中に含まれる Code Smell を開発者の Context に基づいて順位づけする方法と、改版が進むと将来 Code Smell になる可能性のあるモジュールを検出する手法を述べており、実験を行いその有用性、既存手法に対しての優位性を確認している。Code Smell はソースコードの理解性、保守性、信頼性を低下させることから、本手法で開発した技術は高品質のソフトウェアの開発に関して、学術上貢献するところが多い。 よって我々は、本論文が博士(学術)の学位論文として十分価値があるものと認める。

注意：「論文審査の要旨及び審査員」は、東工大リサーチリポジトリ(T2R2)にてインターネット公表されますので、公表可能な範囲の内容で作成してください。