

Title	Efficient Code Clone Management based on Historical Analysis and Refactoring Support
Author(s)	Hotta, Keisuke
Citation	
Issue Date	
Text Version	ETD
URL	https://doi.org/10.18910/26162
DOI	10.18910/26162
rights	
Note	

Osaka University Knowledge Archive : OUKA

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

Osaka University

論文内容の要旨

〔 題 名 〕

Efficient Code Clone Management based on
Historical Analysis and Refactoring Support
(履歴分析とリファクタリング支援に基づく
効率的なコードクローン管理)

学位申請者 堀田 圭佑

近年、ソフトウェア工学における研究対象としてコードクローンが注目を集めている。コードクローンとはソースコード中に存在する同一の、あるいは類似するコード片のことを指す。コードクローンの主たる生成要因として、コピーアンドペーストによる既存のソースコードの再利用(コードの複製)が指摘されている。コードの複製はソフトウェア開発において習慣的に行われており、ソースコード中に少なからず存在することが報告されている。

一般的に、コードクローンの存在はソフトウェア進化を阻害する要因の1つであるとみなされている。その根拠として、互いにコードクローン関係にあるコード片は同じような修正を必要とする可能性が高い、ということが挙げられる。あるコード片に対して修正を施した際、そのコード片とコードクローン関係にあるコード片が存在すれば、そのコード片に対しても同様の修正を検討する必要があるため、コードクローンの存在によってソースコードの修正により多くのコストが必要となる。また、同様の修正を施す必要があるにも関わらずその修正を施し忘れた場合、いわゆる修正漏れが発生した場合、ソースコード中に不具合が混入する危険性がある。さらに、近年のソフトウェアの大規模化により、ソースコード中に存在するコードクローンの量が膨大となっていることが、コードクローンによってもたらされる弊害の大きさに拍車をかけているといえる。このような理由から、コードクローンはソフトウェア進化を阻害するものとして嫌忌されている。

この定説に基づき、ソースコード中に存在するコードクローンを検出する手法や、それらをソースコード中から除去する手法が多数提案されている。その一方で、コードクローンがソフトウェア進化を阻害している、とする定説が必ずしも成り立っていないという調査報告も存在する。コードクローンの有害性に関する議論は結論を得ていないが、これまでの調査報告を総合すると、すべてのコードクローンがソフトウェア進化を阻害しているとはいえないが、一部そのようなコードクローンが存在している、ということができる。

一部のコードクローンのみがソフトウェア進化を阻害している可能性が高い、ということと、コードクローンの主たる生成要因であるコードの複製によってもたらされる利益が少なからず存在する、ということとを踏まえると、コードクローンの生成を完全に禁止し、ソースコード中にコードクローンを存在させない、というコードクローンへの対処方法は、非現実的であり、かつ非効率的であるといえる。したがって、ソースコード中に存在するコードクローンを効率的に管理するための技法が強く求められているということができる。

本研究の目的は、効率的なコードクローン管理の実現を通じて、効率的なソフトウェア進化の実現を支援することである。この研究目的を達成するために、現在までにどのようなコードクローン管理技法が提案されているのかを知る必要がある。そこで、まずはじめに本研究ではこれまでにに行われてきたコードクローンに関する研究の調査を行う。この文献調査では、コードクローン管理技法を、検出、除去、生成抑止、分析効率化、不具合検出の5つの分類に大別し、それぞれの分類についてこれまでにに行われてきた研究を整理する。

次に、本研究はコードクローンの有害性に関する定量的分析を行う。この分析の目的は、コードクローンがソフトウェア進化を阻害している、という定説が成り立っているのか否かを明らかにすること、またそのようなコードクローンがどの程度存在するのかを明らかにすること、という二点である。

これまでにコードクローンの有害性に関する調査はいくつか行われているが、それらには一般性に乏しいなどの課題点がいくつか存在する。そこで本研究では、これらの課題点を解決した手法を用いた調査を行う。コードクローンがソフトウェアの進化を阻害しているか否かを定量的に分析するために、本研究ではコードクローンの修正されやすさに着目する。コードクローンの弊害は、コードクローンに対して修正が加えられたときに生じるため、コードクローンが修正されやすいのであれば、コードクローンによる弊害が生じる危険性が高いと考えられる。この仮説に基づき、15のオープンソースソフトウェアに対して4つのコードクローン検出手法を用いた調査を行った。その結果、コードクローンはそれ以外の箇所と比較して修正されにくい傾向にあることが明らかになった。すなわち、すべてのコードクローンがソフトウェア進化に悪影響を与えているとはいえない、という結果となった。

この結果から、一部のコードクローンのみがソフトウェア進化を阻害している可能性が高いことが明らかになったが、どの程度のコードクローンがソフトウェア進化を阻害しているのか、という点については明らかになっていない。この点を明らかにするためには、ソフトウェアの開発履歴を分析し、それぞれのコードクローンがどのように進化してきたのかを特定する、すなわちコードクローンの追跡を行う必要がある。これまでにコードクローンの追跡手法は提案されているが、その追跡手法はコード片の移動を伴う修正が加えられた場合に、コードクローンの追跡に失敗する、という課題点を持つ。そこで本研究では、既存のコードクローン追跡手法を改良することで、コード片の移動が生じた場合でもコードクローンを追跡することができる手法を提案する。提案手法をコードクローン追跡システムCTECとして実装し、2つのオープンソースソフトウェアに対する評価実験を通じてその有効性を確認した。

続いて、提案したコードクローン追跡手法を用い、コードクローンの進化に関する分析を行った。この調査の目的は、一部のコードクローンのみがソフトウェア進化を阻害している、ということを定量的に示すこと、またそのようなコードクローンがどの程度存在するのかを明らかにすることにある。コードクローンの生存期間の長さ、並びにコードクローンに加えられた修正の回数に着目した分析を行った結果、ソフトウェア進化を阻害していると考えられるコードクローンは全体の約3%程度であることがわかった。この結果は、一部のコードクローンのみがソフトウェア進化を阻害している、という説を定量的に支持するものとなった。

最後に、本研究ではコードクローン管理支援技法の1つとして、差分を含むコードクローンの除去支援手法を提案する。差分を含むコードクローンはそうでないものと比較して除去作業が困難なものとなるため、その除去を行うためにはツールなどによる支援が不可欠である。本研究で提案する手法はコードクローンの除去が適用可能な候補を自動的に検出して利用者に提示することで、除去候補の発見を支援するとともに、どのようにソースコードを修正すれば除去を行うことができるのか、という情報を提示することで、実際の除去作業の支援を行う。提案する手法をコードクローン除去支援システムCRatとして実装し、オープンソースソフトウェアを用いた実験、並びに被験者の協力を得た実験を通じて、その有効性を確認した。

論文審査の結果の要旨及び担当者

氏 名 (堀 田 圭 佑)			
	(職)	氏 名	
論文審査担当者	主 査	教授	楠本 真二
	副 査	教授	井上 克郎
	副 査	教授	増澤 利光
	副 査	教授	松本 健一(奈良先端科学技術大学院大学 情報科学研究科)

論文審査の結果の要旨

本論文は、コードクローンの効率的な管理を通じて、ソフトウェア保守を支援するための手法についてまとめている。コードクローンとはソースコード中に存在する同一の、あるいは類似するコード片のことを意味し、近年、ソフトウェア工学、特に、ソフトウェア保守の研究分野において注目を集めている研究テーマである。また、コードクローンが原因となり、我が国の大規模ソフトウェア開発プロジェクトが失敗したという事例もあり、実用面においても重要なテーマとなっている。

本論文では、先ず、膨大な関連研究を調査し、コードクローン管理技法を、検出、除去、生成抑止、分析効率化、不具合検出の5つの分類に大別し、整理している。コードクローン管理技法について系統的にまとめたものはほとんどなく、この結果は研究者並びに開発現場の実務家にとっても有用な成果である。

次に、コードクローンの特性として従来から指摘されていた「コードクローンがソフトウェア進化を阻害している」という定説が成り立っているかどうかを15種類のオープンソース・ソフトウェアの開発データを利用して定量的に評価している。評価にあたっては、4種類のコードクローン検出手法を用いて、検出手法による差異を含めた分析を行なっている。結果として、すべてのコードクローンがソフトウェア進化に悪影響を与えているとはいえないが、一部のコードクローンがソフトウェア進化を阻害していると結論づけている。引き続き、ソフトウェア進化を阻害しているコードクローンがどの程度存在するのかを2種類のオープンソース・ソフトウェアの開発履歴情報を詳細に分析することにより、ソフトウェア進化を阻害していると考えられるコードクローンは全体の約3%程度であることを確認した。この結果は、一部のコードクローンのみがソフトウェア進化を阻害している、という説を定量的に支持する結果であり、従来研究では得られていない新たな知見として高く評価できる。

最後に、コードクローン管理支援技法の1つとして、差分を含むコードクローンの除去支援手法を提案している。差分を含むコードクローンは他のコードクローンと比較して除去作業が非常に困難なものであることが指摘されている。従って、その作業を開発者が手作業で実施することは多くの手間が必要となるため、ツールによる支援が不可欠である。提案手法はコードクローンの除去が適用可能な候補を自動的に検出して利用者に提示することで、除去候補の発見支援と、どのようにソースコードを修正すれば除去を行うことができるのか、という情報を提示している。提案する手法をコードクローン除去支援システムとして実装し、オープンソースソフトウェアを用いた実験の結果、ソフトウェアの内容に詳しくない開発者であっても、システムの出力結果を用いて短時間で差分を含むコードクローンの除去が可能であることが確認されている。コードクローンの除去は、開発現場においても重要な課題の一つとなっており、実用面において高く評価される成果である。

以上のように、本論文の成果は、ソフトウェア保守を効率的に支援する手法として、技術面、並びに、実用面において高い貢献があると考えられ、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。