



Title	Study on Licensing and Program Understanding for Reuse Support
Author(s)	鹿島, 悠
Citation	大阪大学, 2015, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/52032
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

論文内容の要旨

氏 名 (鹿島 悠)	
論文題名	Study on Licensing and Program Understanding for Reuse Support (再利用支援を目的としたライセンスとプログラム理解の研究)
<p>論文内容の要旨</p> <p>For developing reliable software, it is important to reuse existing software components, which include open source software. In this dissertation, we addressed the following issues of software reuse: software license, retrieval of reusable components implementing a feature, extraction of a reusable component.</p> <p>Software license permits or forbids the usage of the software including reuse. Therefore, the selection of the software license seems to affect to reuse activity, but not surveyed quantitatively. We studied the impact of software licenses to reuse activity among open source software. In particular, we focused on copy-and-paste activity. This study showed the frequent occurrence of copy-and-paste within the same license products, and the tendency that permissive licensed files are more copied than restrictive licensed ones.</p> <p>Since a developer often retrieve a reusable target using search engines, identifiers which are in the searching target should be named appropriately. Unfortunately, not all developers can do, since a broader knowledge and an experience are necessary. We proposed an approach for building domain specific verb-object relationship dictionaries. The dictionary includes tuples consisting of (Verb, Direct Object, Indirect Object) which are extracted from identifiers in method signatures. In the experiment, we showed that the tuples in the dictionaries are popular in the target domain or common Java programs.</p> <p>After a reusable target is found, reuse will complete if a feature of the target is incorporated into a user's product. For supporting this process, program slicing-based techniques are proposed previously. We performed a comparative study of four backward program slicing techniques for Java. The techniques include static execute before, context-insensitive slicing, hybrid technique of static execute before and context-insensitive slicing, improved slicing. The comparison shows that the hybrid technique has the scalability for analyzing a large system, and outputs a slice whose size is 25 percent smaller than static execute before, which is the most lightweight technique, on average. On the other hand, improved slicing has the scalability for analyzing middle size applications, and is 31 percent smaller than compared to the hybrid technique.</p> <p>We believe that the results of these studies support developers to determinate a license of reusable components, to search a product implementing a target feature, and to extract a reusable component from the product.</p>	

論文審査の結果の要旨及び担当者

氏 名 (鹿 島 悠)			
	(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	八木 康史

論文審査の結果の要旨

ソフトウェア再利用は、信頼性の高いソフトウェアを高速に開発するための手法であり、近年では、オープンソースソフトウェアの発展に伴い再利用対象は増加している。ただし、ソフトウェア再利用は古くから提案されてはいるが、様々な課題が存在しており、実際に再利用を行うのは難しいのが実情である。本論文では、ソフトウェア再利用に関わる3つの課題の解決を目指した。その3つとは、ソフトウェアライセンスの選択、再利用対象の検索、再利用部品の抽出である。

ソフトウェアライセンスは、再利用を含めたソフトウェアの利用について、許可あるいは制限を与える。そのため、ソフトウェアライセンスの選択は再利用活動に影響を与えられと考えられるが、定量的には調査が行われていなかった。そこで著者はオープンソースソフトウェアを対象に、ソフトウェアライセンスが再利用活動に与える影響について調査を行った。特に、基本的な再利用方法であるコピーアンドペースト活動に着目して調査を行った。結果として本研究では、同じライセンスを付けられたファイル間でのコピーアンドペーストが多かったこと、また、制限の緩いライセンスが付けられたファイルの方が制限の強いライセンスよりもコピーアンドペーストされていたことを定量的に確信した。

現在、開発者は再利用可能な対象を、検索エンジンを用いて探すと考えられる。そのため、検索対象の一つであるソースコード中の識別子に対して良い命名が付けられていることが望ましい。ただし、良い命名には開発対象のドメインに関する深い知識やプログラミングに関する熟練が必要であるため、全ての開発者が良い命名ができるわけではない。そこで本研究ではドメイン固有の動詞-目的語の関係を収録した辞書の作成を目指す。この辞書は（動詞、直接目的語、間接目的語）の三つ組を収録しており、それらはメソッドシグネチャから取得したものである。実験では、辞書に収録される関係が対象とするドメイン、またはJavaプログラム共通でよく見られる関係であることを確認した。

再利用対象の機能を持つソフトウェアが見つかった後は、その機能をユーザーのソフトウェアに組み込むことが必要となる。既存研究では、この作業に対してプログラムスライシングを用いた支援を行う手法が提案されている。しかし、プログラムスライシングは様々な手法が提案されているため、プログラムスライシングを使用する開発者や研究者にとって、包括的な比較結果は実際に使用する手法の選択の際に重要な情報となる。本研究では、新しく提案されていた手法を含めて、プログラムスライシング手法の包括評価を行った。比較した手法は、スタティックエグゼキューティブフォア、コンテキストインセンティブスライシング、この二つをハイブリッドした手法、そして、インブルーブドスライシングの4手法である。比較の結果、ハイブリッド手法は大規模なプログラムでも解析できるスケーラビリティを示し、スライスの大きさもスタティックエグゼキューティブフォアのものより25%小さいものとなった。一方で、インブルーブドスライシングは、大規模プログラムは解析できなかったが、中規模のプログラムを解析できるスケーラビリティを示し、ハイブリッド手法のスライスに比べて30%小さいスライスを返すことを示した。

これらの研究成果は、ソフトウェア工学分野において有用なものであり、新規性をもつものである。よって、博士（情報科学）の学位論文としての価値があるものとして認める。