



Title	Study on Analysis of Program Collection for Classifying and Understanding Relations
Author(s)	神田, 哲也
Citation	大阪大学, 2016, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/55841
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 (神 田 哲 也)	
論文題名	Study on Analysis of Program Collection for Classifying and Understanding Relations (プログラム群の分類および関係の理解を目的とした 解析に関する研究)
論文内容の要旨	
<p>Throughout a software development historic-lifecycle, a large amount of software and libraries can be generated. Those outputs are collected or categorized to form a "program collection". This dissertation dealt with two types of program collection. "Program collection with time series" is a set of the programs that have the same origin, but now containing multiple versions through branching and updating. "Snapshot of program collection" is a set of programs that is provided as ready to (re)use in deveoping another program. This category contains applications for specific devices and library set.</p> <p>Well-managed program collections are useful for further development, as a target of software reuse. Maintaining program collections and keeping their value as an asset is an important thing, to prevent them from obsolescence. Thus, analysis technique for understanding their characteristics and revealing hidden relations may be helpful.</p> <p>This dissertation describes four researches on analysis of program collection for classifying and understanding their relations. Each research analysis uses very limited inputs, mostly the program code, and reveals important characteristics of program collections. We believe that these results support developers to understanding existing program collections.</p> <p>In the first research, we present a method to approximate evolution history of product family using only the source code of them. Since the history of product family would be lost in typical cases, it is hard for developers to understand these product family. A proposed method only requires the source code of target product family and clarify the branching and latest versions of the software products. The study showed that about 80% of the edge in the approximated evolution history is consistent with the actual evolution history of the products.</p> <p>In the second research, we present a semi-automatic method to extract features from Android applications. Many Android applications with similar purpose are available, however, those applications are developed by independent developer so it is difficult to compare. The method extracts sequences of API calls from source code of Android applications and consider those sequences as features of applications. A case study showed important differences among applications.</p> <p>In the third research, we examined the quality of Java library set Maven2, the popular Java library repository. Maintaining library set is important, but Java library files can contain another library file and it is invisible for the library users. We meajured the nubmer of nested library files and count duplication of them. Analysis revealed that there are many copies of Java library files among the nested library files.</p> <p>In the fourth research, we compared the characteristics of C and Java library set. Especially this research spotting identifier names, because identifier names are important source for program analysis and comprehension. The analysis of identifier definitions in C and Java library APIs reveals that they have different tendency of definition.</p>	

論文審査の結果の要旨及び担当者

氏 名 (神 田 哲 也)			
	(職)	氏 名	
論文審査担当者	主 査	教授	井上 克郎
	副 査	教授	増澤 利光
	副 査	教授	楠本 真二
論文審査の結果の要旨			
<p>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。</p> <p>本学位論文は、ソフトウェアがその進化と数を増大していく中で、互いに関連するソフトウェアの集合であるソフトウェア群を形成し、その品質維持と管理が重要な課題になっていることを指摘している。ソフトウェア群の特性を明らかにし、維持管理に必要な情報を抽出することを目的として、ソフトウェア群のコードを包括的に解析する方法について以下の研究を行ったことを報告している。</p> <p>第2章では、ソフトウェア群の管理において、どのソフトウェアからどのソフトウェアが作られたかという情報が欠落することを指摘し、ソフトウェア製品群の進化の履歴をソースコードのみを用いて近似する手法を提案している。提案手法ではソフトウェア間の類似するファイルの数に着目することにより、ソフトウェア間の類似度を定義している。実験により、提案手法では類似したソースファイルの数に着目することで、バージョン番号などに頼らず、ソースコードのみを用いて製品群の分岐や各分岐の最新版を特定できることを示している。</p> <p>第3章では、機能的に類似する複数のソフトウェアの比較が利用者にとって重要な問題であることを指摘し、Androidアプリケーションの機能を半自動的に抽出する手法を提案している。提案手法ではソースコード中のAPI呼び出しの列を機能とみなすことで、比較を可能にしている。ケーススタディによって、提案手法がアプリケーション間の重要な機能の差異を検出できることを示している。</p> <p>第4章では、ライブラリ集合の品質維持管理が重要な関心事であることを指摘し、その例としてJavaライブラリの内部でのライブラリファイルの重複問題を挙げている。著名なJavaライブラリのリポジトリであるMaven2に対する調査によって、Javaライブラリの内部で同一のライブラリの同一あるいは異なるバージョンのファイルが重複して存在することを明らかにしている。</p> <p>第5章では、ライブラリの識別子がソフトウェアの解析と理解において重要な役割を果たしていることを指摘し、CとJavaのライブラリにおける識別子名の定義における傾向を調査し比較を行っている。調査の結果、CとJavaのライブラリにおいては識別子名の定義の傾向に差異があることを示し、差異の具体的な内容と調査結果の利用法について考察を加えている。</p> <p>以上の通り、本学位論文は顕著な成果を挙げていることが確認された。また、これらの報告は、リポジトリマイニングなどソフトウェア工学における他の課題においても応用できるものであり、ソフトウェア群に対する解析だけでなくコード解析技術自体の洗練に貢献しているといえる。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。</p>			