Title	Tracking Data Dependence of Large-scale Systemsfor Practical Program Understanding
Author(s)	秦野, 智臣
Citation	大阪大学, 2017, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/67169
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 （ 秦野 智臣 ）	
論文題名	Tracking Data Dependence of Large-scale Systems for Practical Program Understanding (大規模システムの実践的なプログラム理解に向けたデータ依存関係の分析)

論文内容の要旨

Software developers must understand program behavior through code reading for software maintenance. When changing existing features, developers must understand locations of the source code related to the features. When fixing bugs, developers must understand which statements cause the bugs.

Existing studies reported that developers spend a lot of time for program understanding. Developers are often required to read the source code of large-scale systems that are unfamiliar to them. Therefore, many researchers investigated how developers understand programs and developed numerous techniques to help developers understand programs.

Program dependence analysis is one of techniques for helping program understanding. It analyzes the source code to extract read/write relationships of variables (called data dependence), method call relationships, and so on. Developers use these relationships to explore the source code effectively and efficiently.

This dissertation describes studies on dependence analysis techniques for program understanding. These studies aim to extract useful information from the source code and provide it for developers. Furthermore, we released our analysis tool to facilitate future studies on dependence analysis by other researchers.

First, we conducted an empirical study to statistically investigate the effectiveness of thin slicing, which is a variant of program slicing to extract statements that produce data used by a particular statement. Although an existing study showed that thin slicing is useful for program understanding in small cases, it is not clear whether thin slicing is effective for program understanding in general. We computed thin slices with respect to all statements that consume data and measured various metrics on extracted statements. The results showed that the size of the extracted statements is small enough on average. Furthermore, we found that 10% of thin slices can be effective for identifying the source statements of data. We believe that these slices help developers track data dependence.

Second, we developed a novel dependence analysis technique tailored to understanding how outputs of a feature are computed from inputs (called business rules). Existing techniques extract statements that correspond to business rules. However, these techniques may include conditional statements that do not correspond to rules. Our technique excludes those conditional statements by constructing a partial control-flow graph, every path of which outputs a computed result. We evaluated whether this technique actually contributes to the performance of developers who extract business rules. A controlled experiment based on an actual understanding process in one company shows that the technique enables developers to more accurately identify business rules without affecting the time required for the task. This is the first study to apply an automated extraction technique to practical tasks in business-rule understanding.

Finally, we developed a program analysis tool for Java named SOBA. It analyzes intra-procedural control-flow, data dependence, control dependence, method call relationships, and so on. Its design enables to easily obtain the above information without detailed knowledge of program analysis. We compared the functional differences and usage differences between SOBA and existing tools. We also compared the performance of them and showed that SOBA was faster than existing tools. Furthermore, SOBA was applicable to a large-scale system which has over 67,000 classes. We believe that releasing the analysis tool as open source software contributes to future studies of software engineering.

様式 7

論文審査の結果の要旨及び担当者

氏　名　（秦野　智臣）		
	（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
副 査	准教授	石尾 隆（奈良先端科学技術大学院大学）

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，プログラム理解を支援することを目的に，プログラムの静的解析技術に関する以下の3つの研究を行った．

一つ目は，本学位論文2章の，Thin Slicingが一般的に有用であるかどうかの評価を行うためのスライスサイズの計測である．従来研究では，少数のケースでしかThin Slicingの有用性が示されておらず，一般的に有用であるかどうかが不明であった．そこで本研究では，これまでの従来研究にならい，Thin Slicing のスライスサイズを計算した結果，従来のプログラムスライシングより小さいことが分かった．

二つ目は，同3章の，ビジネスルールの理解に向けた依存関係解析手法である．従来技術では，ビジネスルールに対応する文をソースコードから抽出する技術を提案しているが，従来技術はビジネスルールに対応しない条件文も含まれてしまうという課題があった．そこで，本研究ではビジネスルールに対応する条件文を抽出する手法を提案し，その評価を行った．その結果，提案手法はビジネスルールに対応する条件文を正確に抽出でき，開発者の理解作業を正確にすることができるということが分かった．

三つ目は，同4章の，Javaプログラムのプログラム解析を行うツールの開発である．従来のプログラム解析ツールは，解析アルゴリズムに関する詳しい知識がないと，それらを利用することが困難であった．そこで本章では，プログラム解析を容易に行うことができるツールを開発し，それをオープンソースソフトウェアとして公開した．本ツールは従来ツールより実行性能がよく，大規模システムへの適用可能性が高いことが分かった．このような解析ツールは，ソフトウェア工学の発展に寄与するものであると考えられる．

以上のことから，本学位論文は上記課題に対して，積極的に解決策を提案／評価し，顕著な成果をあげていることを確認した．また，プログラム理解における課題を解決し，本分野へ貢献しているものであるといえる．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．