Title	
Author(s)	,
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/12750
Rights	
Description	Supervisor: , ,

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Improving Transformation of Object-Oriented Program to Aspect-Oriented Program with Static Analysis

Lin Wang

School of Information Science,
Japan Advanced Institute of Science and Technology

March, 2015

Abstract

Separation of concerns is expected to be supported by development methodology or programming languages through enabling encapsulation of each different concern in its own unit of modularity. Unfortunately, current object-oriented languages and development methodology fail to provide a complete and effective support for the separation of concerns. Undesirable phenomena such as code scattering and code tangling occur.

Aspect-Oriented Programming (AOP) technique supports the separation of crosscutting concerns. Aspect-orientation proposes a technique to obtain better software modularity in a practical situation where object-oriented development/programming and associated design patterns are not appropriate. There is a prospect of aspect-oriented programming becoming a mainstream technology in the near future. The questions of how to deal with existing legacy object-oriented software system when aspect-oriented becomes standard practice and how to introduce aspects to an existing legacy object-oriented software system are emerged. The traditional object-oriented programming technique has known limitations, which it could not modularize the implementation of crosscutting concerns in existing software systems. Thus, applying AOP technique to transform the legacy object-oriented software system is one way to obtain the benefits of aspect-oriented programming technique.

In this dissertation, we explore the approach to resolve two obstacles in the transformation of object-oriented (OO) program to aspect-oriented (AO) program with static program analysis techniques. First, we propose a new heuristic algorithm for optimizing the combination of input metrics for clustering based aspect mining. Aspect mining is the first phrase in the aspect-oriented refactoring (AOR). Aspect mining aims at detecting the code which is likely to implement a crosscutting concern. Aspect refactoring is the second phrase in the AOR, while it is a way to remove the code implementation of crosscutting concerns from an object-oriented program into aspects. Second, After aspect refactoring, the pointcuts is created with pointcut abstraction rules. The pointcuts are always implemented as name-based or enumeration pointcuts, however, these pointcuts are known as fragility against program evolution. We propose a framework Nataly to solve the fragile pointcut problem by (1) automatic inferring the intention properties from the join points matched by the given name-based pointcuts and (2) generating a method to check whether the given join point satisfies the properties or not. We also give a deep discussion of aspect interference problem in the AOR. In this discussion, we propose an idea to check the aspect interference by using static control flow analysis.

The result of experiments revealed that the metric selection approach increased the accuracy of aspect mining in the aspect mining. The fragile pointcut problems are also alleviated, because the pointcuts which are generated automatically by our framework are more robust than the name-based or enumeration counterpart.

Key Words: **Aspect-oriented refactoring, Aspect mining, Crosscutting concern, Analysis-based pointcut, Static analysis**