Abstract

Aspect Orientation removed the code scattering and tangling drawback of Object Orientation by encapsulating the cross cutting concerns into their own modules called Aspects. It is gaining popularity these days as lot of languages, frameworks, programming and modeling tools already support aspects and developers have started to embrace these. But there exists lot of legacy object oriented code that needs to be moved to aspects as this makes cross cutting concerns easy to change (localized changes would be enough), test, extend, more comprehensible, etc. Converting it manually is tedious and there exist different techniques that semi automate the process making the maintenance engineer’s job easier. Another approach to the automation process using program slicing is also possible. In this paper, we discuss aspect mining and extraction from program slicing point of view.

References
- Filippo Lanubile, Giuseppe Visaggio. Extracting Reusable Functions by Flow Graph-Based Program Slicing. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO. 4, APRIL 1997.
- Raghavan Komondoor, Susan Horwitz. Semantics-Preserving Procedure Extraction. In
Proc. of 27th ACM Symp. on Principles of Programming Languages (POPL), (Boston, Massachusetts, January 2000).
- Tonella, P., Ceccato, M., Migrating Interface Implementation to Aspects, ICSM'04, Chicago, USA, September 2004.

Index Terms
Computer Science
Software Engineering

Keywords
Aspect Mining using Program Slicing
Refactoring to Aspects
Program Slicing for Refactoring
Untangling