On the Classification of Cyclic Dependencies in Java Programs

A thesis presented in partial fulfilment of the
requirements for the degree of

Master of Science
in
Computer Science

at Massey University, Manawatū, New Zealand

Hussain Abdullah A. Al-Mutawa

2013

ABSTRACT

Software engineering guidelines and rules discourage cyclic dependencies between modules, yet empirical studies have shown that many software systems are burdened with them. This might indicate that not all cycles are as detrimental to software quality as previously thought. Clearly, a better understanding of the types of cyclic dependencies and their effect on software quality is required. As a first step in this direction, we look closely at the shapes formed by software dependency graphs containing cyclic dependencies. Such cyclic dependencies correspond to the concept of strongly connected components in graph theory. We propose an approach to classify strongly connected components according to their topologies. This allows us to distinguish between dense and sparse, symmetric and asymmetric structures. We extend on previous studies and investigate the relationship between cyclic dependencies and the package containment tree. We validate our approach with experiments based on a corpus of 103 open–source Java systems. We find that cyclic dependencies tend to form in branches of the package containment tree around parent packages that are not critical according to some researchers.

ACKNOWLEDGEMENTS

I am in debt to my advisors, Jens Dietrich, Catherine McCartin, and Stephen Marsland for supervising me throughout the course of this thesis. This thesis would not have been possible without their guidance and continual help. When I had been doing my undergraduate degree, I once asked Jens: *"why cyclic dependency matters?"* Jens's response was: *"why don't you do some postgraduate research with me and find out for yourself?"* My deepest thanks goes to Jens for motivating me towards doing this research and for his continual and invaluable help in putting this thesis together. My thanks also goes to Catherine and Stephen for being tolerant and supportive and for the warm encouragement and guidance they provided.

My sincere thanks also goes to Janet George, Russell Johnson and Briony Coote for their help in proofreading this thesis.

Last but not least, my heartfelt thanks goes to my wife, for her kindness and support she has shown during the past eighteen months it has taken me to finalise this thesis. I would also like to thank my daughters Ayah and Alaa, and my son Ali, for their patience during my study. Furthermore, I would like to offer my special thanks to my parents for teaching me how to be persistent.

This project has been supported by funding from the Ministry of Higher Education, Saudi Arabia as part of the King Abdullah Foreign Scholarship Program (KASP).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

ADP Acyclic Dependency Principle

ANTLR ANother Tool for Language Recognition

ASPL Average Shortest Path Length

AWT Abstract Window Toolkit

DAG Directed Acyclic Graph

DFS Depth First Search

JAR Java Archive

JRE Java Runtime Environment

JSON JavaScript Object Notation

LOC Lines of Code

MFES Minimum Feedback Edge Set

ORM Object-Relational Mapping

OSI Open System Interconnection

PCT Package Containment Tree

PDG Package Dependency Graph

SCC Strongly Connected Component

SVM Support Vector Machine

UML Unified Modelling Language