SpringerBriefs in Computer Science

More information about this series at http://www.springer.com/series/10028

Márcio Ribeiro · Paulo Borba
Claus Brabrand

Emergent Interfaces for Feature Modularization

Springer

Márcio Ribeiro
Computing Institute
Federal University of Alagoas
Maceió, Alagoas
Brazil

Claus Brabrand
Department of Software and Systems
IT University of Copenhagen
Copenhagen
Denmark

Paulo Borba
Informatics Center
Federal University of Pernambuco
Recife
Brazil

ISSN 2191-5768
ISBN 978-3-319-11492-7
DOI 10.1007/978-3-319-11493-4

ISSN 2191-5776 (electronic)
ISBN 978-3-319-11493-4 (eBook)

Library of Congress Control Number: 2014951726

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To our families

Foreword

Software systems in a wide range of domains need to be increasingly configurable. Program families and software product lines are well-known forms of configurable software systems. Companies are often launching initiatives and new projects involving configurable software systems, as they enable organizations to achieve time-to-market gains and cost reductions. These systems are often decomposed into a set of features, which represent semantically cohesive units of behavior. Features describe the commonalities and variabilities of programs potentially derived from a configurable software system.

Unfortunately, developers still struggle to properly maintain the source code of software families and product lines. These systems typically have many features, and each feature may have from dozens to thousands of lines of code. The implementation of each feature is often scattered through the modules of a program. In other words, the boundaries of features are often not the same as the implementation modules in a program. Hence, a long standing challenge faced by the software maintainers is to identify, buried into hundreds of lines of code, relevant feature dependencies while realizing their changes. Existing techniques and tools lack proper modularity support and do not properly inform the boundaries and feature dependencies in the source code. Even after developers spent a huge effort to figure out how to implement their feature changes, they often miss feature dependencies and introduce bugs in their configurable systems.

Over the past two decades an extensive body of literature has emerged to cover a wide range of topics on software engineering for configurable systems. Researchers and practitioners have gathered a good collective understanding of how to design and implement high-quality configurable systems. However, until recently, our knowledge on how to support software developers on maintaining these systems was still very limited. It is stimulating to see a book that makes a significant improvement to modular maintenance of configurable systems. This book is clearly a distinguished asset in the history of software maintenance for product lines and program families.

During the past 5 years, I have had the honor to meet and closely follow the Doctoral research of Márcio Ribeiro, a talented young researcher, who came up with brilliant ideas on how to tame the complexity of maintaining configurable systems. The book characterizes the notion of emergent feature interface, an essential modularity mechanism to support daily tasks of configurable software maintainers. The on-demand computation of emergent feature interfaces is only possible through feature-sensitive data flow analysis, an innovative technique being described in this book. The description and evaluation of emergent feature interfaces are also inspiring, and treated with the rigor that is characteristic of a mature piece of research.

The several awards granted to Márcio's work over the last years are an evidence of the relevance and impact of his research. You should definitely read this book if you want a thorough coverage of an innovative approach that has the potential to shape next-generation tools for maintaining configurable systems.

Rio de Janeiro, December 2013 Alessandro Garcia

Contents

Acronyms

API	Application Programming Interfaces
AST	Abstract Syntax Tree
CFG	Control-Flow Graph
CIDE	Colored IDE
FODA	Feature-Oriented Domain Analysis
SLoC	Source Lines of Code
SPL	Software Product Line
VSoC	Virtual Separation of Concerns