Niels Streekmann

Clustering-Based Support for Software Architecture Restructuring

VIEWEG+TEUBNER RESEARCH

Software Engineering Research

Herausgeber/Editor:

Prof. Dr. Wilhelm Hasselbring

Im Software Engineering wird traditionell ein Fokus auf den Prozess der Konstruktion von Softwaresystemen gelegt. Der Betrieb von Systemen, die kontinuierlich Dienste mit einer geforderten Qualität bieten müssen, stellt eine ebenso große Herausforderung dar. Ziel der Reihe Software Engineering Research ist es, innovative Techniken und Methoden für die Entwicklung und den Betrieb von nachhaltigen Softwaresystemen vorzustellen.

Traditionally, software engineering focuses on the process of constructing and evolving software systems. The operation of systems that are expected to continuously provide services with required quality properties is another great challenge. It is the goal of the Series Software Engineering Research to present innovative techniques and methods for engineering and operating sustainable software systems.

Niels Streekmann

Clustering-Based Support for Software Architecture Restructuring

With a foreword by Prof. Dr. Wilhelm Hasselbring

VIEWEG+TEUBNER RESEARCH

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Dissertation University of Kiel, 2011

1st Edition 2012

Editorial Office: Ute Wrasmann | Anita Wilke

Vieweg+Teubner Verlag is a brand of Springer Fachmedien.
Springer Fachmedien is part of Springer Science+Business Media.
www.viewegteubner.de

Cover design: KünkelLopka Medienentwicklung, Heidelberg
Printed on acid-free paper
Printed in Germany

ISBN 978-3-8348-1953-6

Foreword

Successful software systems are long-living systems. During their lifetime the underlying development platforms, programming languages and software architectures eventually become obsolete. When this happens, these software systems are often called *legacy systems:* those engineers responsible for maintaining the systems are usually not the engineers who built the systems. An emerging question is how the resulting requirements for modernization may be achieved. As a first step, it is useful to reconstruct the existing software architecture via reverse engineering techniques. If we aim at a real modernization, just a transformation of legacy code into the syntax of a new programming language will not lead to a maintainable system. Therefore, as a second step, the target architecture for the modernized software system needs to be defined. Then, a migration strategy from the legacy systems architecture toward the chosen target architecture is needed.

Niels Streekmann introduces in this research work a new approach to restructuring software architectures, meant to support engineers with modernizing existing legacy systems. From the above-mentioned modernization tasks, the work presented in this book focuses on finding an appropriate migration strategy on the architectural level. To support this task, a semi-automatic method is introduced which groups elements of the reconstructed legacy architecture via clustering algorithms. These clustered groups of source code elements are then mapped to components of the target architecture.

Besides the presented conceptual research work, this book reports on an extensive experimental evaluation of the restructuring approach. In particular, the reconstruction of the architectural elements of a large industrial system is presented. Thus, this book will be a valuable resource for both researchers and practitioners interested in restructuring (legacy) software systems.

Wilhelm Hasselbring

Acknowledgements

During the past five years many people influenced this thesis by providing helpful comments, criticism and reviews to my ideas and publications. I would like to express my deepest gratitude to all of them, even though they may not be mentioned explicitly in the following.

First of all I would like to thank Willi Hasselbring for supervising my thesis and giving me useful hints and suggestions from the selection of the topic to the final completion. I also thank the second examiner of this thesis Andreas Winter for his extensive remarks on reengineering and graph clustering.

My former employer, the OFFIS - Institute for Information Technology, had a vital part in the creation of this thesis by giving me the time to develop and try out my ideas in the context of interesting research projects and to write down this thesis. I thank all my former colleagues and especially all current and former members of the Software Engineering for Business Information Systems Group for the pleasant and productive working atmosphere. I very much appreciated working with you.

Furthermore, I would like to thank all participants of the PhD student seminars (*Doktorandenrunden*) of the software engineering groups of the universities in Oldenburg and Kiel and the former OFFIS division Business Information Management. These seminars have been a source for many helpful remarks and inspiring discussions. I also thank all participants of the talks I gave at other universities, workshops, and conferences and the reviewers of my submitted papers for their expert feedback. Moreover, I would like to thank all people who helped me during the evaluation of my approach by providing assistance, experiment subjects, tools and advice.

However, my biggest thanks go to my family. My parents Ulla and Rolf Streekmann, who made all this possible with their long lasting financial and above all personal support, and my girlfriend Silvia, who endured the temporal constraints in the past five years and who was always there for me with much understanding.

Niels Streekmann

Abstract

The modernisation of existing software systems is an important topic in software engineering research and practice. A part of the modernisation of software systems is the restructuring of their architecture. This has to be done in numerous contexts, including the evolution to service-oriented architectures, the re-establishment of the maintainability of a system or the smooth migration of a system to a new development environment. Architecture restructurings are coarse-grained changes to the internal structure of the system that are performed in temporally limited projects. The planning of the transfer of a existing implementation to the target architecture of a system is currently a mostly manual task. While the analysis of the existing system is supported by e.g. architecture reconstruction approaches, the actual restructuring process is not supported by current approaches.

The MARE approach, which is introduced in this thesis, was developed to provide support for the stepwise restructuring of the implementation towards a target architecture. MARE supports architecture restructurings by semi-automatically creating a complete mapping of elements of the existing implementation to components of the target architecture. The creation of the mapping bases on explicit knowledge about the target architecture and its decomposition criteria. MARE employs graph clustering to implement the creation of the complete mapping.

The MARE Method describes an iterative process model for the overall architecture restructuring process. It emphasises the target architecture as the basis for the architecture restructuring. The iterations of the process model allow for a stepwise restructuring of the system and the integration of human influence on the result of MARE.

The clustering algorithm employed by MARE to create the complete mapping bases on agglomerative hierarchical clustering. It is adjusted to incorporate knowledge about the target architecture. The decomposition criteria are considered by the definition of weights for the different types of dependencies that relate the elements of the existing implementation.

The MARE approach was evaluated in three case studies. These examined the application of MARE in small and middle-sized open source projects as well as for an industrial system with 3.5 million lines of code. The main goal of the evaluation is to show the quality and stability of the clustering algorithm. It furthermore shows the influence factors for the creation of the complete mapping.

Contents

List of Figures

List of Tables