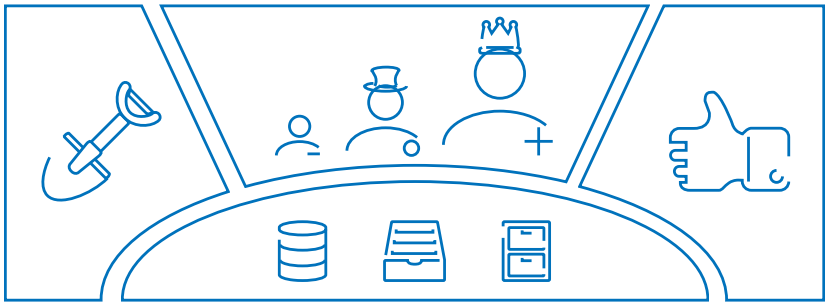
SE | RWTH AACHEN UNIVERSITY

Andreas Ganser

Operation-Based Model Recommenders



Aachener Informatik-Berichte,
Software Engineering

Band 34

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe
Prof. Dr. rer. nat. Horst Lichter

Operation-Based Model Recommenders

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker
Andreas Ganser
aus Mönchengladbach

Berichter: Universitätsprofessor Dr. rer. nat. Horst Lichter
Universitätsprofessor Dr.-Ing. habil. Matthias Riebisch

Tag der mündlichen Prüfung: 6. November 2017

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 34

Andreas Ganser
RWTH Aachen University

Operation-Based Model Recommenders

Shaker Verlag
Aachen 2018

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.

Zugl.: D 82 (Diss. RWTH Aachen University, 2017)

Printed in Germany.

ISBN 978-3-8440-5946-5
ISSN 1869-9170

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

YOU

Abstract

"Reuse is Boring!" states the title of the introductory chapter to IEEE Standard 1517, *Software Reuse Processes*, published at the turn of the century. Later, it states that "[f]or years, we have been hearing about the benefit reuse offers, but have yet to see them realized in practice." This is certainly debatable as a general statement, and success stories in the early 1980s counter this, at least for source code. Still, there must be some reason for such a harsh statement while disregarding types of reuse; after all, loops, methods, and classes are forms of reuse.

For higher-level reuse, i.e., for activities from the preservation until the reutilization of knowledge, we can firstly say that activities involved in reuse do not pay off immediately, but only in the long run. Even worse, all these activities are generally considered tedious, because they expose no immediate benefit. Thus, starting with the harvesting of knowledge and storage for later reuse, continuing with looking for suitable harvested knowledge, i.e., retrieving it, and finishing with reusing (or reutilizing) it, these activities are perceived as rather unappealing. Fortunately, integrated development environments for lower-level reuse, e.g., source code, have already demonstrated how to approach this using completion mechanisms that foster enhanced querying and recommender systems. This places the experience of whole communities at the fingertips of every programmer. Yet, there is no such support for modeling.

Instead, in cases of higher-level reuse, i.e., for activities from harvesting until the reutilization of knowledge, we can secondly state that modelers need to deal with often unorganized information overflow. Fortunately, the abovementioned approaches can help, but need proper information organization. This is known as the challenge of data representation, and can be addressed using a combination of well-suited information representation with a clever retrieval mechanism to enable model reuse that is tightly embedded in tooling.

Further, for higher-level reuse, we can thirdly note that recommended models should be of good quality. Hence, guided assurance in terms of the evolution of harvested models completes the picture of model reuse activities.

Simply put, the abovementioned points are commonly considered to be unappealing activities dealing with challenges denoted by representation, harvesting, evolution, and retrieval. These challenges shall be addressed subsequently. Eventually, we contribute an approach tailored for modeling with UML or models akin to class diagrams, and this approach turns out to be a knowledge-based recommender system based on property graphs and metagraphs suitable for a broader scope. Further, we provide a cookbook for developing such a system, which includes schema for model recommendation production for operation-based model recommenders based on our deployment experiences with HERMES. As we are taking into account contextual information monitored as modeling operations, this too could be denoted as an operation-and-knowledge-based recommender system that (semi-)automates tedious activities.

Kurzdarstellung

Ein einleitendes Kapitel des IEEE Standards 1517 *Softwarewiederverwendungsprozesse* um die Jahrtausendwende heißt „Wiederverwendung ist langweilig!" und später setzt es fort, dass wir „seit Jahren von den Vorteilen hören, die Wiederverwendung bietet und doch muss die Praxis sie erst noch zeigen." Sicherlich ist das als generelle Aussage diskutabel und Berichte aus den frühen Achtzigern zeigen zumindest Erfolge bei Quelltexten. Dennoch muss es einen Grund für solch eine herbe Aussage geben, selbst wenn gewisse Typen von Wiederverwendung ignoriert werden; Immerhin sind Schleifen, Methoden, Klassen und dergleichen auch Formen von Wiederverwendung.

Für Wiederverwendung auf höherem Abstraktionsniveau, d.h. für alle Tätigkeiten Wissen zu konservieren bis hin es als solches wieder einzusetzen, können wir erstens feststellen, dass beteiligte Tätigkeiten sich nicht kurz- sondern nur langfristig lohnen. Es ist gar so, dass sie gewöhnlich als lästig erachtet werden, weil sie keinen direkten Vorteil liefern. Folglich werden diese Tätigkeiten, die mit dem Ernten von Wissen zwecks späterer Verwendung beginnen, sich mit dem Wiederauffinden solches fortsetzen und beim Wiedereinsetzen münden, als eher uninteressant wahrgenommen. Glücklicherweise gibt es integrierte Entwicklungsumgebungen für Wiederverwendung auf niederem Abstraktionsniveau, beispielsweise Quelltext, die veranschaulichen, wie Wiederverwendungsmechanismen mittels spezieller Anfragen und Empfehlungssystemen funktionieren. Damit hat ein Programmierer die Erfahrung von Entwicklergemeinschaften stets unmittelbar zur Hand. Für Modellierer gibt es jedoch nichts Vergleichbares.

Stattdessen können wir zweitens für Wiederverwendung auf höherem Abstraktionsniveau, also alle Tätigkeiten vom Wissen-Ernten bis hin zum Wiedereinsetzen, sagen, dass Modellierer oft mit chaotischem Informationsüberfluss fertig werden müssen. Glücklicherweise können die obigen Ansätze helfen; benötigen aber angemessene Organisation. Diese, typischerweise als Aufgabe der Datendarstellung bekannte Situation, kann nun mit einer Mischung aus wohlstrukturierten Informationen zusammen mit klugen Instrumenten zwecks Wiederfinden angegangen werden und ermöglicht Modellwiederverwendung eng eingebunden in Modellierungswerkzeuge.

Darüber hinaus können wir drittens anmerken, dass empfohlene Modelle von gute Qualität sein sollten. Deshalb vervollständigt angeleitete Betreuung im Sinne von Evolution der geernteten Modelle die Wiederverwendungstätigkeiten.

Nachfolgend gehen wir auf die oben genannten Punkte ein, die üblicherweise als uninteressante Tätigkeiten wahrgenommen werden und die wir einfach gesprochen als Aufgaben hinsichtlich Darstellung, Ernten, Evolution und Wiederauffinden bezeichnen. Letztlich führt das dazu, dass wir einen Ansatz zugeschnitten für UML Modellierung ähnlich zu Klassendiagramme einbringen, der ein wissensbasiertes Empfehlungssystem für Attribut- und Metagrafen umsetzt; aber generischer ist. Zudem stellen wir auf Grundlage unserer Erfahrungen mit HERMES eine Kochanleitung zur Entwicklung solcher bereit, die Schema für Erzeugung von Modellempfehlungen für wissensbasierte Modellempfehlungssysteme enthält. Diese könnten wir letztlich operations- und wissensbasierte Empfehlungssysteme nennen, da sie Umgebungsinformationen in Form von Operationssequenzen betrachten und so lästige Tätigkeiten, sofern sinnvoll, (teil-)automatisieren.

Acknowledgment

The course of this text covers findings from two projects and neglects that they were achieved during a long process and by a tremendous team effort. I would ask whomever has contributed but is not mentioned here to pardon my mistake and accept my sincerest apologies.

First and foremost, my supervisor Universitätsprofessor Dr. rer. nat. Horst Lichter offered me the chance to undertake the endeavor of this research in his department, Research Group Software Construction. I found myself extremely lucky to have an advisor who never got tired of believing in this project whilst providing freedom beyond belief. I also want to thank Universitätsprofessor Dr.-Ing. habil. Matthias Riebisch for being my secondary advisor and providing countless productive suggestions and feedback.

Furthermore, I would like to thank my student workers, bachelor's, master's, and diploma students for their contributions to the HERMES tool. Their efforts and dedication always pushed forward the current state and had an impact on various concepts and realizations. In its final stage, HERMES has the fingerprints of Felix Bohuschke, Stefan Dollase, Andrej Dyck, Christian Fuchs, Niklas Franken, David Mularski, Ramya Nagarajan, Junior Lekane Nimpa, Ruslan Ragimov, Dr. Alexander Roth, Daniel Schiller, Nils Sewing, and Tran Ngoc Viet. Moreover, Roland Hildebrandt, Stefan Hurtz, Christian Kuhl, and Thanabordee Thanarukvudhikorn, though not directly related to these projects, contributed in numerous ways.

I also want to express my sincerest gratitude to my former colleagues Andrej Dyck, Muhammad Firdaus Harun, Dr. Veit Hoffmann, Dr. Simona Jeners, Ana Nicolaescu, Dr. Alexander Nyßen, Malek Obaid, Dr. Chayakorn Piyabunditkul, Dr. Holger Schackmann, Dr. Tanya Sattaya-aphitan, and Dr. Matthias Vianden. They always established a productive, friendly, and encouraging atmosphere at Research Group Software Construction.

In every working group, there are magical fairies doing a job beyond price. We, at Research Group Software Construction, had Bärbel Kronewetter and Marion Zinner backing us up all the time. No matter whether we needed a breakfast during busy times, a helping hand when things got out of hand, or a shoulder to cry on, they were always there for each and every one of us.

Last, but not least, I would like to thank my brother, grandfather, parents, and methyltheobromine for always having my back. You had much more impact on this result than you could possibly imagine and I could possibly express in gratitude.

Andreas Ganser

Contents