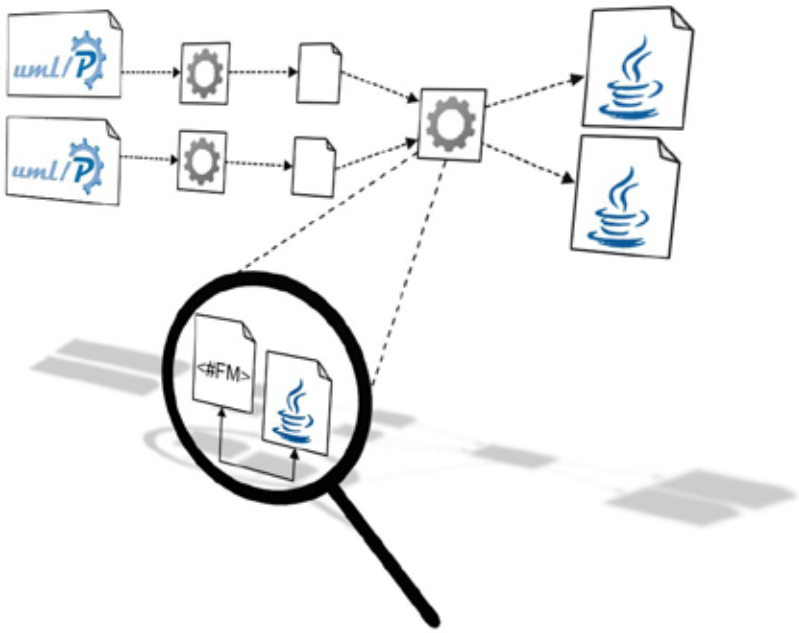
SE | RWTH AACHEN UNIVERSITY

Timo Greifenberg
Steffen Hillemacher
Bernhard Rumpe

Towards a Sustainable Artifact Model

Artifacts in Generator-Based Model-Driven Projects



Aachener Informatik-Berichte,
Software Engineering

Band 30

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

Aachener Informatik-Berichte, Software Engineering

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

Band 30

Timo Greifenberg
Steffen Hillemacher
Bernhard Rumpe
RWTH Aachen University

Towards a Sustainable Artifact Model

Artifacts in Generator-Based Model-Driven Projects

Shaker Verlag
Aachen 2017

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.

Printed in Germany.

ISBN 978-3-8440-5678-5
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

Abstract

Model-driven development (MDD) is an enabler for the automatic generation of programming language files for products or for tests from explicitly defined models. MDD projects manage a large magnitude of artifacts (files, etc.) with various relationships.

A large class of artifact relations comes from artifacts *using* others, e.g., via importing types and signatures. This form of usage strongly differs from *generation* dependencies, where one artifact is generated, compiled, and transformed from or to other artifacts.

An MDD project usually entails a number of potentially dependent process steps, where a chain of artifact generations, compilations, and packagings arises. During these steps a multitude of artifacts are created, read or even executed. Those artifacts are thus related to each other in various ways.

The number and complexity of occurring dependencies and other relationships between development artifacts can lead to several problems, such as poor maintainability and long development times of both, MDD tools and the product, in an MDD process. To tackle these problems, it is important to understand which artifacts are involved and how these artifacts are related to each other in MDD projects.

In this report, we (1) develop an abstract and rather general artifact model and (2) apply the artifact model by examining in detail the kinds of artifacts and related concepts relevant for a form of wide-spread projects, namely Java projects. We also dive into the core of generative projects, by looking at the generator as a set of artifacts executed at design time. Artifacts are regarded as storable and explicitly named elements of MDD projects, such as model files, directories, libraries, and source code files. Thus, artifacts are the physical manifestation of all information in an MDD project.

For a precise definition of all relevant concepts, we introduce the *Artifact Model* (AM), which allows the precise, model-based specification of involved kinds of artifacts, corresponding concepts, and their relations. The AM can also be considered as a specific form of *meta-model* for models representing the concrete elements and relations between these in MDD projects.

Contents