

Aachener Informatik-Berichte, Software Engineering

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

Band 11

Martin Schindler

**Eine Werkzeuginfrastruktur
zur agilen Entwicklung mit der UML/P**

Shaker Verlag
Aachen 2012

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: D 82 (Diss. RWTH Aachen University, 2011)

Copyright Shaker Verlag 2012

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-0864-7

ISSN 1869-9170

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Eine Werkzeuginfrastruktur zur agilen Entwicklung mit der UML/P

Autor: Diplom-Informatiker Martin Schindler

Einer der vielversprechendsten Ansätze, um der steigenden Komplexität von Softwareentwicklungsprojekten zu begegnen, ist die Abstraktion durch Modellierung, für die sich in den letzten Jahren die UML etabliert hat. Diese eher schwergewichtige und in Teilen redundante Familie von Modellierungssprachen wurde mit der UML/P auf wesentliche Kernnotationen reduziert und hinsichtlich der Anforderungen agiler, modellgetriebener Entwicklungsprozesse optimiert. In Kombination mit Java als Aktionsprache erlaubt die UML/P eine vollständige Modellierung und Qualitätssicherung von Softwaresystemen, deren technische Realisierung mittels Generatoren aus den Modellen abgeleitet wird.

Die praktische Anwendung der agilen Konzepte und Modellierungstechniken der UML/P wird mit dieser Arbeit erstmalig durch eine Werkzeuginfrastruktur unterstützt. Zusammengefasst sind die wichtigsten Ergebnisse wie folgt:

- Die *Modellierung* von Struktur, Verhalten und Qualitätssicherung von Softwaresystemen wird durch eine an Java angelehnte textuelle Notation der UML/P unterstützt. Diese bietet eine kompakte, effiziente und werkzeugunabhängige Modellerstellung, eine Strukturierung der Modelle in Pakete, eine Unterstützung von Sprachprofilen sowie explizite Modellbeziehungen durch qualifizierte Namensreferenzen und Importe.
- Die *Kontextanalyse* erlaubt eine modell- und sprachübergreifende Konsistenzsicherung der Modelle und kann flexibel an Sprachprofile oder unterschiedliche Modellabstraktionen angepasst und erweitert werden. Dabei wird insbesondere Java als abstrakte Aktionsprache auf Modellebene angehoben und interpretiert.
- Die *Codegenerierung* wird mit Templates als zentrales Artefakt für Entwicklung, Konfiguration und Ausführung von Generatoren modular bezüglich Sprachen und Modellen umgesetzt. Das Tracing von Generator und dessen Quellen, die Optimierung auf lesbare Templates, die automatisierte Ableitung des Datenmodells und die Anbindung von Java vereinfacht die Anpassung und Erweiterung der Generierung im Projektkontext. Dazu wird eine Methodik zur schrittweisen Ableitung von Generatoren aus exemplarischem Quellcode definiert. Konzepte für die Generierung von Produktiv- und Testcode aus UML/P-Modellen sowie ein Verfahren zur sprachunabhängigen Messung der Testüberdeckung auf Modellen ergänzen diesen Ansatz und demonstrieren, wie aus abstrakten Modellen eine kompositionale, erweiterbare und testbare Architektur für das Zielsystem abgeleitet werden kann.
- Die *Werkzeuginfrastruktur* wird als modulares und leichtgewichtiges Framework umgesetzt. Dabei wird jede Sprache als eigenständige Komponente realisiert, die jeweils die grammatikbasierte Sprachdefinition, Sprachverarbeitung, Kontextbedingungen und Generatoren unabhängig von anderen Sprachen enthält. Dies erlaubt eine flexible Erweiterung und variable Zusammensetzung der Sprachen der UML/P nach dem Baukastenprinzip.

Insgesamt ist damit eine umfassende Werkzeuginfrastruktur zur agilen modellgetriebenen Entwicklung mit der UML/P entstanden, die sich flexibel erweitern und an Technologien sowie Domänen anpassen lässt.