

20 Jahre Softwaretechnik an der TU Braunschweig

Bernhard Rumpe¹, Ina Schaefer², Gregor Snelting³

¹ Software Engineering, RWTH Aachen

² Institut für Softwaretechnik und Fahrzeuginformatik, TU Braunschweig

³ Lehrstuhl Programmierparadigmen, Karlsruhe Institute of Technology

Abstract: Die Softwaretechnik ist einer der essentiellen Kernbereiche der Informatik. Softwaretechnik erarbeitet und lehrt innovative Konzepte, Methoden und Werkzeuge für die ingenieurmäßige Erstellung von großen und komplexen Systemen mit hohem Softwareanteil. Dies beinhaltet Websysteme mit cloud-basierten Servern, wie auch vernetzte eingebettete Systeme und den darauf aufbauenden Möglichkeiten, Mehrwert durch zusätzliche Funktionen zu generieren. In diesem Beitrag wird die Historie der Softwaretechnik an der TU Braunschweig dargestellt, die zu dem heute existierenden Institut für Softwaretechnik und Fahrzeuginformatik geführt hat.

1 Einleitung

Im Jahr 1991 wurde an der TU Braunschweig erstmals das Fach Softwaretechnik etabliert. Zum 1. Oktober 1991 nahm mit der Berufung von Dr.-Ing. Gregor Snelting auf eine C3-Professur die neue Abteilung Softwaretechnologie ihre Arbeit an der Fakultät für Informatik auf. Snelting war damals mit einem Alter von 33 Jahren einer der jüngsten deutschen Informatikprofessoren. Ausgestattet mit 2 Landes- und 2 DFG-Mitarbeitern gelang es der Abteilung, in kurzer Zeit einflussreiche internationale Publikationen zu erarbeiten. Insgesamt wurden in den Jahren 1991-1999 18 Personenjahre Drittmittel von DFG und BMBF eingeworben. In der Lehre wurden insbesondere fakultätsweit softwaretechnische Teampraktika etabliert. 2 Diplomanden bzw Doktoranden des Lehrstuhls (M. Krone 1994, A. Zeller 1997) erhielten den Denert-Preis der GI für die beste softwaretechnische Diplom- bzw Doktorarbeit. Drei Doktoranden der ersten Lehrstuhlgeneration sind heute selbst Professor: A. Zeller (Universität des Saarlandes, Saarbrücken), B. Fischer (University of Southampton), J. Krinke (King's College, London). Im Jahr 1998 erhielt Snelting einen Ruf an die Universität Passau, den er zum 1. April 1999 annahm, nachdem sich die TU Braunschweig ausserstande erklärt hatte, ein Bleibeangebot zu unterbreiten. Snelting leitet heute den Lehrstuhl für Programmiersprachen am Karlsruher Institut für Technologie als Nachfolger von Prof. Goos, einem der Gründerväter der deutschen Informatik.

Danach wurde die Softwaretechnik zu einer vollen Professur (C4) aufgewertet. Dr. rer. nat. Bernhard Rumpe habilitierte sich im August 2003 an der TU München und erhielt im Juli 2003 den Ruf auf diese Stelle. Am 17. Dezember 2003 konnte er die Professur als W3-Professor antreten. Ausgestattet war der Lehrstuhl mit 3,5 wissenschaftlichen Mitarbeitern auf Landesstellen, die bald um Stellen aus DFG-, EU-, BMBF- und vor allem auch



[RSS12] B. Rumpe, I. Schaefer, G. Snelting
20 Jahre Softwaretechnik an der TU Braunschweig
In: H.-D. Ehrich (Hrsg.): 40 Jahre Informatik an der Technischen Universität Braunschweig 1972-2012.
GI-Edition Lecture Notes in Informatics, pp.140-147, 2012.
www.se-rwth.de/publications

aus Industriemitteln ergänzt wurden. Im Frühjahr 2004 wurde das Institut für Software geteilt und dabei das Institut für Software Systems Engineering (SSE) neu gegründet. In der Lehre wurde das Teampraktikum um eine Abschlusspräsentation im Stil einer Mini-Messe mit Preisverleihung erweitert und so der Ansporn und Lerneffekt im Praktikum deutlich erhöht. Das SSE wuchs bis Ende 2008 auf 25 Mitglieder an, die sich thematisch auf Themen der Modellierung, Architektur und Methodik vor allem in den anwendungsnahen Domänen der Automotive Industrie, der Webservices und des Green Energy-Managements fokussierten. Daraus entwickelte sich unter anderem die Teilnahme der TU an der Darpa Urban Challenge (Caroline), die Entwicklung und der Organisation diverser Systeme für die Lehre und der Gründung der Synavision GmbH für Monitoring und Optimierung des Energieverbrauchs von Fabriken. Heute sind mit Prof. S. Maoz, Prof. Chr. Berger (Teamleiter des Projekts Carolo) und Prof. J. Whittle bereits drei ehemalige Mitglieder des SSE Professoren. Im Jahr 2008 erhielt Bernhard Rumpe einen Ruf an die RWTH Aachen, den er zum 1. Januar 2009 annahm. Bernhard Rumpe leitet heute den Lehrstuhl für Software Engineering an der RWTH Aachen als Nachfolger von Prof. Nagl, einem der Gründerväter der deutschen Informatik.

Am 1. April 2012 übernahm Ina Schaefer die Leitung des neugegründeten Instituts für Softwaretechnik und Fahrzeuginformatik (ISF). Das Institut für Softwaretechnik und Fahrzeuginformatik steht in der Tradition der bisherigen Softwaretechnik-Institute der TU Braunschweig. Stärker als zuvor soll das Anwendungsfeld der Softwareentwicklung in Fahrzeugen und insbesondere in modernen Automobilen in den Fokus genommen werden. Die Professur für Softwaretechnik und Fahrzeuginformatik ist eine Eckprofessur im Niedersächsischen Forschungszentrum für Fahrzeugtechnik (NFF). Daher verfügt das Institut über zwei Standorte am Informatikzentrum der TU Braunschweig und am Mobile Life Campus in Wolfsburg in den Räumlichkeiten des NFF.

Im folgenden werden Forschungsschwerpunkte aus den vergangenen Phasen vorgestellt und ein Ausblick auf aktuelle und zukünftige Forschungsthemen gegeben.

2 Phase 1: 1991-1999, Abteilung Softwaretechnologie

Die Abteilung Softwaretechnologie unter der Leitung von Gregor Snelting verfolgte drei Forschungsthemen, die sich alle als sehr erfolgreich erwiesen. Zum einen wurde untersucht, wie mathematische Begriffsverbände für die Analyse und Transformation von Software eingesetzt werden können. Begriffsverbände – die auf G. Birkhoff, R. Wille und B. Ganter zurückgehen – machen verborgene Strukturen in einer Relation zwischen gewissen "Objekten" und "Attributen" sichtbar, wozu ein ausgefeilter und sehr eleganter algebraischer Apparat zur Verfügung steht. Es wurde gezeigt, wie sich modulare Strukturen in Altsystemen (falls vorhanden) mit Begriffsverbänden finden lassen. Später wurde in Zusammenarbeit mit dem IBM T.J.Watson Research Center (Dr. F. Tip) ein Verfahren entwickelt, das mittels Begriffsverbänden eine gegebene Klassenhierarchie so transformiert, dass die Semantik garantiert erhalten bleibt, aber Klassen nur noch wirklich verwendete Members enthalten; dies verbessert die Kohäsion und reduziert die Kopplung in der Klassenhierarchie. Die Ergebnisse wurden auf Top-Konferenzen und -journalen, wie z.B. [ST00], publiziert und oft zitiert.

Des weiteren erforschte A. Zeller Techniken des automatischen Debugging und der automatischen Lokalisation von Programmfehlern. Sein ca. 1998 entstandenes Delta-Debugging brachte ihm 2001 (schon in Passau) den Ruf nach Saarbrücken ein, gewann später den ACM SIGSOFT Most Influential Paper Award, und Zeller wurde 2010 ACM Fellow.

Schließlich begründete die Abteilung auch ein langjähriges Projekt zu Programmanalyse und Program Slicing mit Anwendungen in der Software-Sicherheitsanalyse. Damals redeten nur wenige über „Information Flow Control“ und „Nichtinterferenz“; heute – in den Zeiten permanenter Cyberattacken – sind diese Begriffe von größter Bedeutung. Der Abteilung gelang es, ein Analyseverfahren zu entwickeln, das auf Slicing und Programmabhängigkeitsgraphen basiert, Nichtinterferenz garantiert (also jedes potentielle Sicherheitsleck findet) und deutlich weniger Fehlalarme erzeugt als etwa die bekannten Sicherheitstypsysteme. Dieses Projekt wurde auch später weiterverfolgt. Heute verfügt der Nachfolgelehrstuhl am KIT dank der seinerzeitigen Aktivitäten über eine führende Analysetechnologie für Software-Vertraulichkeits- und Integritätsprüfungen, die große Programme in vollem Java mit beliebigen Threads behandeln kann. Sie nutzt Programmabhängigkeitsgraphen für fundamental neue, durchschlagende Analysen zur probabilistischen Nichtinterferenz, die dank ihrer hohen Präzision in diversen Projekten zur Software-Sicherheit angewendet werden.

3 Phase 2: 2003-2008, Institut für Software Systems Engineering

Die Aktivitäten des Instituts für Software Systems Engineering (SSE) wurde in der zweiten Phase zwischen 2003 und 2008 geprägt von der Mission zur

Verbesserung der Softwareentwicklung

durch die Identifikation von *Methoden, Konzepten* und *Werkzeugen* zur *besseren* und *schnelleren* Entwicklung von Softwaresystemen, so dass in *kurzer Zeit* und mit *flexibler Einarbeitung* von sich *wandelnden Anforderungen* ein *qualitativ hochwertiges* Ergebnis entsteht.

Wichtig war dabei insbesondere auch der Nachweis der Tragfähigkeit der entwickelten Methoden und Konzepte durch ihre Anwendung auf praxisrelevante und daher meist große Softwaresysteme in innovativen Anwendungsbereichen. Deshalb wurden neben Grundlagenprojekten immer auch Anwendungsprojekte durchgeführt, die gleichzeitig zu neuen Forschungsideen geführt haben.

Modellierungsinfrastruktur MontiCore. Zur Erfüllung des Mission Statements gehört der Einsatz von Modellierungstechniken in agilen Projekten. Dies ist möglich, wenn die Infrastruktur für die Erstellung und Nutzung von Modellen die Effizienz der Entwicklung unterstützt, die Verbindung zum Code durch Generierung oder Synthese erlaubt, die einfache Definition von Tests ermöglicht und komplexe Analysen auf Modellebene anbietet, um schnell Feedback über mögliche Fehlerquellen zu erhalten. Leider unterstützen heute verbreitete UML-Werkzeuge die agile Entwicklung zumeist sehr unzureichend. Deshalb wurde mit der Modellierungsinfrastruktur **MontiCore** [GKR⁺06, Kra10] eine Plattform gestartet, die es Forschern und Entwicklern erlaubt, schnell und effizient projektspezifisch

anpassbare Modellierungswerkzeuge zu erarbeiten und in Projekten einzusetzen. MontiCore ist heute erfolgreich im Einsatz zur Modellierung von Teilen von Softwaresystemen, die Datenstrukturen, Abläufe, Kommunikation, z.B., zwischen Client und Server und weitere systematisierbare Aspekte in modernen Softwarearchitekturen realisieren. Zu diesem Zweck können Modellierungssprachen (Frontend) sowie die Generierung für die Zielsprachen und die dabei zu verwendenden Technologiestacks (Backend) entkoppelt und dadurch unabhängig voneinander weiterentwickelt werden. So wird eine agile, evolutionäre Entwicklung auf Basis von Modellen ermöglicht. MontiCore erlaubt insbesondere auch die Definition von domänenspezifischen Sprachen, die typischerweise sehr kompakt sind und genau die benötigten Informationen darstellen können, um ein bestimmtes Problem zu lösen. Dies sind häufig Konfigurationssprachen oder Sprachen zur Definition von Policies.

Das in Braunschweig gestartete und in Aachen deutlich ausgebauten Projekt **MontiCore** bietet heute eine Infrastruktur für die UML/P [Rum11, Sch12], webbasierte Informationssysteme [DRRS09], automotiv Funktionsnetze bzw. verteilte Architekturen [HRR12], C++, Java und eine Reihe weiterer, oft domänenspezifische Sprachen an. Besondere Alleinstellungsmerkmale von MontiCore gegenüber anderen Werkzeugen ist die Möglichkeit zur modularen Modell- und Sprachkomposition [Völ11], die auch Kontextbedingungen und die Synthese von ausführbarem Code beinhaltet, sowie die explizite Darstellung von Sprachvarianten [Grö10] und komplexe Analysen, wie die semantische Differenzbildung von Modellen [MRR11]. Aktuelle Arbeiten adressieren die Integration von Variantenmanagement und Modellierungssprache, zum Beispiel beim der Delta-Modellierung von Architekturen [HRRS11]. Siehe auch <http://monticore.de/>.

StuBS - EDV in Studium und Lehre. Mit der Umstellung auf das Bachelor/Master-System wurde eine umfangreiche Renovierung der Lehre-bezogenen EDV notwendig. Die bisher eingesetzte Lösung konnte weder die Organisation neuer Studiengänge auf Basis des Bachelor/Master-Systems noch eine adäquate Kommunikation zwischen Studierenden, Lehrenden und der Verwaltung darstellen. Deshalb wurde die Arbeitsgruppe *StuBS* (Studium an der TU BS) mit Finanzmitteln der TU am SSE im Jahr 2006 gegründet und ab 2008 gemeinsam mit dem Institut für Wirtschaftsinformatik, Abt. Informationsmanagement (WI2), fortgeführt. Die Arbeitsgruppe hat zahlreiche Projekte durchgeführt und Produkte bzw. Komponenten TU-weit ausgerollt. Beispielsweise wurden das elektronische Modulhandbuch (MHB), die Prüfungsanmeldung, die Noteneingabe, der Bewerbungsprozess für zulassungsbeschränkte Studiengänge, der Seminarbelegungsprozess, der Stipendienprozess und das Antragssystem für die Finanzierung aus Studienbeiträgen online realisiert, das Alumni-Portal eingeführt, eine Statistik-Komponente entwickelt, elektronische Lehr/Zugnis-Bescheinigungen eingeführt, und das integrierte Identity Management an der TU mit vorbereitet.

Besonders hervorzuheben ist das MHB, das entwickelt und an ein existierendes Veranstaltungssystem angebaut werden musste, weil es kein alternatives System gab, um die für Bachelor- und Masterstudiengänge notwendigen Modulhandbücher universitätsübergreifend redundanzfrei zu definieren. Sukzessive wurde dieses System erweitert, um die Hörsaalvergabe und die Terminplanung zu integrieren, Sonderveranstaltungen in den Vorlesungskanon einzubauen und letztlich damit das jeweils nächste Semester zu planen. Das

MHB erlaubt zum Beispiel den verbindlichen Import und Export von Veranstaltungen über Fakultätsgrenzen hinweg und hat deutlichen Einfluss auf die Definition der Rahmenprüfungsordnung bis hin zu einer Effizienzsteigerung bei der redundanzfreien Anpassung von Curricula.

Bei den genannten Projekten war nicht nur die softwaretechnische Umsetzung der Anbindung an das Altsystem der Firma HIS, sondern insbesondere die adäquate, agile Einbindung der unterschiedlichsten Statusgruppen und ihrer Interessenslagen eine besondere Herausforderung. Die erfolgreiche Einführung neuer EDV-Systeme in eine Verwaltung erfordert immer ein präzises Verständnis dieser Geschäftsprozesse und die Einbindung der daran Beteiligten bei der Umgestaltung von Prozessen, Zuständigkeiten und Verantwortlichkeiten. Dies ist nur durch die präsidiale Unterstützung weitgehend gut gelungen. Aufgrund der finanziellen und organisatorischen Rahmenbedingungen rückten (ganz im Sinne des genannten Mission-Statements) zunächst vor allem technische Qualität, z.B. im Sinne von Ausfallsicherheit, und rechtzeitig verfügbare Funktionalität in den Fokus, während bei der Bedienbarkeit Abstriche gemacht werden mussten. Leider ist es aufgrund der Rahmenbedingungen nicht mehr gelungen, hier zu optimieren, obwohl sichtbar wurde, dass die zunächst anvisierte Betriebsdauer der entwickelten Lösungen deutlich verlängert werden wird. So ist heute zum Beispiel MHB weiter intensiv im Einsatz.

Agile Entwicklung auf Basis vordefinierter technischer Bausteine und angepasster Entwicklungswerkzeuge war ein Kernelement dieser Projekte. Unter <http://stubs.tu-bs.de/> finden sich weitere Informationen zu diesem Projekt. Unter <http://sselab.de/> gibt es zudem Informationen zur serviceorientierten Entwicklungsumgebung, wie sie in diesen Projekten zum Einsatz kam.

Caroline in der Darpa Urban Challenge 2007. Die Teilnahme der TU Braunschweig ist sicher eines der Highlights der Informatik. Unter der Leitung des SSE haben sich insgesamt 28 Studierende, Mitarbeiter und Professoren zusammengefunden, um in eineinhalb Jahren Caroline, das völlig autonom fahrende Fahrzeug zu entwickeln, aufzubauen und an der Darpa Urban Challenge 2007 in den USA teilnehmen zu lassen. Beteiligt waren ein Institut der Elektrotechnik, ein Institut des Maschinenbaus und drei Institute der Informatik. Sie alle haben neben Aufbau und Absicherung vor allem daran mitgearbeitet, die Software zu erstellen.

Neben den vielen organisatorischen Aufgaben, die damit begannen die Finanzierung sicherzustellen, war Caroline aus Sicht des SSE vor allem ein cyber-physisches Integrationsprojekt. Die vielen einzelnen Komponenten der einzelnen Teams (Vision, Spur-, Hinderniserkennung, Aktorik, GPS, Radar, Mission Control, Monitoring, Sicherheitsschaltung, etc.) mussten unter harten Deadlines integriert und qualitätsgesichert werden. Zu diesem Zweck wurde am SSE eine eigenständige, agile Methodik entwickelt, die sich speziell für Forschungssoftware in eingebetteten oder cyber-physischen Systemen eignet und in der Qualitätssicherung sehr stark auf automatisiertes Testen durch Simulation des Systemkontexts beruht. Nur so war es möglich, in der sehr begrenzt zur Verfügung stehenden Zeit und mit nur einem Versuchsträger (Caroline) eine virtuelle Absicherung durch tausende simulierte, virtuelle Fahrsituationen mit beliebigen Kreuzungen und Verkehr zu ermöglichen. Der damalige Teamkoordinator und heutige Professor Dr. Chr. Berger hat

die damals entwickelte Vorgehensweise in [Ber10] ausgebaut und professionalisiert. Unter <http://carolo.tu-bs.de/> finden sich weitere Informationen zum Caroline-Projekt und unter <http://sse-tubs.de/press/> die Caroline-Pressemappe.

4 Ab 2012: Institut für Softwaretechnik und Fahrzeuginformatik

Das Institut für Softwaretechnik und Fahrzeuginformatik vereint die grundlagenorientierte Forschung auf dem Gebiet der Softwaretechnik mit der anwendungsbezogenen Forschung im Bereich der Fahrzeuginformatik. Das Ziel der Forschungsarbeit ist die Entwicklung von Methoden und Techniken zur Erhöhung der Softwarequalität und zur Steigerung der Effizienz in der Softwareentwicklung. Die entwickelten Vorgehensweisen und Konzepte sollen insbesondere für die Softwareentwicklung in Fahrzeugen, wie modernen Automobilen, Zügen, Flugzeugen oder Satelliten, angewendet werden.

Forschungsschwerpunkte am ISF. Die Forschungsschwerpunkte am Institut für Softwaretechnik und Fahrzeuginformatik sind:

- Modellbasierte Entwicklung und Analyse von Softwaresystemen; der Fokus sind Modellierungssprachen und -methoden, um die anwendungsbezogene Softwarebeschreibung und formale Analyse bereits in frühen Entwicklungsphasen zu unterstützen.
- Entwurfs- und Implementierungskonzepte für diverse Softwaresysteme; das Ziel ist die modulare Repräsentation von Softwarevarianten und -versionen, sowie die Darstellung von Laufzeitvariabilität mit integrierten Korrektheitsgarantien.
- Qualitätssicherung von Softwaresystemen durch Integration von formalen Methoden in Softwareentwicklungsprozesse; im Besonderen wird die Entwicklung von skalierbaren Test- und Verifikationsverfahren für variantenreiche Softwaresysteme vorangetrieben.

Aus den vielfältigen Arbeiten am Institut für Softwaretechnik und Fahrzeuginformatik sollen im folgenden zwei Projekte herausgegriffen und als Beispiele für aktuelle Forschungsthemen dargestellt werden. In diesen beiden Projekten stehen variantenreiche und langlebige Softwaresysteme im Mittelpunkt. Der Variantenreichtum und die ständige Weiterentwicklung und Anpassung von Software sind Phänomene, die in der Zukunft immer stärker an Bedeutung gewinnen werden. Grund dafür ist dass, Software zunehmend an Nutzerbedürfnisse angepasst wird und jahrzehntelang in Betrieb ist. Durch diese Phänomene steigt die Komplexität der Software in der Erst- und Weiterentwicklung, so dass softwaretechnische Methoden, Konzepte und Werkzeug benötigt werden, um die inhärente Komplexität handhabbar zu machen.

Delta-Modellierung zur modularen Variabilitätsbeschreibung. Die Variabilität von Software wird gegenwärtig in der industriellen Praxis durch sogenannte 150%-Modelle dargestellt. Dabei werden in einer gemeinsamen Darstellung alle möglichen Softwarevarianten antizipiert, so dass alle Bestandteile der Software enthalten sind, die in irgendeiner Systemvariante vorkommen können. Für eine bestimmte Variante werden dann die nicht notwendige Teile entfernt. Dieses Beschreibungsverfahren führt jedoch zu einer

großen Unübersichtlichkeit, insbesondere wenn die beschriebenen Systemvarianten verändert werden sollen.

Seit 2009 arbeitet Ina Schaefer daher an modularen Variabilitätsmodellierungskonzepten, welche kompakt, aber dennoch flexibel Varianten und Versionen von Softwaresystemen beschreiben können. Das Ergebnis dieser Forschungsarbeit ist das Konzept der *Delta-Modellierung* [SBB⁺10]. Dabei wird eine Menge von Softwaresystemen mit wohldefinierten Gemeinsamkeiten und Unterschieden durch ein ausgewähltes Kernsystem und eine Menge von Deltas beschrieben. Ein Delta enthält eine Menge von Änderungen des Kernsystems. So kann ein Delta z.B. Systembestandteile hinzufügen, entfernen oder modifizieren. Durch Anwendung einer ausgewählten Teilmenge von Deltas in einer festgelegten Ordnung können so aus dem Kernsystem weitere Systemvarianten erzeugt werden.

Die Delta-Modellierung kann als Variabilitätsmodellierungskonzept durchgängig durch alle Software-Entwicklungsphasen eingesetzt werden. Ihr Vorteil ist, dass in einem Delta Veränderungen modular gekapselt sind und damit die Nachvollziehbarkeit und Übersichtlichkeit bei der Variantenbeschreibung erhöht wird. Wenn Varianten verändert werden sollen, kann dies einfach durch die Ergänzung weiterer Deltas erfolgen. Bisherige Anwendungsbereiche für die Delta-Modellierung sind variable Softwarearchitekturen [HRRS11] oder variantenreiche Java-Programme [SBD11].

Effizienter Test von Varianten und Versionen. Eine besondere Herausforderung stellt ein hohe Anzahl von Varianten und Versionen für die Qualitätssicherung von Softwaresystemen dar [SH11]. Wenn ein System in vielen Varianten existiert, ist es oft nicht praktikabel, alle Varianten einzeln zu prüfen. Gleiches gilt, wenn sich ein System verändert. Hier ist es oft sehr aufwändig, dass veränderte System grundlegend neu zu betrachten.

Um den Test von Softwaresystemen mit vielen Varianten und Versionen effizienter zu gestalten, entwickelt Ina Schaefer seit 2011 gemeinsam mit dem Institut für Programmiersprachen und reaktive Systeme modellbasierte Testverfahren für variable Softwaresysteme [LSKL12]. Grundidee dieser Arbeit ist es, die durch Deltas beschriebenen Unterschiede zwischen verschiedenen Systemvarianten und -versionen für die Testplanung auszunutzen. Dabei sollen bei einer Veränderung in einer Variante oder Version nur noch die Unterschiede geprüft werden, während der unveränderte Teil nicht mehr geprüft werden muss. Fallstudien zeigen, dass dadurch die Anzahl der notwendigen Überprüfungen signifikant reduziert werden kann. In diesem Zusammenhang wurde die Masterarbeit von Sascha Lily, die sich mit dem effizienten Komponententest für variantenreiche Softwaresysteme beschäftigt, mit dem Preis der GI-Fachgruppe „Test, Analyse und Verifikation“ für die beste Masterarbeit im Jahr 2011 ausgezeichnet.

Literatur

- [Ber10] C. Berger. *Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles*. Aachener Informatik-Berichte, Software Engineering, Band 6. Shaker Verlag, 2010.
- [DRRS09] M. Dukaczewski, D. Reiss, B. Rumpe, and M. Stein. MontiWeb - Modular Development

of Web Information Systems. In *Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM'09)*, 2009.

- [GKR⁺06] H. Grönniger, H. Krahn, B. Rumpe, M. Schindler, and S. Völkel. MontiCore 1.0 - Ein Framework zur Erstellung und Verarbeitung domänenspezifischer Sprachen. Technical Report Informatik-Bericht 2006-04, Software Systems Engineering Institute, TU Braunschweig, 2006.
- [Grö10] H. Grönniger. *Systemmodell-basierte Definition objektbasierter Modellierungssprachen mit semantischen Variationspunkten*. Aachener Informatik-Berichte, Software Engineering, Band 4. Shaker Verlag, 2010.
- [HRR12] A. Haber, J. Ringert, and B. Rumpe. MontiArc - Architectural Modeling of Interactive Distributed and Cyber-Physical Systems. Technical Report AIB-2012-03, RWTH Aachen University, Technical Report, February 2012.
- [HRRS11] A. Haber, H. Rendel, B. Rumpe, and I. Schaefer. Delta Modeling for Software Architectures. In *Workshop zur Modellbasierten Entwicklung von eingebetteten Systemen (MBEES)*, 2011.
- [Kra10] H. Krahn. *MontiCore: Agile Entwicklung von domänenspezifischen Sprachen im Software-Engineering*. Aachener Informatik-Berichte, Software Engineering, Band 1. Shaker Verlag, 2010.
- [LSKL12] M. Lochau, I. Schaefer, J. Kamischke, and S. Lity. Incremental Model-based Testing of Delta-oriented Software Product Lines. In *6th International Conference on Tests & Proofs (TAP)*, 2012.
- [MRR11] S. Maoz, J. Ringert, and B. Rumpe. Semantically Configurable Consistency Analysis for Class and Object Diagrams. In *Model Driven Engineering Languages and Systems (MODELS 2011)*, Wellington, New Zealand, LNCS 6981 (Received Best Paper Award as well as ACM Distinguished Paper Award at the MODELS 2011 Conference), 2011.
- [Rum11] B. Rumpe. *Modellierung mit UML*. Springer, 2. edition, 2011.
- [SBB⁺10] I. Schaefer, L. Bettini, V. Bono, F. Damiani, and N. Tanzarella. Delta-oriented Programming of Software Product Lines. In *Intl. Software Product Line Conference (SPLC)*. Springer, 2010.
- [SBD11] I. Schaefer, L. Bettini, and F. Damiani. Compositional Type-Checking for Delta-oriented Programming. In *Intl. Conference on Aspect-oriented Software Development (AOSD'11)*. ACM Press, 2011.
- [Sch12] M. Schindler. *Eine Werkzeuginfrastruktur zur Agilen Entwicklung mit der UML/P*. Aachener Informatik-Berichte, Software Engineering, Band 11. Shaker Verlag, 2012.
- [SH11] I. Schaefer and R. Hähle. Formal Methods in Software Product Line Engineering. *IEEE Computer*, 44(2):82–85, February 2011.
- [ST00] G. Snelting and F. Tip. Understanding class hierarchies using concept analysis. *ACM Trans. Program. Lang. Syst.*, 22(3):540–582, 2000.
- [Völ11] S. Völkel. *Kompositionale Entwicklung domänenspezifischer Sprachen*. Aachener Informatik-Berichte, Software Engineering, Band 9. Shaker Verlag, 2011.