

Jahresbericht 2016

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Softwaresystemtechnik
an der Universität Potsdam

Jahresbericht 2016

Fachgebiet Systemanalyse und Modellierung
Hasso-Plattner-Institut für Softwaresystemtechnik
Universität Potsdam



Fachgebiet *Systemanalyse und Modellierung*
Hasso-Plattner-Institut für Softwaresystemtechnik GmbH
Universität Potsdam
Prof.-Dr.-Helmert-Str. 2-3, D-14482 Potsdam
Leitung: Prof. Dr. Holger Giese

<http://www.hpi.de/giese>

Inhaltsverzeichnis

1	Personelle Zusammensetzung / Staff	1
2	Lehrveranstaltungen / Courses	3
2.1	Vorlesungen / Lectures	3
2.2	Übungen/Projekte / Projects	3
2.3	Seminare / Seminars	3
3	Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations	4
3.1	Betreuung von Masterprojekten / Supervised Masterprojects	4
3.1.1	Masterprojekte / Masterprojects	4
3.2	Betreuung von Masterarbeiten / Supervised Masterthesises abgeschlossen/finished 2016	5
3.3	Betreuung von Dissertationen / Supervised PhDs	5
3.3.1	Abgeschlossene Dissertationen / Finished PhDs	5
3.3.2	Laufende Dissertationen / Running PhDs	5
4	Bearbeitete Forschungsthemen / Research Topics	6
4.1	Eine Modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme	6
4.2	Human-in-the-loop mechanism for Locating Software Faults	6
4.3	Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten	7
4.4	Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems . . .	8
4.5	Utility-driven Architecture-based Self-adaptive Systems	8
4.6	Towards Graph-based Runtime Monitoring for Large Systems	9
4.7	Testing for Self-Adaptive Software Systems	9
5	Drittmittelprojekte / Third-Party funded Projects	10
5.1	DFG – Graduiertenkolleg SOAMED	10
5.2	DFG – Korrekte Modelltransormationen (KorMoran) – Fortsetzungsprojekt . . .	10
5.3	Quantum	11
6	Forschungskooperationen / Research Cooperations	12
6.1	Kooperationspartner aus der Wissenschaft / Partners from Research Institutions .	12
6.2	Kooperationspartner aus der Wirtschaft / Partners from Industry	12
7	Publikationen / Publications	13
7.1	Zeitschriftenartikel / Journals	13
7.2	Beiträge zu Büchern und Sammlungen / Books and Collections	13
7.3	Begutachtete Konferenz- und Workshopartikel / Peer-Reviewed Conference and Workshop Papers	13
7.4	Technische Berichte / Technical Reports	14

8	Vorträge / Talks	15
8.1	Eingeladene Vorträge / Invited Talks	15
8.2	Vorträge auf Konferenzen und Workshops / Talks at Conferences and Workshops	15
9	Web-Portale und -Services / Web-Portals and Services	16
9.1	Self-adaptive.org	16
9.2	MDELab.org	16
9.3	CPSLab.org	16
10	Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten / Memberships, Committee- and Reviewing-Activities	17
10.1	Mitgliedschaften / Memberships	17
10.2	Mitarbeit in Programmkomitees / Activities in Committees	17
10.3	Organisation von Tagungen und Workshops / Organisations of Conferences and Workshops	19
10.4	Gutachtertätigkeiten / Reviewing Activities	19
10.4.1	Forschungsprojekte / Research Projects	19
10.4.2	Zeitschriften und Magazine / Journals	19

1 Personelle Zusammensetzung / Staff



Leiter des Fachgebiets / Director

Prof. Dr. Holger Giese

Sekretariat / Secretary

Kerstin Miers

Senior Researcher

Dr. Leen Lambers

Postdocs

Dr. Dominique Blouin (bis 31.7.2016)

Wissenschaftliche Mitarbeiter / Research Assistant

Johannes Dyck, M.Sc.

Dipl.-Inform. Joachim Hänsel

Dipl.-Inform. Maria Maximova

Lucas Sakizoglou, M.Sc.

Dipl.-Inform. Sven Schneider

Dipl.-Wirtsch.Inf. Thomas Vogel

Sebastian Wätzoldt, M.Sc.

Thomas Beyhl, M.Sc.

PhD-Stipendiaten / Scholarship Holders

Christian Adriano

Dipl.-Wirtsch.Inf. Thomas Brand

Sona Ghahremani, M.Sc.

Studentische Hilfskräfte / Student Assistants

Matthias Barkowsky

Robert Fruth

Lukas Pirl

Christian Zöllner

Christian Nicolai

Tobias Dürschmid

Paul Geppert

Marianne Thieffry

Konstantin Harmuth

Maximilian Söchting

Micheal Fabian

Tobias Knöschke

Christopher Weyand

Leon Masopust

2 Lehrveranstaltungen / Courses

2.1 Vorlesungen / Lectures

Sommersemester / Summerterm 2016

- Modellierung II
- Modellgetriebene Softwareentwicklung

Wintersemester / Winterterm 2016/2017

- Software Engineering for Embedded Systems
- Software Engineering for Self-Adaptive Systems
- Software Testen, Analysieren und Verifizieren

2.2 Übungen/Projekte / Projects

Sommersemester / Summerterm 2016

- Modellierung II
- Modellgetriebene Softwareentwicklung

Wintersemester / Winterterm 2016/2017

- Software Engineering for Embedded Systems
- Software Engineering for Self-Adaptive Systems
- Software Testen, Analysieren und Verifizieren

2.3 Seminare / Seminars

Sommersemester / Summerterm 2016

- Trends in Software Engineering for Self-Adaptive Software
- Modellgetriebene Validierung und Verifikation von Software
- Graphendatenbanken

Wintersemester / Winterterm 2016/2017

- Trends in Software Engineering for Self-Adaptive Software

3 Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations

3.1 Betreuung von Masterprojekten / Supervised Masterprojects

3.1.1 Masterprojekte / Masterprojects

- *Rebuilding and Maintaining Architecture Traces*

Betreuer: Prof. Dr. Holger Giese, Dr. Dominique Blouin, Johannes Dyck

Studierende: Matthias Barkowsky, Melanie Schneider

Abstract: Traceability within software development is the ability to establish and maintain the relationships between artefacts of various development phases, such as determining for a component of a software architecture from which requirement(s) it originates. While this information is essential for determining the impact of changing requirements on the actual system, and conversely, each architectural element is linked to a set of associated requirements from which it originated. However, over time often architectures evolve to meet new requirements or to improve their quality attributeing the impact of changing requirements on the actual system, it is also a vital part of the certification processes that safety-critical systems must undergo to ensure that no malfunction will lead to catastrophic consequences or endanger human lives.

Maintaining accurate tracing information between complex systems and their requirements is currently a major problem. Traceability needs to be updated for every change that occurs in either the requirement or the architecture of the system or both. Under ideal circumstances, a requirement specification, which consists of a set of individual requirements can be completely traced forward and backwards to elements of a software architecture that fulfill them. Each requirement is linked to a set of associated architectural elements involved in aims such as maintainability, performance, resilience, etc. For example, component interfaces may be relocated to ease access, or components may be split or restructured to improve adaptability and evolvability. While these changes do not (or at least should not) affect the functional behaviour, the trace information between requirements and architecture elements can be damaged and it may no longer be possible to determine for each architecture element to which requirement(s) it belongs, and for requirements by which architecture elements they are realised.

The goal of this project is to develop a method to rebuild and maintain the traces between requirements specifications and system architectures when at least one of them evolves. In a first step, the scenario where traces have been eroded following the refactoring of an architecture will be considered. Assuming that the original requirements and architecture specifications and their trace relations are still available, specific insights can be used to rebuild the traces. In a second step, the scenario where the requirements specification is modified after the refactoring of the software architecture will be considered, and the developed approach will be integrated into the reasoning framework developed for the first step.

To validate the developed method, a case study from the medical domain will be used, which puts into practice traceability management between requirements modeled with the RDAL language and system architectures modeled with the AADL language. The approach will be

implemented with the Eclipse Modeling Framework and released under the Eclipse platform integrating the Open Source AADL Tool Environment for editing AADL and RDAL models.

3.2 Betreuung von Masterarbeiten / Supervised Masterthesises abgeschlossen/finished 2016

[MA1] Christian Nicolai. Using exchangeable constraint solvers for invariant checking on attributed graph transformation systems. Master's thesis, Hasso-Plattner-Institut für Software-systemtechnik, Universität Potsdam, 2016.

3.3 Betreuung von Dissertationen / Supervised PhDs

3.3.1 Abgeschlossene Dissertationen / Finished PhDs

[D1] Sebastian Wätzoldt. *Modeling collaborations in adaptive systems of systems*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 2016.

3.3.2 Laufende Dissertationen / Running PhDs

Christian Adriano: Human-in-the-loop mechanism for Locating Software Faults

Thomas Brand: Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

Johannes Dyck: Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten

Sona Ghahremani: Utility-driven Architecture-based Self-adaptive Systems

Joachim Hänsel: Testing for Self-Adaptive Software Systems

Lucas Sakizoglou: Towards Graph-based Runtime Monitoring for Large Systems

Thomas Vogel: Eine modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme

4 Bearbeitete Forschungsthemen / Research Topics

4.1 Eine Modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme

Diese Arbeit verbindet die beiden Forschungsbereiche Model-Driven Engineering (MDE) und Selbst-adaptive Softwaresysteme, indem eine modellgetriebene Infrastruktur die Selbst-Adaption eines Systems unterstützen oder gar ermöglichen soll. Während der Fokus von MDE auf der Entwicklung und dem Deployment von Softwaresystemen liegt, können MDE Konzepte und Technologien auch für die Laufzeitverwaltung von Systemen hilfreich sein. Beispielsweise können Modelle verschiedene Sichten auf unterschiedlichen Abstraktionsebenen eines laufenden Systems bieten und damit eine reichhaltige semantische Grundlage für die Selbst-Adaption sein. Desweiteren soll der Einsatz von MDE Technologien für die Verwaltung und Evolution von Laufzeitmodellen untersucht werden und wie diese Laufzeitmodelle zu Modellen der Entwicklungsphase in Beziehung stehen können. Service-orientierte Systeme unterstützen aufgrund ihrer Modularität und losen Kopplung grundlegend die Adaption auf der Ebene der Architektur, aber ihre inherente Komplexität und Verteilung bringen Herausforderungen mit sich. Das Ziel dieser Arbeit sind Konzepte für eine generische modellgetriebene Infrastruktur, die die Laufzeitverwaltung und insbesondere die (Selbst-)Adaption von verteilten, service-orientierten Softwaresystemen ermöglicht. Diese Konzepte sollen durch einen Prototyp evaluiert werden.

Ansprechpartner: Thomas Vogel

4.2 Human-in-the-loop mechanism for Locating Software Faults

Context Much of the current software development work involves exploring diverse and large software data spaces, e.g., source code lines, software execution traces, and system configurations. These software data spaces are present when a software engineer works to locate a software fault among hundreds of lines of code, or to isolate performance bottlenecks in complex systems interactions, or even test an autonomous systems with hundreds of configurations. Note that, although distinct, these types of software development work share at least two characteristics with regards to execution and planning. Execution is widely supported by tools (e.g., debuggers and test runners) and automated methods (e.g., static and dynamic analysis, machine learning algorithms). Meanwhile, planning is dependent on the quality of human decision making, which involves ad hoc application of heuristics, e.g., as divide-and-conquer, encapsulation, and abstraction. Hence, software development is faced with partial automation of work that involves manipulating large volumes of software data.

Research Problem One consequence of the partial automation is that the outcomes are as fast and good as the skill of the software engineers involved. Besides being scarce, skill is not seamlessly transferable among engineers or even across domains for the same engineer. Therefore, there is an inherent uncertainty about the speed and quality of the exploratory software development work. Therefore, the research problem that we are tackling is to investigate the feasibility of combining automated tools and human decision making in a closed feedback loop. The goal is to improve speed and quality of software development that involves exploring diverse and large data spaces.

Insight - Software development as a service While deciding who and how to perform these types of software development work, teams are confronted with many options of engineering skill, tools, and tasks. Since these options are interdependent, their combination consists of a workflow. In order to investigate these different combinations interchangeably, we treated the tasks performed by either humans or tools as “exploratory software services”.

Best of humans and machines Combining humans and automated tools in a seamless workflow was done in many domains and has been shown to enable leveraging on the strengths of both. Automated mechanisms are good to analyze large amounts of data (source code) and to monitor multiple parts of a system to detect relevant events (via source code instrumentation). Humans are good at weighing on alternative root-causes (answer subjective questions and evaluate hypotheses) and creating explanations through counter-factual thinking (what if scenarios).

Approach In order to validate my approach we have been utilizing a large data set of answers to fault localization questions. These answers (2860) were collected from 666 Mechanical Turk workers who were qualified through a software programming test. Answers consist of questions about a possible relationship between a source code fragment and unit test failure as demonstrated in figure

Ansprechpartner: Christian Adriano

4.3 Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten

Invariant-Checking ist eine statische Analyse-Technik, mit der auf Basis der Verhaltensspezifikation eines Systems die Gültigkeit oder Ungültigkeit bestimmter Eigenschaften des Systems formal nachgewiesen werden kann. Typische Beispiele für derartige Eigenschaften sind Sicherheits- und Lebendigkeitseigenschaften, die für die Korrektheit, Sicherheit und konstante Ausführbarkeit eines Systems eine wichtige Rolle spielen. Insbesondere für sicherheitskritische oder auch für selbstadaptive Systeme sind solche Eigenschaften und deren formale Verifikation interessant.

Der im konkreten Fall verfolgte Ansatz des Invariant-Checking basiert auf Graphtransformationen zur Verhaltensspezifikation und Graphbedingungen zur Darstellung der gewünschten Eigenschaften. Dabei kann festgestellt werden, ob eine solche Eigenschaft eine induktive Invariante ist, also ob sie für einen Übergang des Systems von einem Zustand in den nächsten in jedem Fall bewahrt bleibt.

Das Forschungsthema beschäftigt sich mit der Erweiterung des Konzepts der induktiven Invarianten auf k-induktive Invarianten, wobei nicht lediglich einschrittige Zustandsübergänge betrachtet werden. Vielmehr kann durch die Untersuchung eines Zustandspfades der Länge k eine detailliertere Aussage über die Gültigkeit der zu beweisenden Eigenschaften getroffen werden. Beispielsweise könnte eine Eigenschaft als induktive Invariante zurückgewiesen werden, weil die Eigenschaft nach einem Zustandsübergang aus einem Zustand verletzt wird, der wiederum nur aus einem anderen verbotenen Zustand erreichbar ist. Durch die Untersuchung eines längeren Pfades wird die Zahl der Gegenbeispiele, die auf derartigen nicht korrekt erreichbaren Zuständen basieren, reduziert. Ein weiterer Punkt im Rahmen des Themas ist die Ausdrucksmächtigkeit des Ansatzes und die potentielle Erweiterung derselben.

Ansprechpartner: Johannes Dyck

4.4 Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

To which extend a software system and its underlying software products can maintain or extend their relevance for users depends significantly on how both evolve and get adapted to feedback and changing conditions.

Thus for manufacturers of software products the following tasks are crucial:

- Understand the customers' change requests and requirements.
- Generalize customer requests, prioritize and integrate them into existing software products. Additionally foster the maintainability and adaptability of the products.
- Offer and provide the resulting changes to the customers.

With our research we want to investigate how software manufactures and their ecosystem partners can make well-grounded software evolution decisions with less effort through runtime data. After conducting an explorative empirical study we decided to focus on how to obtain runtime data for feedback purposes through adaptive monitoring. Our motivation is to speed-up feedback cycles, make them more flexible with regard to changing and unforeseen data demands as well as more efficient concerning resource consumption.

Ansprechpartner: Thomas Brand

4.5 Utility-driven Architecture-based Self-adaptive Systems

Self-adaptive software provides the capability to observe changes of itself at runtime, reason about itself, and autonomously adapt itself. However, this additional capability is always limited concerning its power to reason on itself and there is always the trade off which costs come with the reasoning and adapting. On the one end of the spectrum of possible approaches are rule-based approaches which are often limited concerning their reasoning power but highly efficient and on the other end are utility-driven approaches that consider often costly algorithms to achieve a good optimization of the utility.

We peruse a hybrid adaptation policy in a modularized multi-concern self-adaptive system where each module is allowed to apply the type of adaptation strategy which fits better to its concern. The goal of the hybrid adaptation is to benefit from the strong points of each methodology and let them compensate for each others weaknesses. Regarding the challenges caused by modularization such as the order in which modules need to be executed and dependencies among modules, we intend to apply a policy in which we avoid the probable unwanted interferences. We also propose benefiting from a utility-driven policy to obtain the optimal ordering scenarios for executing modules, for that purpose we need to assign utility values to each module and define a utility function which assigns a real-valued scalar representing the desirability of system configuration identifying how good that specific state or configuration is as opposed to others. The proposed approach allows to define a utility function for architectural runtime models, to specify the possible improvements or repairs of the architecture by means of rules, and finally to achieve the incremental triggering of the rules according to their impact on the utility. We were able to show that the suggested utility-driven

rule-based adaptation works incrementally and that under certain assumptions it always results in the optimal improvement of the utility over time.

Ansprechpartner: Sona Ghahremani

4.6 Towards Graph-based Runtime Monitoring for Large Systems

Complex organizations nowadays require the constant interaction of people and technology. Monitoring correctness of such software systems is a challenging task in itself, primarily because: a) such systems often feature legacy technology which cannot be reconfigured or re-purposed but just observed b) exhaustive verification tools often prove impractical because of the “state-space explosion” problem and c) even if a) and b) were addressed, the interaction of software with humans would still pose a big threat to verification at design-time: an engineer can only make assumptions about how humans will interact with the software, and there is no guarantee that these assumptions will be valid during execution.

In the context of these large systems and their correctness, we think it is important to look into online runtime monitoring techniques. Runtime monitoring focuses on monitoring the correctness of a given (typical the current) execution. Specifically, our goal is to develop a runtime monitoring approach where the notion of typed, attributed graphs is used to represent a system state and to express properties on these states. The approach can express complex properties that pertain to the content and structure of monitored data, and the ordering and timing constraints of events.

Ansprechpartner: Lucas Sakizloglou

4.7 Testing for Self-Adaptive Software Systems

Self-adaptive software systems are equipped with feedback loops to adapt autonomously to changes of the software or environment. In established fields, such as embedded software, sophisticated approaches have been developed to systematically study feedback loops early during the development. In order to cover the particularities of feedback, techniques like one-way and in-the-loop simulation and testing have been included. However, related approaches for systematic testing of feedback loops in self-adaptive software system do not exist.

We propose a systematic testing approach based on architectural runtime models for self-adaptive software systems. The aim is to exploit architectural runtime models for testing early in the development phase, since they are usually available, even before the different activities of a feedback loop are realised or even designed. Furthermore we research testing of self-adaptive software systems at runtime in order to benefit from knowledge about the changed environment which is not available at design time.

Ansprechpartner: Joachim Hänsel

5 Drittmittelprojekte / Third-Party funded Projects

5.1 DFG – Graduiertenkolleg SOAMED

Gefördert: ???

Drittmittelgeber: DFG

Service-orientierte Architekturen zur Integration Software-gestützter Prozesse am Beispiel des Gesundheitswesens und der Medizintechnik.

**TODO:
Holger,
Lucas**

5.2 DFG – Korrekte Modelltransformationen (KorMoran) – Fortsetzungsprojekt

Gefördert: ab 08/2013

Drittmittelgeber: DFG

Bislang gibt es bis auf eigene Vorarbeiten keine Arbeiten, in denen Methoden für den formalen Nachweis der Korrektheit einer durch Modelltransformationen beschriebenen Transformation basierend auf Graphtransformationen vorgestellt werden. Ausgehend von den auf Graphtransformationssystemen basierenden Story Diagrammen und Triplegraphgrammatiken als Repräsentanten für operationale und relationale Modelltransformationsansätze wollen wir die Tatsache nutzen, dass Graphtransformationssysteme sich auch zur Spezifikation der Semantik von Modellen eignen, so dass wir das Problem der formalen Verifikation von Modelltransformationen mit einem einzigen formalen Modell angehen können.

Darauf aufbauend soll ein Ansatz für die systematische Entwicklung korrekter Modelltransformationen entwickelt und erprobt werden, der entsprechende Konzepte und Algorithmen für die formale Analyse und Verifikation der Modellsynchronisationen, Modelltransformationen und Modelltransformationsergebnisse enthält, die existierende Werkzeugunterstützung für Story Diagramme und Triplegraphgrammatiken soll um Werkzeuge für die formale Verifikation (automatisch und semi-automatisch) ergänzt werden, und es soll ein Vorgehen bzw. ein Prozess zur Verifikation aus Entwickler- und Benutzersicht ausgearbeitet werden. Anhand von zwei Fallstudien (aus dem Automotive-Bereich und dem Maschinenbau) soll die Praxistauglichkeit der entwickelten Methoden nachgewiesen werden.

In der Fortsetzungsphase des KorMoran-Projekts soll die Projektarbeit in zwei Richtungen vorangetrieben werden: Zum einen sollen offene theoretische Fragestellungen weiter bearbeitet werden. Zum anderen soll der Transfer der Projektergebnisse in die Praxis vorangetrieben werden. Noch offene theoretische Probleme betreffen vor allem die Definition einer adäquaten Verfeinerungsrelation für Strukturmodelle. Zu den für die Praxis wichtigen Fragestellungen gehört die Entwicklung eines Gesamtansatzes für die Absicherung der Korrektheit der Modelltransformationen, die Abbildung bzw. Herstellung von Bezügen zu anderen Modelltransformations-, Modellierungs- bzw. Programmiersprachen, die von praktischer Bedeutung sind, sowie die Entwicklung eines Vorgehensmodells für die Praxis. Außerdem ist auch in der Fortsetzungsphase geplant, sowohl die theoretischen als auch die praktischen Projektergebnisse anhand von Fallstudien zu evaluieren.

Ansprechpartner: Holger Giese, Leen Lambers, Johannes Dyck

5.3 Quantum

Gefördert: ab 01/2015

Drittmittelgeber: DFG

Ziel von QUANTUM ist die Entwicklung neuer quantitativer Modelle und quantitativer Analysetechniken für service-orientierte Echtzeitsysteme, welche die nötigen Kombinationen aus probabilistischem Verhalten, Echtzeitverhalten und Strukturdynamik bieten, die besondere Relevanz im Bereich von service-orientierten Echtzeitsystemen haben. Obwohl bereits limitierte Kombinationen aus probabilistischem Verhalten, Echtzeitverhalten und Strukturdynamik existieren, und auch substantielle Fortschritte bezüglich ihrer Analyse in den letzten Jahren gemacht wurden, fehlt noch immer eine komplette Kombination, welche alle geforderten Aspekte in einem Modell vereint.

Im Projekt ist deshalb geplant, die existierenden Modelle von zeitbehafteten Graphtransformationssystemen und probabilistischen Graphtransformationssystemen zu kombinieren und zu erweitern und passende Analysemöglichkeiten durch Integrieren von existierenden Werkzeugen bereitzustellen, welche die quantitative Analyse einer größeren Klasse von Systemen und ihrer Eigenschaften erlaubt als es durch die bisher existierenden Modelle möglich ist. Neben dem neuen formalen Modell, welches alle relevanten Aspekte abdeckt, wird eine probabilistische zeitbehaftete Spezifikationslogik, eine auf dem formalen Modell basierende, abstrakte QUANTUM-Modellierungssprache, welche durch Erweitern des SoaML UML-Profiles direkt die Beschreibung von service-orientierten Echtzeitsystemen ermöglicht, sowie eine verwandte visuelle Spezifikationsprache für QUANTUM-Modelle entwickelt, um die Modellierungskonzepte und Analysetechniken für ein breiteres Publikum nutzbar zu machen.

Ansprechpartner: Holger Giese, Maria Maximova

6 Forschungsk Kooperationen / Research Cooperations

6.1 Kooperationspartner aus der Wissenschaft / Partners from Research Institutions

Sabine Glesner (TU Berlin)
Verifikation von Code-Generierung und Modelltransformationen

Christian Krause (SAP Innovation Center)
Graphdatenbanken

Jürgen Dingel (Queen's University, Kanada)
Semantikbewahrung für Modelltransformationen

**TODO:
Check and
update**

6.2 Kooperationspartner aus der Wirtschaft / Partners from Industry

SAP AG
Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

7 Publikationen / Publications

7.1 Zeitschriftenartikel / Journals

- [A1] Levi L'ucio, Moussa Amrani, Juergen Dingel, Leen Lambers, Rick Salay, Gehan M. K. Selim, Eugene Syriani, and Manuel Wimmer. Model transformation intents and their properties. *Software and System Modeling*, 15(3):647–684, 2016.

7.2 Beiträge zu Büchern und Sammlungen / Books and Collections

- [S1] Thomas Beyhl and Holger Giese. The Design Thinking Methodology at Work: Capturing and Understanding the Interplay of Methods and Techniques. In Hasso Plattner, Christoph Meinel, and Larry Leifer, editors, *Design Thinking Research - Taking Breakthrough Innovation Home*, Understanding Innovation, pages 49–65. Springer International Publishing, 2016.
- [S2] Marin Litoiu, Mary Shaw, Gabriel Tamura, Norha M. Villegas, HausiA. Müller, Holger Giese, Eric Rutten, and Romain Rouvoy. What Can Control Theory Teach Us About Assurances in Self-Adaptive Software Systems? In Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese, editors, *Software Engineering for Self-Adaptive Systems III. Assurances*, pages 90–134. Springer International Publishing, 2017.

7.3 Begutachtete Konferenz- und Workshopartikel / Peer-Reviewed Conference and Workshop Papers

- [K1] Thomas Beyhl, Dominique Blouin, Holger Giese, and Leen Lambers. On the Operationalization of Graph Queries with Generalized Discrimination Networks. In Rachid Echahed and Mark Minas, editors, *Proceedings of the 9th International Conference on Graph Transformations*, pages 170–186. Springer, 2016.
- [K2] Thomas Beyhl and Holger Giese. Incremental View Maintenance for Deductive Graph Databases Using Generalized Discrimination Networks. In Alexander Heußner, Aleks Kissinger, and Anton Wijs, editors, *2nd Graphs as Models workshop (GaM 2016)*, number EPTCS 231, page 57–71, 2016.
- [K3] Dominique Blouin and Holger Giese. Combining Requirements, Use Case Maps and AADL Models for Safety-Critical Systems Design. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 00:266–274, 2016.
- [K4] Sona Ghahremani, Holger Giese, and Thomas Vogel. Towards Linking Adaptation Rules to the Utility Function for Dynamic Architectures. In *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO '16*, pages 142–143. IEEE, 2016.

- [K5] Christian Krause, Matthias Tichy, and Holger Giese. Implementing Graph Transformations in the Bulk Synchronous Parallel Model. In *Software Engineering Management 2015, Multikonferenz der GI-Fachbereiche Softwaretechnik (SWT) und Wirtschaftsinformatik (WI), FA WI-MAW, 17. März - 20. März 2015 Dresden, Germany*, pages 99–100, 2015.
- [K6] Marisa Navarro, Elvira Pino, Fernando Orejas, and Leen Lambers. A Logic of Graph Conditions Extended with Paths. In *7th Int. Workshop on Graph Computation Models*, 2016. Pre-proceedings.

7.4 Technische Berichte / Technical Reports

- [TR1] Thomas Beyhl, Dominique Blouin, Holger Giese, and Leen Lambers. On the Operationalization of Graph Queries with Generalized Discrimination Networks. Technical Report 106, Hasso Plattner Institute at the University of Potsdam, 2016.
- [TR2] Sona Ghahremani. Utility-Driven Modularized MAPE-K loop architectures for Self-adaptive systems. Technical report, Proceedings of the 9th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering, 2016.
- [TR3] Regina Hebig, Holger Giese, Kimon Batoulis, Philipp Langer, Armin Zamani Farahani, Gary Yao, and Mychajlo Wolowyk. Development of AUTOSAR standard documents at Carmeq GmbH: a case study. Technical Report 92, Hasso-Plattner Institut at University of Potsdam, 2016.

8 Vorträge / Talks

Sona Ghahremani

April 2016 *Utility-Driven Incremental Rule-based Adaptation for Architecture-Based Self-adaptive Systems*. HPI Symposium in Capetown University, Capetown South Africa, April 21, 2016.

8.1 Eingeladene Vorträge / Invited Talks

8.2 Vorträge auf Konferenzen und Workshops / Talks at Conferences and Workshops

Leen Lambers

July 2016 *On the Operationalization of Graph Queries with Generalized Discrimination Networks*. 9th International Conference on Graph Transformation (ICGT 2016), Vienna, Austria, July 6, 2016.

Thomas Vogel

April 2016 *Toward Self-Adaptive Software Employing Model Predictive Control*. NII Shonan Meeting on Controlled Adaptation of Self-adaptive Systems (CASaS), Shonan, Japan, April 25, 2016.

Sona Ghahremani

September 2016 *Towards Linking Adaptation Rules to the Utility Function for Dynamic Architectures*. 10th International Conference on Self-Adaptive and Self-Organizing Systems, Augsburg, Germany, September 15, 2016.

9 Web-Portale und -Services / Web-Portals and Services

9.1 Self-adaptive.org

Das Online-Angebot <http://www.self-adaptive.org> dient als Übersichtsseite für das jährliche Symposium *Software Engineering for Adaptive and Self-Managing Systems* (SEAMS) im Rahmen der *International Conference on Software Engineering* (ICSE). Auf der Webseite sind alle Call for Papers für aktuelle und vergangene SEAMS Symposien, eine umfassende themenspezifische Bibliographie, Informationen zu weiterführenden Veranstaltungen wie den Dagstuhl Seminaren 08031 und 10431 sowie eine Liste von Wissenschaftlern, die auf dem Gebiet forschen, zu finden.

9.2 MDELab.org

Mit dem Online-Angebot <http://www.mdelab.org> informieren wir über Forschungsarbeiten unseres Fachgebiets im Bereich des *Model-Driven Engineering* (MDE). Dabei liegt der Schwerpunkt auf Werkzeugen unter anderem für die modellgetriebene Softwareentwicklung, die an unserem Fachgebiet entwickelt werden und die zum Download bereitstehen.

9.3 CPSLab.org

Mit dem Online-Angebot <http://www.cpslab.org> informieren wir über Aktivitäten im Kontext unseres Labors im Bereich *Cyber-Physical-Systems*. Inhalte beziehen sich auf vergangene, aktuelle als auch geplante Forschungsarbeiten. Weiterhin werden ausgewählte Projekte, welche im Kontext der Lehre umgesetzt wurden, repräsentiert.

10 Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten / Memberships, Committee- and Reviewing-Activities

10.1 Mitgliedschaften / Memberships

Prof. Dr. Holger Giese

- Mitglied der Association for Computing Machinery (ACM)
- Mitglied der folgenden Special Interest Groups: SIGSOFT, SIGBED, SIGPLAN
- Mitglied der IEEE (Valued IEEE Member, Member since 1994)
- Mitglied der IEEE Computer Society
- Mitglied der folgenden Technical Councils: TCSE, TCDP, TCRTS, TFAAS
- Mitglied der IEEE Systems, Man, and Cybernetics Society
- Mitglied der Gesellschaft für Informatik e.V. (GI)
- Mitglied der folgenden Fachgebiete und Fachgruppen: ST, TAV, OOSE, ASE, PN, SPECS, FOMSESS
- Mitglied des Deutschen Hochschulverbandes (DHV)

Thomas Vogel

- Mitglied der Association for Computing Machinery (ACM)
- Mitglied der folgenden Special Interest Groups: SIGSOFT
- Mitglied der Gesellschaft für Informatik e.V. (GI)

Christian Adriano

- Mitglied der Association for Computing Machinery (ACM)
- Mitglied der IEEE (IEEE Member, Member since 2008)
- Mitglied der IEEE Computer Society

10.2 Mitarbeit in Programmkomitees / Activities in Committees

Prof. Dr. Holger Giese

- 8th International Workshop on Software Engineering for Resilient Systems (SERENE 2016)
Gothenburg, Sweden, September 5-6, 2016, [↗ website](#)
- 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2016)
Augsburg, Germany, September 12-16, 2016, [↗ website](#)
- Workshop on Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems (FORECAST 2016)
July 8, 2016, Vienna, Austria, [↗ website](#)

- The International Workshop on Models@run.time for Self-aware Computing Systems
July 18, 2016, Augsburg, Germany, [↗ website](#)
- 9th International Conference on Software Language Engineering (SLE'16)
October 31-November 1, 2016, Amsterdam, Netherlands, [↗ website](#)
- 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'16)
May 16, 2016, Austin, United States, [↗ website](#)
- 19th International Conference on Model Driven Engineering Languages and Systems (MODELS 2016)
October 2-7, 2016, Saint Malo, France, [↗ website](#)
- 13th International Conference on Autonomic Computing (ICAC2016)
Würzburg, Germany, 19-22.7.2016, [↗ website](#)

Dr. Leen Lambers

- 9th International Conference on Graph Transformation (ICGT 2016)
Vienna, Austria, July 5-6, 2016, [↗ website](#)
- 5th International Workshop on the Verification of Model Transformation (VOLT 2016)
Saint Malo, France, October 2, 2016, [↗ website](#)
- 2nd International Workshop on Model-Driven Engineering, Verification and Validation (ModeVVA 2016)
Saint Malo, France, October 3, 2016, [↗ website](#)
- 2nd International Workshop on Graphs as Models
Eindhoven, Netherlands, 02-03 April, 2016, [↗ website](#)

Thomas Vogel

- Workshop on Open Source Software for Model-Driven Engineering (OSS4MDE) at MoDELS
Saint-Malo, France, October 3, 2016, [↗ website](#)
- International Workshop on Models@run.time (MRT) at MoDELS
Saint-Malo, France, October 4, 2016, [↗ website](#)
- International Workshop on Models@run.time for Self-aware Computing Systems (MRT-ICAC) at ICAC
Würzburg, Germany, July 18, 2016, [↗ website](#)
- International Symposium on the Foundations of Software Engineering (FSE), Demo Paper Track
Seattle, WA, USA, November 13–18, 2016, [↗ website](#)
- European Conference on Software Architecture (ECSA), Doctoral Symposium
Copenhagen, Denmark, November 28–December 02, 2016, [↗ website](#)

10.3 Organisation von Tagungen und Workshops / Organisations of Conferences and Workshops

10.4 Gutachtertätigkeiten / Reviewing Activities

10.4.1 Forschungsprojekte / Research Projects

Prof. Dr. Holger Giese

- European Research Council (ERC)
- European Union Seventh Framework Programme (EU FP7)
- Deutsche Forschungsgemeinschaft (DFG)
- Die niederländische Organisation für wissenschaftliche Forschung (NWO)
- Swedisch Knowledge Foundation (KK-stiftelsen)
- Austrian Science Fund (FWF)
- Natural Sciences and Engineering Research Council (NSERC) Canada

10.4.2 Zeitschriften und Magazine / Journals

Prof. Dr. Holger Giese

- Science of Computer Programming (Zeitschrift)
- Transactions on Software Engineering and Methodology (Zeitschrift)
- Formal Aspects of Computing (Zeitschrift)
- IEEE Computer (Magazin)
- IEEE Robotics and Automation (Magazine)
- IEEE Software
- IEEE Transactions on Control Systems Technology
- IEEE Transactions on Industrial Informatics
- IEEE Transactions on Software Engineering
- Information and Software Technology
- Journal of Systems and Software (Zeitschrift)
- Journal of Visual Languages and Computing (Zeitschrift)
- Requirements Engineering (Zeitschrift)
- Simulation: Transactions of the Society for Modeling and Simulation International (Zeitschrift)
- Software Quality Journal (Zeitschrift)
- Software and Systems Modeling (Zeitschrift)
- Journal of Software Engineering for Robotics (JOSER)

- International Journal on Software Tools for Technology Transfer (STTT)
- International Journal of Aerospace Engineering (IJAE)
- Information Systems (IS) (Zeitschrift)
- Concurrency and Computation: Practice and Experience (Zeitschrift)
- Mechatronics (Zeitschrift)
- Algorithms (Zeitschrift)

Dr. Leen Lambers

- Journal of Systems and Software, Elsevier (Zeitschrift)

Thomas Vogel

- IEEE Transactions on Software Engineering (TSE) (Zeitschrift)
- ACM Transactions on Autonomous and Adaptive Systems (TAAS) (Zeitschrift)
- Software and Systems Modeling (Zeitschrift)
- Journal of Systems and Software (JSS) (Zeitschrift)
- IEEE Transactions on Cybernetics (Zeitschrift)
- Computing (Zeitschrift)
- LNCS Book on *Models@run.time* (Buch)