

Jahresbericht 2017

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Digital Engineering gGmbH

Campus Griebnitzsee
Universität Potsdam

Jahresbericht / Annual Report 2017

Fachgebiet Systemanalyse und Modellierung
Hasso-Plattner-Institut für Digital Engineering
Universität Potsdam



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

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

Inhaltsverzeichnis / Table of Contents

1	Personelle Zusammensetzung / Staff	1
2	Lehrveranstaltungen / Courses	3
2.1	Vorlesungen / Lectures	3
2.2	Übungen/Projekte / Exercises/Projects	3
2.3	Seminare / Seminars	3
3	Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations	4
3.1	Betreuung von Masterprojekten / Supervised Master's projects	4
3.1.1	Masterprojekte / Master's projects (abgeschlossen / finished in 2017)	4
3.2	Betreuung von Masterarbeiten / Supervised Master's theses	5
3.2.1	Abgeschlossene Masterarbeiten / Finished Master's theses	5
3.2.2	Laufende Masterarbeiten / Running Master's theses	5
3.3	Betreuung von Dissertationen / Supervised PhD theses	5
3.3.1	Abgeschlossene Dissertationen / Finished PhD theses	5
3.3.2	Laufende Dissertationen / Running PhD theses	5
4	Bearbeitete Forschungsthemen / Research Topics	7
4.1	Human-in-the-loop mechanism for Locating and Fixing Software Faults	7
4.2	Eine Modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme	7
4.3	Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten	8
4.4	Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems	8
4.5	Utility-driven Architecture-based Self-adaptive Systems	9
4.6	Towards Graph-based Runtime Monitoring for Large Systems	9
4.7	Testing for Self-Adaptive Software Systems	10
5	Drittmittelprojekte / Third-Party funded Projects	11
5.1	DFG – Graduiertenkolleg SOAMED	11
5.2	DFG – Quantitative Analyse von service-orientierten Echtzeitsystemen mit Struktur- dynamik (QUANTUM)	12
5.3	DFG – Korrekte Modelltransformationen (KorMoran III) – 2. Fortsetzungsprojekt	12
5.4	DFG – Modulares und inkrementelles globales Modell-Management (miGMM)	13
6	Forschungskooperationen / Research Cooperations	14
6.1	Projektpartner aus der Wissenschaft / Project Partners from Research Institutions	14
6.2	Projektpartner aus der Wirtschaft / Project Partners from Industry	14
6.3	Externe Kooperationspartner bei Publikationen / External Partners in Publications	14
7	Publikationen / Publications	16
7.1	Zeitschriftenartikel / Journal Articles	16
7.2	Beiträge zu Büchern und Sammlungen / Contributions to Books and Collections	16
7.3	Begutachtete Konferenz- und Workshopartikel / Peer-Reviewed Conference and Workshop Papers	18

7.4	Bücher und Tagungsbände / Books and Proceedings	19
7.5	Technische Berichte / Technical Reports	19
8	Vorträge / Talks	20
8.1	Eingeladene Vorträge / Invited Talks	20
8.2	Vorträge auf Konferenzen und Workshops / Talks at Conferences and Workshops	20
9	Auszeichnungen / Awards	22
10	Web-Portale und -Services / Web-Portals and Services	22
10.1	Self-adaptive.org	22
10.2	MDELab.org	22
10.3	CPSLab.org	22
11	Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten / Memberships, Committee and Reviewing Activities	23
11.1	Mitgliedschaften / Memberships	23
11.2	Mitarbeit in Programmkomitees / Activities in Program Committees	23
11.3	Organisationstätigkeiten / Organizational Activities	25
11.4	Gutachtertätigkeiten / Reviewing Activities	25
11.4.1	Forschungsprojekte / Research Projects	25
11.4.2	Zeitschriften und Magazine / Journals	25
11.4.3	Verschiedenes / Miscellaneous	26

1 Personelle Zusammensetzung / Staff



Leiter des Fachgebiets / Head

Prof. Dr. Holger Giese

Sekretariat / Secretary

Kerstin Miers

Senior Researcher

Dr. Leen Lambers

Postdocs

Dr. Soumyadip Bandyopadhyay (ab. 15.08.2017)

Dr. Dominique Blouin (extern)

Wissenschaftliche Mitarbeiter / Research Assistants

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 (bis 31.01.2017)

PhD-Stipendiaten / Scholarship Holders

Christian Adriano, M.Sc.
Dipl.-Wirtsch.Inf. Thomas Brand
Sona Ghahremani, M.Sc.

Studentische Hilfskräfte / Student Assistants

Matthias Barkowsky	Tim Cech	Jacob Freise
Paul Geppert	Konstantin Harmuth	Katrin Klein
Niklas Köhnecke	Maximilian König	Leon Masopust
Lukas Pirl	Marianna Thieffry	Ole Wegen
Christian Zöllner		

2 Lehrveranstaltungen / Courses

2.1 Vorlesungen / Lectures

Sommersemester / Summer term 2017

- Modellierung II

Wintersemester / Winter term 2017/2018

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

2.2 Übungen/Projekte / Exercises/Projects

Sommersemester / Summer term 2017

- Intelligent Autonomous Systems: Assurances and Ethics
- Modellierung II
- Test Automation for Graph Databases

Wintersemester / Winter term 2017/2018

- Modellierung I
- Predictions based on Graphs and Machine Learning
- Software Engineering for Self-Adaptive Systems
- Software Testen, Analysieren und Verifizieren

2.3 Seminare / Seminars

Sommersemester / Summer term 2017

- Intelligent Autonomous Systems: Assurances and Ethics
- Test Automation for Graph Databases

Wintersemester / Winter term 2017/2018

- Microservices and Domain-Driven Design
- Predictions Based on Graphs and Machine Learning

3 Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations

3.1 Betreuung von Masterprojekten / Supervised Master's projects

3.1.1 Masterprojekte / Master's projects (abgeschlossen / finished in 2017)

- *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, 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 fulfills them. Each requirement is linked to a set of associated architectural elements involved in implementing the requirement, 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 attributes 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. If 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 Master's theses

3.2.1 Abgeschlossene Masterarbeiten / Finished Master's theses

[MA1] Christian Zöllner. On the Formalization and Complexity of Triple Graph Grammars for Model Driven Engineering. Master's thesis, Hasso-Plattner-Institut, Universität Potsdam, 2017.

3.2.2 Laufende Masterarbeiten / Running Master's theses

Matthias Barkowsky: Optimizing Matching Multiple Graphs by the Tight Integration of Complex Parametrized Indices

Tim Naumann: Constraint-based Self-optimization of Deployments for Modular Systems

Melanie Schneider: Effiziente Simulation von probabilistischen Graphtransformationssystemen und deren Visualisierung

3.3 Betreuung von Dissertationen / Supervised PhD theses

3.3.1 Abgeschlossene Dissertationen / Finished PhD theses

[D1] Thomas Beyhl. *A Framework for Incremental View Graph Maintenance*. PhD thesis, Hasso-Plattner-Institut, Universität Potsdam, 2017.

3.3.2 Laufende Dissertationen / Running PhD theses

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

Open source software (OSS) projects democratized the access to high quality software and became a training ground for programmers to learn about new technology. However, to effectively participate in an OSS project, programmers still face many barriers such as having advanced programming skills and time to discover suitable tasks.

We investigated how to lower these barriers by enabling programmers to perform smaller and simpler tasks that can be later be composed into a larger task. We were particularly interested in the tasks of identifying the cause of a software failure and suggesting valid bug fixes. These tasks are a good starting point because they are self-contained and can be automatically generated and validated.

Nonetheless, as any new method to develop software, we have to deal with concerns of quality, speed, and cost. We coped with these concerns with a threefold approach: (1) we recruited and qualified programmers on a crowdsourcing platform, (2) we generated tasks automatically and incrementally based on template questions and on the outcome of previous tasks, and (3) we utilized the task outcomes train a series of decision trees to predict the correct location of the bug in the source code.

We evaluated our approach through a series of experiments with real software failures from popular OSS projects. Our preliminary results are promising in a sense that (1) different groups of programmers (subcrowds) were able to correctly find the cause of the software failures within a few lines of code, (2) speed and cost were reduced by iteratively deciding which tasks to generate next, and (3) as part of the tasks, programmers provided explanations that contributed positively to suggest valid bug fixes. However, we still have many open challenges such as selecting the source code to be part of each task, processing task outcomes in real-time, and automatically validating the bug fix suggestions.

Ansprechpartner / Contact: Christian Adriano

4.2 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 / Contact: Thomas Vogel

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 / Contact: 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 / Contact: Thomas Brand

4.5 Utility-driven Architecture-based Self-adaptive Systems

Architecture-based self-adaptive systems abstract the observed behavior of the running system into features of an architectural model, this makes it possible for the adaptation engine to reason about the changes that should be made to a system using variety of existing architectural analysis techniques. There are various ways how self-adaptation following the MAPE-K feedback loop and in particular the analyzing and planning phases of the loop can be realized. Rule-based approaches prescribe the adaptation to be executed if the system or environment satisfy certain conditions and result in scalable solutions, however, with often only satisfying adaptation decisions. In contrast, utility-driven approaches determine optimal adaptation decisions by using an often costly optimization step, which typically does not scale well for larger problems.

We propose a rule-based and utility-driven approach that achieves the beneficial properties of each of these directions such that the adaptation decisions are optimal while the computation remains scalable as an expensive optimization step can be avoided. The approach can be used for the architecture-based self-healing of large software systems. In our approach, we model the dynamic architecture of the self-adaptive system as a graph. Natural state of the system as well as the abstract syntax of the runtime models of the software are depicted via an annotated graph. We apply architectural utility functions in which any possible architectural configuration of the system is mapped to a scalar value.

We define the utility for large dynamic architectures of such systems based on patterns capturing issues the self-healing must address and we use pattern-based adaptation rules to resolve the issues. Defining the utility as well as the adaptation rules in a pattern-based manner allows us to compute the impact of each rule application on the overall utility and realize an incremental and efficient utility-driven self-adaptation. We target both self-healing and self-optimization in architectural manner. Achieving optimal adaptation decisions on-line within a reasonable time is an important challenge of self-adaptive software systems that is addressed.

Ansprechpartner / Contact: 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 (typically 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 / Contact: 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 / Contact: Joachim Hänsel

5 Drittmittelprojekte / Third-Party funded Projects

5.1 DFG – Graduiertenkolleg SOAMED

Gefördert / Funded: ab 04/2016

Drittmittelgeber / Funding organisation: DFG

Das DFG-Graduiertenkolleg SOAMED fokussiert sich auf service-orientierte Architekturen zur Integration softwaregestützter Prozesse am Beispiel des Gesundheitswesens und der Medizintechnik. Am Graduiertenkolleg beteiligte Universitäten und Institute sind die Humboldt-Universität zu Berlin, die Technische Universität Berlin, die Charité - Universitätsmedizin Berlin und das Hasso-Plattner-Institut.

Service-Orientierung ist ein viel versprechendes Architekturkonzept, um gekapselte Software-Komponenten (Services) effektiv und kosteneffizient zu komponieren und an neue Anforderungen anzupassen. Service-Orientierung wird bisher vorwiegend für kooperierende Geschäftsprozesse vorgeschlagen; zunehmend wird die Technologie aber auch zur Koppelung technischer (eingebetteter) Systeme und für die Gestaltung komplexer Informationssysteme eingesetzt. Service-Orientierung ist aus sehr pragmatischen Überlegungen und Fragestellungen heraus entwickelt worden. Weniger Aufmerksamkeit haben bisher grundlegende Betrachtungs- und Beschreibungsweisen sowie theoretische und konzeptionelle Problemstellungen gefunden. Auch sind softwaretechnische Methoden zur systematischen Konstruktion Service-orientierter Architekturen erst in Ansätzen verfügbar.

Die Informationstechnik ist eine Schlüsseltechnologie für die innovative Gestaltung unseres Gesundheitswesens und für die Nutzung der Medizintechnik. Im Vergleich zu anderen Bereichen sind allerdings die Prozesse vielfältiger und die Anforderungen an Zuverlässigkeit und Korrektheit höher. Prozesse in der Medizin sind zumeist lose gekoppelt; ihre Integration ist zugleich besonders schwierig und wichtig. Ihre derzeit praktizierte informationstechnische Unterstützung, zumeist historisch und unsystematisch gewachsen, kann mit einer systematischen, Service-orientierten, theoretischen und methodischen Fundierung der Herstellungsprozesse und Strukturen aller beteiligten softwaregesteuerten Komponenten substantiell verbessert werden.

In dieser Situation setzt das Graduiertenkolleg SOAMED mit der Idee an, das derzeit vorwiegend pragmatisch gehandhabte Service-orientierte Vorgehen in der Softwaretechnik sowohl theoretisch zu untermauern, als auch mit etablierten Software-Engineering-Verfahren zu kombinieren und so die Service-orientierte Systemkonstruktion konzeptionell, methodisch und werkzeuguunterstützt auszubauen.

Der Innovationsgehalt des Vorhabens ist umfangreich: Im Gesundheitswesen und in der Medizintechnik werden Strukturen, Prozesse und Kommunikationsprinzipien verwendet, die mit den im Graduiertenkolleg entwickelten Konzepten und Methoden signifikant besser als bisher konstruiert und beherrscht werden können. Die Beteiligung medizinischer Arbeitsgruppen gewährleistet die Praxisrelevanz der im Kolleg entwickelten Konzepte.

Ansprechpartner / Contact: Holger Giese, Lucas Sakizloglou. [↗ website](#)

5.2 DFG – Quantitative Analyse von service-orientierten Echtzeitsystemen mit Strukturodynamik (QUANTUM)

Gefördert / Funded: ab 01/2015

Drittmittelgeber / Funding organisation: 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 Strukturodynamik bieten, die besondere Relevanz im Bereich von service-orientierten Echtzeitsystemen haben. Obwohl bereits limitierte Kombinationen aus probabilistischem Verhalten, Echtzeitverhalten und Strukturodynamik 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 Spezifikationssprache für QUANTUM-Modelle entwickelt, um die Modellierungskonzepte und Analysetechniken für ein breiteres Publikum nutzbar zu machen.

Ansprechpartner / Contact: Holger Giese, Maria Maximova. [↗ website](#)

5.3 DFG – Korrekte Modelltransformationen (KorMoran III) – 2. Fortsetzungsprojekt

Gefördert / Funded: ab 11/2017

Drittmittelgeber / Funding organisation: DFG

Eingebettete Systeme sind heutzutage allgegenwärtig. Durch immer größer werdende Rechenleistungen und Vernetzung der Systeme sind diese in der Lage, immer komplexere Aufgaben zu erfüllen. Um diese Komplexität beherrschen zu können ist es notwendig, standardisierte und bewährte Methoden der Softwareentwicklung anzuwenden. Dazu zählt die modellgetriebene Entwicklung (MDE), die den Entwickler beim Design der abstrakten Anwendungsfälle bis zum konkreten, ausführbaren Code begleitet. Auch die abstrakten Modelle können sehr komplex werden. Eine Technik zur Reduzierung der Komplexität ist das Refactoring – Modelltransformationen, die äquivalentes Verhalten bei Ausführung der Modelle garantieren. In besonders sicherheitskritischen Bereichen, beispielsweise in der Automobil-, Luftfahrt- und Schienenverkehrsindustrie, spielt darüber hinaus formale Verifikation eine große Rolle. In diesen Industriezweigen wird zum MDE überwiegend MATLAB/Simulink eingesetzt.

Das DFG-Projekt KorMoran widmet sich daher dem Problem der Verifikation von Modelltransformationen, konkret dem formalen Beweis der Verhaltensäquivalenz von Quell- und Zielmodell. In den Vorgängerprojekten KorMoran I und II wurden sowohl Transformationen für zeit-diskrete und wert-diskrete Transitionsmodelle als auch für zeit-diskrete, zeit-kontinuierliche und wert-kontinuierliche Datenflussmodellen untersucht. In KorMoran III sollen nun zunächst die Untersuchungen zu zeit-diskreten und zeit-kontinuierlichen Datenflussmodellen fortgesetzt werden. Insbesondere ist eine Erweiterung der Verifikationsmethoden geplant, um hybride Systeme zu unterstützen – Modelle also, in denen sowohl zeit-diskrete als auch zeit-kontinuierliche Anteile gemischt vorkommen.

Bezüglich der Verifikation von Transformationen für Transitionsmodelle sollen Erweiterungen bestehender Techniken bezüglich der Ausdrucksmächtigkeit und Anwendbarkeit unternommen werden. Konkret sollen die existierenden Verifikationstechniken erweitert werden, um neben Bisimulation und Simulation auch schwache Bisimulation und Simulation zu unterstützen. Zusätzliche sollen auch Methoden für Transformationen für Transitionsmodelle mit wert-kontinuierlichen Signalen entwickelt werden.

KorMoran III ist ein Kooperationsprojekt zwischen der Technischen Universität Berlin und dem Hasso-Plattner-Institut, wobei erstere den Fokus auf den Bereich der Datenflussmodelle legt, während Modelltransformationen für Transitionsmodelle am Hasso-Plattner-Institut betrachtet werden. Gemeinsamer Teil des Projekts wird die Kombination entwickelter Techniken sein, um Anwendbarkeit für in Datenflussmodelle eingebettete Transitionsmodelle zu untersuchen und zu erreichen.

Ansprechpartner / Contact: Holger Giese, Leen Lambers, Johannes Dyck. [↗ website](#)

5.4 DFG – Modulares und inkrementelles globales Modell-Management (miGMM)

Genehmigt / Granted: 07/2017, noch nicht gestartet / not yet started

Drittmittelgeber / Funding organisation: DFG

Die Entwicklung komplexer Systeme mit Hilfe einer Vielzahl von Modellen benötigt ein globales Modell-Management, das sicherstellt, dass neben den Arbeiten auf einzelnen Modellen auch das Wechselspiel zwischen den Modellen geeignet verwaltet wird. Solch eine Verwaltung muss dabei die Integration der Modellierungssprachen, die Koordination der Aktivitäten auf Basis der Modelle sowie die Verwaltung der Modelle und all der Aktivitäten auf diesen abdecken. Es existiert zwar eine Reihe von Ansätzen, die Teile dieses Problem zu adressieren versuchen; ein fundiertes Verständnis der Bedürfnisse und Herausforderungen fehlt jedoch bisher. Darüber hinaus skalieren die meisten Lösungen nicht für die heutzutage durchaus vorkommenden sehr großen Modelle und sie unterstützen auch keine Modularität. Diese Einschränkung gilt sowohl für die Konstruktion als auch Ausführung der Modelle und der entsprechenden Aktivitäten. Im Projekt “modulares und inkrementelles Globales Modell-Management” (miGMM) wollen wir deswegen die Herausforderung des globales Modell-Management angehen, indem wir einen Ansatz für Mega-Modelle mit Integrationslinks, Integrationsichten, Nachverfolgbarkeitslinks, Modellkonsistenz und Modelloperationen entwickeln und dabei insbesondere die notwendigen Konzepte für eine inkrementelle und modulare Lösung erforschen.

Ansprechpartner / Contact: Holger Giese.

6 Forschungsk Kooperationen / Research Cooperations

6.1 Projektpartner aus der Wissenschaft / Project Partners from Research Institutions

DFG-Projekt KorMoran II/III

Jürgen Dingel (Queen's University)

Sabine Glesner (Technische Universität Berlin)

DFG-Graduiertenkolleg SOAMED

Oliver Blankenstein (Charité Berlin)

Johann-Christoph Freytag (Humboldt-Universität zu Berlin)

Sabine Glesner (Technische Universität Berlin)

Mehmet Gövercin (Deutsches Herzzentrum Berlin)

Annette Grüters-Kieslich (Charité Berlin)

Stefan Jähnichen (Technische Universität Berlin)

Odej Kao (Technische Universität Berlin)

Ulf Leser (Humboldt-Universität zu Berlin)

Thomas Meyer (Charité Berlin)

Uwe Nestmann (Technische Universität Berlin)

Wolfgang Reisig (Humboldt-Universität zu Berlin)

Björn Scheuermann (Humboldt-Universität zu Berlin)

Mathias Weske (Hasso-Plattner-Institut)

Scalable Model Management

Etienne Borde (Télécom ParisTech, Université Paris-Saclay)

Dalila Tamzalit (LS2N, Université de Nantes)

6.2 Projektpartner aus der Wirtschaft / Project Partners from Industry

DFG-Projekt QUANTUM / Graphdatenbanken

Christian Krause (SAP Innovation Center)

Runtime Data-driven Software Evolution in Enterprise Software Ecosystems

Patrick Fiegl (Vice President, S/4 HANA Co-innovation & Product Management, SAP)

Martin Günther (Head of SAP Usage Measurement Program, SAP)

Matthias A. Schmitt (Global Vice President, Best Practices Development for Lines of Business & Industries Solutions, SAP)

6.3 Externe Kooperationspartner bei Publikationen / External Partners in Publications

Prof. Dr. Holger Giese hat in 2017 mit folgenden externen Kooperationspartnern gemeinsame Publikationen herausgegeben oder veröffentlicht: Tarek Abdelzaher, Jesper Andersson, Luciano Baresi, Kirstie Bellman, Nelly Bencomo, Robert Birke, Jonas Bjurhede,

Etienne Borde, Yuriy Brun, Radu Calinescu, Javier Cámara, Lydia Y. Chen, Myra B. Cohen, Rogério de Lemos, Ada Diaconescu, Lukas Esterle, Antonio Filieri, Sylvain Frey, David Garlan, Kurt Geihs, Erol Gelenbe, Carlo Ghezzi, Alessandra Gorla, Sebastian Götz, Vincenzo Grassi, Lars Grunske, Regina Hebig, Henry Hoffmann, Hank Hoffmann, Paola Inveradi, Jean-Marc Jézéquel, Jeffrey O. Kephart, Samuel Kounev, Anne Koziol, Christian Krause, Christopher Landauer, Peter Lewis, Marin Litoiu, Martina Maggio, Sam Malek, Wilhelm Meding, Ole J. Mengshoel, Raffaella Mirandola, Marco Mori, Hausi A. Müller, Joost Noppen, Alessandro Vittorio Papadopoulos, Anders Robertsson, Romain Rouvoy, Cecilia M. F. Rubira, Eric Rutten, Hartmut Schmeck, Bradley Schmerl, Gehan Selim, Mary Shaw, Ola Soder, Simon Spinner, Miroslaw Staron, Giordano Tamburrelli, Gabriel Tamura, Dalila Tamzalit, Matthias Tichy, Norha M. Villegas, Danny Weyns, Katinka Wolter, Franco Zambonelli, Xiaoyun Zhu und Andrea Zisman.

Dr. Leen Lambers hat in 2017 mit folgenden externen Kooperationspartnern gemeinsame Publikationen herausgegeben oder veröffentlicht: Fernando Orejas, Daniel Strüber und Gabriele Taentzer.

7 Publikationen / Publications

7.1 Zeitschriftenartikel / Journal Articles

- [A1] Antonio Filieri, Martina Maggio, Konstantinos Angelopoulos, Nicolas D'Ippolito, Ilias Gerostathopoulos, Andreas Berndt Hempel, Henry Hoffmann, Pooyan Jamshidi, Evangelia Kalyvianaki, Cristian Klein, Filip Krikava, Sasa Misailovic, Alessandro Vittorio Papadopoulos, Suprio Ray, Amir Molzam Sharifloo, Stepan Shevtsov, Mateusz Ujma, and Thomas Vogel. Control Strategies for Self-Adaptive Software Systems. *ACM Trans. Auton. Adapt. Syst.*, 11(4):24:1–24:31, 2017.
- [A2] Regina Hebig and Holger Giese. On the Complex Nature of MDE Evolution and its Impact on Changeability. *Software & Systems Modeling*, 16(2):333–356, 2017.
- [A3] Mirosław Staron, Wilhelm Meding, Matthias Tichy, Jonas Bjurhede, Holger Giese, and Ola Soder. Industrial experiences from evolving measurement systems into self-healing systems for improved availability. *Software: Practice and Experience*, 48(3):719–739, 2017.

7.2 Beiträge zu Büchern und Sammlungen / Contributions to Books and Collections

- [S1] Robert Birke, Javier Camara, Lydia Y. Chen, Lukas Esterle, Kurt Geihs, Erol Gelenbe, Holger Giese, Anders Robertsson, and Xiaoyun Zhu. Self-Aware Computing Systems: Open Challenges and Future Research Directions. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 26, pages 709–722. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S2] Rogério de Lemos, David Garlan, Carlo Ghezzi, Holger Giese, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Danny Weyns, Luciano Baresi, Nelly Bencomo, Yuriy Brun, Javier Camara, Radu Calinescu, Myra B. Cohen, Alessandra Gorla, Vincenzo Grassi, Lars Grunskke, Paola Inverardi, Jean-Marc Jézéquel, Sam Malek, Raffaella Mirandola, Marco Mori, Hausi A. Müller, Romain Rouvoy, Cecilia M. F. Rubira, Eric Rutten, Mary Shaw, Giordano Tamburrelli, Gabriel Tamura, Norha M. Villegas, Thomas Vogel, and Franco Zambonelli. Software Engineering for Self-Adaptive Systems: Research Challenges in the Provision of Assurances. In Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese, editors, *Software Engineering for Self-Adaptive Systems III. Assurances*, pages 3–30. Springer International Publishing, 2017.
- [S3] Ada Diaconescu, Kirstie Bellman, Lukas Esterle, Holger Giese, Sebastian Götz, Peter Lewis, and Andrea Zisman. Architectures for Collective Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 7, pages 191–235. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.

- [S4] Holger Giese, Thomas Vogel, Ada Diaconescu, Sebastian Götz, and Kirstie Bellman. Generic Architectures for Individual Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 6, pages 149–189. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S5] Holger Giese, Thomas Vogel, Ada Diaconescu, Sebastian Götz, and Samuel Kounev. Architectural Concepts for Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 5, pages 109–147. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S6] Holger Giese, Thomas Vogel, Ada Diaconescu, Sebastian Götz, Nelly Bencomo, Kurt Geihs, Samuel Kounev, and Kirstie Bellman. State of the Art in Architectures for Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 8, pages 237–275. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S7] Jeffrey O. Kephart, Ada Diaconescu, Holger Giese, Anders Robertsson, Tarek Abdelzaher, Peter Lewis, Antonio Filieri, Lukas Esterle, and Sylvain Frey. Self-Adaptation in Collective Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 13, pages 401–435. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S8] Jeffrey O. Kephart, Martina Maggio, Ada Diaconescu, Holger Giese, Henry Hoffmann, Samuel Kounev, Anne Koziol, Peter Lewis, Anders Robertsson, and Simon Spinner. Reference Scenarios for Self-Aware Computing. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 4, pages 87–106. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S9] Samuel Kounev, Jeffrey O. Kephart, Peter Lewis, Ada Diaconescu, Kurt Geihs, Holger Giese, Paola Inverardi, Andrea Zisman, Kirstie Bellman, Lukas Esterle, Nelly Bencomo, Hartmut Schmeck, Hank Hoffmann, Sebastian Götz, and Javier Camara. Self-Aware Computing Systems: Related Concepts and Research Areas. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, pages 17–49. Springer-Verlag, Berlin/Heidelberg, Germany, 2017.
- [S10] Samuel Kounev, Jeffrey O. Kephart, Peter Lewis, Ada Diaconescu, Kurt Geihs, Holger Giese, Paola Inverardi, Andrea Zisman, Kirstie Bellman, Lukas Esterle, Nelly Bencomo, Hartmut Schmeck, Hank Hoffmann, Sebastian Götz, and Javier Camara. The Notion of Self-Aware Computing. In Samuel Kounev, Jeffrey O. Kephart, A. Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, pages 3–16. Springer-Verlag, Berlin/Heidelberg, Germany, 2017.
- [S11] Peter Lewis, Kirstie Bellman, Christopher Landauer, Javier Camara, Holger Giese, Nelly Bencomo, Ada Diaconescu, and Lukas Esterle. Towards a Framework for the Levels and Aspects of Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, pages 51–85. Springer-Verlag, Berlin/Heidelberg, Germany, 2017.

- [S12] Martina Maggio, Tarek Abdelzaher, Lukas Esterle, Holger Giese, Jeffrey O. Kephart, Ole J. Mengshoel, Alessandro Vittorio Papadopoulos, Anders Robertsson, and Katinka Wolter. Self-Adaptation for Individual Self-Aware Computing Systems. In Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu, editors, *Self-Aware Computing Systems*, chapter 12, pages 375–399. Springer International Publishing, Berlin/Heidelberg, Germany, 2017.
- [S13] Bradley Schmerl, Jesper Andersson, Thomas Vogel, Myra B. Cohen, Cecilia M. F. Rubira, Yuriy Brun, Alessandra Gorla, Franco Zambonelli, and Luciano Baresi. Challenges in Composing and Decomposing Assurances for Self-Adaptive Systems. In Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese, editors, *Software Engineering for Self-Adaptive Systems III. Assurances*, pages 64–89. Springer International Publishing, 2017.

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

- [K1] Soumyadip Bandyopadhyay, Santonu Sarkar, Dipankar Sarkar, and Chittaranjan A. Mandal. SamaTulyata: An Efficient Path Based Equivalence Checking Tool. In Deepak D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis - 15th International Symposium, ATVA 2017, Pune, India, October 3-6, 2017, Proceedings*, volume 10482 of *LNCS*, pages 109–116. Springer, 2017.
- [K2] Dominique Blouin, Matthias Barkowsky, Melanie Schneider, Holger Giese, Johannes Dyck, Etienne Borde, Dalila Tamzalit, and Joost Noppen. A Semi-Automated Approach for the Co-Refinement of Requirements and Architecture Models. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 36–45, September 2017.
- [K3] Kristopher Born, Leen Lambers, Daniel Strüber, and Gabriele Taentzer. Granularity of Conflicts and Dependencies in Graph Transformation Systems. In Juan de Lara and Detlef Plump, editors, *Graph Transformation*, volume 10373 of *LNCS*, pages 125–141, Cham, 2017. Springer.
- [K4] Johannes Dyck and Holger Giese. K-Inductive Invariant Checking for Graph Transformation Systems. In Juan de Lara and Detlef Plump, editors, *Graph Transformation*, volume 10373 of *LNCS*, pages 142–158, Cham, 2017. Springer.
- [K5] Sona Ghahremani, Holger Giese, and Thomas Vogel. Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures. In *2017 IEEE International Conference on Autonomic Computing (ICAC)*, pages 59–68, July 2017.
- [K6] Joachim Hänsel and Holger Giese. Towards Collective Online and Offline Testing for Dynamic Software Product Lines. In *Proceedings of the 2nd International Workshop on Variability and Complexity in Software Design*. ACM, 2017.
- [K7] Maria Maximova, Holger Giese, and Christian Krause. Probabilistic Timed Graph Transformation Systems. In Juan de Lara and Detlef Plump, editors, *Graph Transformation*, volume 10373 of *LNCS*, pages 159–175, Cham, 2017. Springer.

- [K8] Sven Schneider, Leen Lambers, and Fernando Orejas. Symbolic Model Generation for Graph Properties. In Marieke Huisman and Julia Rubin, editors, *Proc. Fundamental Approaches to Software Engineering (FASE'17)*, Lecture Notes in Computer Science, pages 226–243, Uppsala, Sweden, 2017. Springer.

7.4 Bücher und Tagungsbände / Books and Proceedings

- [B1] Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese, editors. *Software Engineering for Self-Adaptive Systems III. Assurances*. Springer International Publishing, 2017.

7.5 Technische Berichte / Technical Reports

- [TR1] Johannes Dyck and Holger Giese. K-Inductive Invariant Checking for Graph Transformation Systems. Technical Report 119, Hasso Plattner Institute, University of Potsdam, 2017.
- [TR2] Johannes Dyck, Holger Giese, and Leen Lambers. Automatic Verification of Behavior Preservation at the Transformation Level for Relational Model Transformation. Technical Report 112, Hasso Plattner Institute, University of Potsdam, 2017.
- [TR3] Maria Maximova, Holger Giese, and Christian Krause. Probabilistic Timed Graph Transformation Systems. Technical Report 118, Hasso-Plattner Institute at the University of Potsdam, 2017.
- [TR4] Sven Schneider, Leen Lambers, and Fernando Orejas. Symbolic Model Generation for Graph Properties. Technical Report 115, Hasso Plattner Institute at the University of Potsdam, Potsdam, Germany, 2017.

8 Vorträge / Talks

8.1 Eingeladene Vorträge / Invited Talks

Prof. Dr. Holger Giese

May 2017 *A Multi-paradigm Approach Supporting the Modular Execution of Reconfigurable Hybrid Systems*. International Workshop “Towards Holistic Computational Engineering”, Darmstadt, Germany, May 30, 2017.

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

Dr. Leen Lambers

July 2017 *Granularity of Conflicts and Dependencies in Graph Transformation Systems*. 10th International Conference on Graph Transformation (ICGT 2017), Marburg, Germany, July 19, 2017.

Dr. Soumyadip Bandyopadhyay

October 2017 *SamaTulyata: An Efficient Path Based Equivalence Checking Tool*. 15th International Symposium on Automated Technology for Verification and Analysis (ATVA 2017), Pune, India, October 4, 2017.

Christian Adriano

October 2017 *Humans in the Loop of Exploratory Software Services*. Research School Fall Retreat, Neuruppin, Germany, October 20, 2017.

December 2017 *Locating Software Faults by means of Humans in the Loop Mechanisms*. HPI Symposium at SAP, Walldorf, Germany, December 12, 2017.

Thomas Brand

October 2017 *Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems*. Research School Fall Retreat, Neuruppin, Germany, October 21, 2017.

Johannes Dyck

July 2017 *k-Inductive Invariant Checking for Graph Transformation Systems*. 10th International Conference on Graph Transformation (ICGT 2017), Marburg, Germany, July 19, 2017.

Sona Ghahremani

- June 2017 *Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures*. International Summer School on Software Engineering, University of Salerno, Italy, June 21, 2017.
- July 2017 *Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures*. 14th International Conference on Autonomic Computing (ICAC 2017), Columbus, USA, July 18, 2017.
- October 2017 *Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures*. HPI Symposium at SAP Next-Gen, New York, USA, October 9, 2017.
- October 2017 *Model-Driven Self-Adaptation for Large Dynamic Architectures*. Research School Fall Retreat, Neuruppin, Germany, October 21, 2017.
- November 2017 *Impact of Failure Profiles on the Performance of Self-healing Approaches*. Research School Doctoral Seminar, Potsdam, Germany, November 8, 2017.

Joachim Hänsel

- May 2017 *Towards Collective Online and Offline Testing for Dynamic Software Product Line Systems*. 2nd Workshop on Variability and Complexity in Software Design (VACE 2017), Buenos Aires, Argentina, May 27, 2017.

Maria Maximova

- July 2017 *Probabilistic Timed Graph Transformation Systems*. 10th International Conference on Graph Transformation (ICGT 2017), Marburg, Germany, July 19, 2017.

Sven Schneider

- April 2017 *Symbolic Model Generation for Graph Properties*. 20th International Conference on Fundamental Approaches to Software Engineering (FASE 2017), Uppsala, Sweden, April 27, 2017.

9 Auszeichnungen / Awards

Karsten Schwan Best Paper Award, ICAC 2017

Sona Ghahremani, Holger Giese, and Thomas Vogel.

Efficient Utility-Driven Self-Healing Employing Adaptation Rules for Large Dynamic Architectures.

EATCS Best Paper Award, ICGT 2017

Kristopher Born, Leen Lambers, Daniel Strüber, and Gabriele Taentzer.

Granularity of Conflicts and Dependencies in Graph Transformation Systems.

EASST Award Best Software Science Paper, ICGT 2017

Maria Maximova, Holger Giese, and Christian Krause.

Probabilistic Timed Graph Transformation Systems.

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

10.1 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.

10.2 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.

10.3 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.

11 Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten / Memberships, Committee and Reviewing Activities

11.1 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)

Christian Adriano

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

Thomas Vogel

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

11.2 Mitarbeit in Programmkomitees / Activities in Program Committees

Prof. Dr. Holger Giese

- 1st International Conference on Robotic Computing (IRC 2017)
Taichung, Taiwan, April 10-12, 2017, [↗ website](#)
- 3rd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SE-sCPS 2017)
Bueons Aires, Argentina, May 21, 2017, [↗ website](#)
- 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2017)
Bueons Aires, Argentina, May 22-23, 2017, [↗ website](#)

- International Workshop on Modeling and Software Engineering in Business and Industry (MoSEBIn 2017)
Zaragoza, Spain, June 26-27, 2017, [↗ website](#)
- 2nd International Workshop on Models@run.time for Self-Aware Computing Systems (MRT 2017)
Columbus, USA, July 17, 2017, [↗ website](#)
- 14th International Conference on Autonomic Computing (ICAC 2017)
Columbus, USA, July 17-21, 2017, [↗ website](#)
- 10th International Conference on Model Transformation (ICMT 2017)
Marburg, Germany, July 17-18, 2017, [↗ website](#)
- 10th International Conference on Graph Transformation (ICGT 2017)
Marburg, Germany, July 18-19, 2017, [↗ website](#)
- Workshop on Self-Aware Computing (SeAC 2017)
Columbus, USA, July 17, 2017, [↗ website](#)
- Software Engineering for Autonomous and Intelligent Systems (SE4AIS 2017)
Porto, Portugal, September 5-8, 2017, [↗ website](#)
- 1st International Workshop on Software Engineering for Internet of Things (SoftIoT 2017)
Canterbury, UK, September 11, 2017, [↗ website](#)
- 18th EPIA Conference on Artificial Intelligence (EPIA 2017)
Porto, Portugal, September 15-18, 2017, [↗ website](#)
- 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2017)
Tuscon, USA, September 18-22, 2017, [↗ website](#)
- International Workshop on Model-Driven Engineering Tools (MDETools 2017)
Austin, USA, September 19, 2017, [↗ website](#)
- 20th International Conference on Intelligent Transportation Systems (ITSC 2017)
Yokohama, Japan, October 16-19, 2017, [↗ website](#)
- 2nd Workshop on Formal and Model-Driven Techniques for Developing Trustworthy Systems (FM&MDD 2017)
Xi'an, China, November 13-17, 2017, [↗ website](#)

Dr. Leen Lambers

- Graphs as Models (GaM 2017)
Uppsala, Sweden, April 23, 2017, [↗ website](#)
- 10th International Conference on Graph Transformation (ICGT 2017)
Marburg, Germany, July 18-19, 2017, [↗ website](#)
- 14th Workshop on Model-Driven Engineering, Verification and Validation (ModeVVA 2017)
Austin, USA, September 19, 2017, [↗ website](#)

11.3 Organisationstätigkeiten / Organizational Activities

Prof. Dr. Holger Giese

- Chair of the Selection Committee for the next Editor in Chief of the ACM TAAS Journal
[↗ website](#)
- Management Committee Member for Germany at the ICT COST ACTION IC1404 “Multi-paradigm Modelling for Cyber-physical Systems” (MPM4CPS)
[↗ website](#)

Dr. Leen Lambers

- Publication Chair of STAF 2017
Marburg, Germany, July 17-21, 2017, [↗ website](#)

11.4 Gutachtertätigkeiten / Reviewing Activities

11.4.1 Forschungsprojekte / Research Projects

Prof. Dr. Holger Giese

- Deutscher Akademischer Austauschdienst (DAAD)
- Deutsche Forschungsgemeinschaft (DFG)
- Österreichische Forschungsförderungsgesellschaft (FFG)
- European Union Seventh Framework Programme (EU FP7)
- French Institute for Research in Computer Science and Automation (INRIA)
- Swedisch Knowledge Foundation (KK-stiftelsen)

11.4.2 Zeitschriften und Magazine / Journals

Prof. Dr. Holger Giese

- ACM Transactions on Autonomous and Adaptive Systems (TAAS)
- IEEE Transactions on Software Engineering (TSE)
- Information and Software Technology (IST)
- Journal of Engineering Education (JEE)
- Sensors
- Software and Systems Modeling (SoSyM)
- Theoretical Computer Science

11.4.3 Verschiedenes / Miscellaneous

Prof. Dr. Holger Giese

- Berufungskommission W3-Professur “Modellbasierte Entwicklung”, Universität Siegen