

Jahresbericht 2018

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Digital Engineering gGmbH

Campus Griebnitzsee
Universität Potsdam

Jahresbericht / Annual Report 2018

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	5
3.1	Betreuung von Masterarbeiten / Supervised Master's theses	5
3.1.1	Abgeschlossene Masterarbeiten / Finished Master's theses	5
3.1.2	Laufende Masterarbeiten / Running Master's theses	5
3.2	Betreuung von Dissertationen / Supervised PhD theses	5
3.2.1	Abgeschlossene Dissertationen / Finished PhD theses	5
3.2.2	Laufende Dissertationen / Running PhD theses	5
4	Bearbeitete Forschungsthemen / Research Topics	7
4.1	Human-in-the-loop Mechanism for Locating and Fixing Software Failures	7
4.2	Modular and Incremental Global Model Management	7
4.3	Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems	8
4.4	Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten	8
4.5	Utility-driven Architecture-based Self-adaptive Systems	9
4.6	Testing for Self-Adaptive Software Systems	9
4.7	Efficient Graph-based Runtime Monitoring for Large Systems	10
4.8	Run-time Verification and Validation for Self-adaptive System	10
5	Drittmittelprojekte / Third-Party funded Projects	12
5.1	DFG – Graduiertenkolleg SOAMED	12
5.2	DFG – Quantitative Analyse von service-orientierten Echtzeitsystemen mit Struktur- dynamik (QUANTUM)	13
5.3	DFG – Korrekte Modelltransformationen (KorMoran III) – 2. Fortsetzungsprojekt	13
5.4	DFG – Modulares und inkrementelles globales Modell-Management (miGMM)	14
6	Forschungskooperationen / Research Cooperations	16
6.1	Projektpartner aus der Wissenschaft / Project Partners from Research Institutions	16
6.2	Projektpartner aus der Wirtschaft / Project Partners from Industry	16
6.3	Externe Kooperationspartner bei Publikationen / External Partners in Publications	16
7	Publikationen / Publications	18
7.1	Zeitschriftenartikel / Journal Articles	18
7.2	Beiträge zu Büchern und Sammlungen / Contributions to Books and Collections	18
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	Web-Portale und -Services / Web-Portals and Services	22
9.1	Self-adaptive.org	22
9.2	MDELab.org	22
9.3	CPSLab.org	22
10	Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten / Memberships, Committee and Reviewing Activities	23
10.1	Mitgliedschaften / Memberships	23
10.2	Mitarbeit in Programmkomitees / Activities in Program Committees	23
10.3	Organisationstätigkeiten / Organizational Activities	26
10.4	Gutachtertätigkeiten / Reviewing Activities	26
10.4.1	Forschungsprojekte / Research Projects	26
10.4.2	Zeitschriften und Magazine / Journals	26
10.4.3	Verschiedenes / Miscellaneous	27

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 (bis 10.2018)

Wissenschaftliche Mitarbeiter / Research Assistants

Matthias Barkowsky, M.Sc.

Johannes Dyck, M.Sc. (bis 30.04.2018)

Dipl.-Inform. Joachim Hänsel

Dipl.-Inform. Maria Maximova

Lucas Sakizoglou, M.Sc. (ab 01.04.2018)

Dipl.-Inform. Sven Schneider
Christian Zöllner, M.Sc.

PhD-Stipendiaten / Scholarship Holders

Christian Adriano, M.Sc.
Dipl.-Wirtsch.Inf. Thomas Brand
Johannes Dyck, M.Sc. (ab 01.05.2018)
Sona Ghahremani, M.Sc.
He Xu, M.Sc.
Lucas Sakizoglou, M.Sc. (bis 31.03.2018)

Extern

Dr. Dominique Blouin
Dr. Soumyadip Bandyopadhyay (seit 11.2018)

Studentische Hilfskräfte / Student Assistants

Christoph Anders
Björn Daase
Ben Hurdelhey
Maximilian König
Paul Methfessel
Ole Wegen

Hendrik Bomhardt
Jacob Freise
Katrin Klein
Jonas Kordt
Nils Strassenburg

Tim Cech
Konstantin Harmuth
Niklas Köhnecke
Leon Masopust
Marianna Thieffry

2 Lehrveranstaltungen / Courses

2.1 Vorlesungen / Lectures

Sommersemester / Summer term 2018

- Modellierung II

Wintersemester / Winter term 2018/2019

- Fundamentals of Programming
- Safety-Critical Systems: From Predictable Systems to Autonomous AI
- Software Testen, Analysieren und Verifizieren

2.2 Übungen/Projekte / Exercises/Projects

Sommersemester / Summer term 2018

- Automated Verification Tools for Embedded Software
- Engineering Smart Systems and Service
- Modellierung II
- Test Automation for Graph Databases

Wintersemester / Winter term 2018/2019

- Fundamentals of Programming
- Runtime Verification: Online Testing Powered by Formal Methods
- Safety-Critical Systems: From Predictable Systems to Autonomous AI
- Software Testen, Analysieren und Verifizieren

2.3 Seminare / Seminars

Sommersemester / Summer term 2018

- Automated Verification Tools for Embedded Software
- Engineering Smart Systems and Service
- Runtime Monitoring
- Test Automation for Graph Databases

Wintersemester / Winter term 2018/2019

- Advanced Planning for Self-Aware Systems
- Runtime Verification: Online Testing Powered by Formal Methods

3 Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations

3.1 Betreuung von Masterarbeiten / Supervised Master's theses

3.1.1 Abgeschlossene Masterarbeiten / Finished Master's theses

- [MA1] Matthias Barkowsky. Tight Integration of Indices into Graph Query Execution. Master's thesis, Hasso-Plattner-Institut für Digital Engineering, Universität Potsdam, 2018.
- [MA2] Tim Naumann. Constraint-based Self-optimisation of Deployments for modular Systems. Master's thesis, Hasso-Plattner-Institut für Digital Engineering, Universität Potsdam, 2018.
- [MA3] Melanie Schneider. Effiziente Simulation von probabilistischen Graphtransformationssystemen und deren Visualisierung. Master's thesis, Hasso-Plattner-Institut für Digital Engineering, Universität Potsdam, 2018.

3.1.2 Laufende Masterarbeiten / Running Master's theses

3.2 Betreuung von Dissertationen / Supervised PhD theses

3.2.1 Abgeschlossene Dissertationen / Finished PhD theses

- [D1] Thomas Beyhl. *A Framework for Incremental View Graph Maintenance*. PhD thesis, Hasso-Plattner-Institut für Digital Engineering, Universität Potsdam, 2018.
- [D2] Thomas Vogel. *Model-Driven Engineering of Self-Adaptive Software*. PhD thesis, Hasso-Plattner-Institut für Digital Engineering, Universität Potsdam, 2018.

3.2.2 Laufende Dissertationen / Running PhD theses

Christian Adriano: Human-in-the-loop Mechanism for Locating and Fixing Software Failures

Matthias Barkowsky: Modular and Incremental Global Model Management

Thomas Brand: Generic Adaptive Monitoring with Architectural Runtime Models

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 Sakizloglou: Efficient Graph-based Runtime Monitoring for Large Systems

He Xu: Run-time Verification and Validation for Self-adaptive System

Christian Zöllner: Modellierung und Simulation von Kollaborationen

4 Bearbeitete Forschungsthemen / Research Topics

4.1 Human-in-the-loop Mechanism for Locating and Fixing Software Failures

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 composed into a larger outcome. 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.

Nonetheless, as any new method to develop software, we have to deal with concerns of quality, speed, and cost, which we are investigating with a threefold approach: (1) we recruited and qualified programmers on a crowdsourcing platform, (2) we generated tasks automatically and incrementally based on a set of template questions and on the outcome of previous tasks, and (3) we utilized the task outcomes to train a series of models to predict the correct location of the bug. 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 identify the cause of the software failures within a few lines of code, (2) the 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 bug fixes.

These results opened a new research problem: how to select a minimal set of tasks that maximize bug finding precision?. This is a difficult problem because it requires us to decide at any given moment which program statements to inspect by whom and how many programmers. Our approach has been to partition this problem in three sub-problems: a causal model, an aggregation model, and a task sequencing model. The causal model explains the accuracy of task outcomes based on the programmers' coding skill and the tasks attributes (duration, perceived difficulty). The aggregation model consists of mechanisms (majority and cardinal voting) that summarize the competing opinions about the bug location. The task sequencing model combines the Bayes update procedure (by learning from previous tasks) with the expected utility of each available task (extracted from the causal model). This way we iteratively update the knowledge about which minimal set of new tasks, if executed, would maximize the chances of precisely locating a given software bug.

Ansprechpartner / Contact: Christian Adriano

4.2 Modular and Incremental Global Model Management

The development of complex software systems involves the creation and maintenance of a multitude of models describing various aspects such as the architecture, behavior and requirements of the software. In model-driven development, models are assigned an important role in the development process and are subject to both manual and automated activities. Since these models may cover overlapping parts of the system under development, the execution of such activities has

to be coordinated properly by global model management in order to avoid inconsistencies in the system's description.

Because of the heterogeneity and growing size of the involved models, global model management poses several challenges. We want to address these issues by employing the concept of megamodels to document and execute the interplay between models. In particular, we are studying the concepts required to achieve a solution which can cope with incremental changes to existing models on the one hand and allows a modular introduction of additional models and activities on the other hand. To that end, we are working on extending the triple-graph-grammar approach for model synchronization by an efficient change propagation between models while keeping track of their version history in a compact manner. This is supported by our work on optimizing the execution of graph queries over large models via a combination of existing static and dynamic techniques for graph pattern matching and a decomposition into simpler subqueries.

Ansprechpartner / Contact: Matthias Barkowsky

4.3 Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

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 generic 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. We investigate how the monitoring can be adapted in a generic way based on the queries which are performed on a runtime model representing the running system. We also investigate how besides humans also software such as a system self-adaptation engines can benefit from the approach.

Ansprechpartner / Contact: Thomas Brand

4.4 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 einschränkende 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

4.5 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 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

4.7 Efficient 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 execution. Specifically, our goal is to develop an efficient 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

4.8 Run-time Verification and Validation for Self-adaptive System

The software is now the backbone of human activity. Software systems play important roles in industrial facilities, automobile, and aircraft etc.

In self-adaptive systems, the software has to deal with the rapidly changing environment conditions and the failures of its own system. How to guarantee the functional and non-functional requirements of the system during and after the adaptation process is a crucial problem.

Verification and Validation theory is wildly adopted in the whole cycle of software system development. Expanding these techniques into run-time verification and validation for self-adaptive systems is a great challenge. Run-time V&V can ensure, during or after the adaptation, system's requirements and its core qualities will not be compromised, and at the same time, the goals of adaptation process will be satisfied. Run-time V&V methods and tools are critical for the success of autonomous, autonomic, smart, self-adaptive and self-managing systems.

There are three parts in my research topic: First, to investigate the formal methods and their use at run-time, especially run-time model checking. Second, to implement the research on system

modeling and requirements/properties specification methods. Third, to integrate the run-time verification and self-adaptive system and to find out a general structure for providing assurances for the self-adaptive system in its whole life cycle.

Ansprechpartner / Contact: He Xu

5 Drittmittelprojekte / Third-Party funded Projects

5.1 DFG – Graduiertenkolleg SOAMED

Gefördert / Funded: ab 04/2016 bis 03/2019

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 Sakizoglou. [↗ 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, Soumyadip Bandyopadhyay. [↗ website](#)

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

Gefördert / Funded: ab 07/2018

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, Matthias Barkowsky.

6 Forschungsk Kooperationen / Research Cooperations

6.1 Projektpartner aus der Wissenschaft / Project Partners from Research Institutions

DFG-Projekt KorMoran 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, Mirosław 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 2018 mit folgenden externen Kooperationspartnern gemeinsame Publikationen herausgegeben oder veröffentlicht: Federico Cicozzi, Michalis Famelis, Gerti Kappel, Sébastien Mosser, Marisa Navarro, Fernando Orejas, Richard F. Paige, Alfonso Pierantonio, Elvira Pino, Arend Rensink, Rick Salay, Daniel Strüber, Gabriele Taentzer, Antonio Vallecillo, Jens Weber, und Manuel Wimmer.

Maria Maximova hat in 2018 mit dem externen Kooperationspartner Christian Krause eine gemeinsame Publikation veröffentlicht.

Sven Schneider hat in 2018 mit dem externen Kooperationspartner Fernando Orejas eine gemeinsame Publikation veröffentlicht.

7 Publikationen / Publications

7.1 Zeitschriftenartikel / Journal Articles

- [A1] Soumyadip Bandyopadhyay, Dipankar Sarkar, and Chittaranjan Mandal. Equivalence checking of Petri net models of programs using static and dynamic cut-points. *Acta Informatica*, April 2018.
- [A2] Johannes Dyck, Holger Giese, and Leen Lambers. Automatic verification of behavior preservation at the transformation level for relational model transformation. *Software & Systems Modeling*, 2018.
- [A3] Maria Maximova, Holger Giese, and Christian Krause. Probabilistic timed graph transformation systems. *Journal of Logical and Algebraic Methods in Programming*, 101:110 – 131, 2018.
- [A4] Fernando Orejas, Elvira Pino, Marisa Navarro, and Leen Lambers. Institutions for navigational logics for graphical structures. *Theoretical computer science*, 741:19 – 24, 2018. An Observant Mind : Essays Dedicated to Don Sannella on the Occasion of his 60th Birthday.
- [A5] Sven Schneider, Leen Lambers, and Fernando Orejas. Automated reasoning for attributed graph properties. *Software Tools for Technology Transfer*, June 2018.

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

- [S1] Leen Lambers, Kristopher Born, Fernando Orejas, Daniel Strüber, and Gabriele Taentzer. Initial Conflicts and Dependencies: Critical Pairs Revisited. In Reiko Heckel and Gabriele Taentzer, editors, *Graph Transformation, Specifications, and Nets*, volume 10800 of *Lecture Notes in Computer Science*, pages 105–123. Springer, 2018.
- [S2] Leen Lambers, Marisa Navarro, Fernando Orejas, and Elvira Pino. Towards a Navigational Logic for Graphical Structures. In Reiko Heckel and Gabriele Taentzer, editors, *Graph Transformation, Specifications, and Nets*, volume 10800 of *Lecture Notes in Computer Science*, pages 124–141. Springer, 2018.

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

- [K1] Christian M Adriano. Microtasking Software Failure Resolution: Early Results. In *2018 IDoESE - Doctoral Symposium at the International Conference on Empirical Software Engineering*, pages 1–4. IEEE, ACM, 2018.
- [K2] Thomas Brand and Holger Giese. Towards Generic Adaptive Monitoring. In *Proceedings of the 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 156–161, 2018.

- [K3] Thomas Brand and Holger Giese. Towards software architecture runtime models for continuous adaptive monitoring. In *Proceedings of the 13th International Workshop on Models@run.time (MRT)*, 2018.
- [K4] Federico Ciccozzi, Michalis Famelis, Gerti Kappel, Leen Lambers, Sebastien Mosser, Richard F. Paige, Alfonso Pierantonio, Arend Rensink, Rick Salay, Gabi Taentzer, Antonio Vallecillo, and Manuel Wimmer. How Do We Teach Modelling and Model-driven Engineering?: A Survey. In *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '18*, pages 122–129, New York, NY, USA, 2018. ACM.
- [K5] Federico Ciccozzi, Michalis Famelis, Gerti Kappel, Leen Lambers, Sebastien Mosser, Richard F. Paige, Alfonso Pierantonio, Arend Rensink, Rick Salay, Gabi Taentzer, Antonio Vallecillo, and Manuel Wimmer. Towards a Body of Knowledge for Model-based Software Engineering. In *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '18*, pages 82–89, New York, NY, USA, 2018. ACM.
- [K6] Sona Ghahremani, Christian M Adriano, and Holger Giese. Training Prediction Models for Rule-Based Self-Adaptive Systems. In *2018 IEEE International Conference on Autonomic Computing (ICAC)*, pages 187–192, 2018.
- [K7] Leen Lambers, Daniel Strüber, Gabriele Taentzer, Kristopher Born, and Jevgenij Hübert. Multi-Granular Conflict and Dependency Analysis in Software Engineering based on Graph Transformation. In *Proceedings of the 40th International Conference on Software Engineering, ICSE 2018*, pages 716–727, New York, NY, USA, 2018. ACM.
- [K8] Santonu Sarkar, Prateek Kandelwal, Soumyadip Bandyopadhyay, and Holger Giese. Analysis of GPGPU Programs for Data-race and Barrier Divergence. In *Proceedings of the 13th International Conference on Software Technologies - Volume 1: ICSoft*, pages 460–471. INSTICC, SciTePress, 2018.

7.4 Bücher und Tagungsbände / Books and Proceedings

- [B1] Leen Lambers and Jens Weber, editors. *Graph Transformation - 11th International Conference, ICGT 2018 Held as Part of STAF 2018, Toulouse, France, June 25-26, 2018, Proceedings*, volume 10887 of *Lecture Notes in Computer Science*. Springer, 2018.

7.5 Technische Berichte / Technical Reports

- [TR1] Sona Ghahremani. Improving Self-Healing by Estimating the Impact of Adaptation Rules on the Utility at Runtime. Technical report, Proceedings of the 10th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering, 2018.
- [TR2] Holger Giese, Maria Maximova, Lucas Sakizoglou, and Sven Schneider. Metric temporal graph logic over typed attributed graphs. Technical Report 123, Hasso Plattner Institute at the University of Potsdam, 2018.

8 Vorträge / Talks

8.1 Eingeladene Vorträge / Invited Talks

Prof. Dr. Holger Giese

Mar 2018 *How Models lead from Software Evolution to Self- Adaptive Software and Runtime Models*. 4th IEEE Intern. Conference on Big Data Computing Service and Applications, 6th IEEE Intern. Conference on Mobile Cloud Computing, Services, and Engineering 12th IEEE Intern. Symposium on Service Oriented System Engineering, Bamberg, Germany, Mar 26, 2018.

November 2018 *The Challenge of Model-Based Integration for CyberPhysical Systems*. Multi-Paradigm Modelling for CyberPhysical Systems, MPM4CPS Conference, Pisa, Italy, November 18, 2018.

December 2018 *Multi-Paradigm Modeling for Cyber-Physical Systems: Implications for Multidirectional Transformations and Synchronizations*. Dagstuhl Seminar 18491 on Multidirectional Transformations and Synchronisations, Dagstuhl, Germany, December 2, 2018.

Dr. Leen Lambers

June 2018 *Graph Transformation: Theory and Practice*. Advanced Course in Formal Methods, PhD School, University of Eindhoven, Netherlands, June 15, 2018.

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

Dr. Leen Lambers

January 2018 *How to reconcile model, transformation, and verification engineering?*. Winter modeling meeting, San Vigilio di Marebbe, Italy, January 22, 2018.

June 2018 *Multi-Granular Conflict and Dependency Analysis in Software Engineering based on Graph Transformation*. International Conference on Software Engineering, Gothenburg, Sweden, June 1, 2018.

Christian Adriano

March 2018 *Fault Failure Understanding*. Forschungskolleg Symposium, Capetown, South Africa, March 07, 2018.

September 2018 *Can a crowd locate the root-cause of a failure and suggest valid bug fixes?*. Invited talk at Humboldt University, Berlin, Germany, September 10, 2018.

October 2018 *Microtasking Fault Localization*. IDOESE Symposium at the Empirical Software Engineering Conference, Oulu, Finland, October 10, 2018.

October 2018 *Humans in the Loop of Exploratory Software Services*. Research School Fall Retreat, Neuruppin, Germany, October 18, 2018.

November 2018 *Explaining Software Failures With A Crowd Of Programmers*. HPI Symposium at SAP, Berlin, Germany, November 05, 2018.

November 2018 *Microtask Crowdsourcing Failure Resolution*. Forschungskolleg Symposium, Nanjing, China, November 21, 2018.

December 2018 *Exploring Wisdom Crowds Identify Explain Software Bugs*. HPI Symposium at SAP, Walldorf, Germany, December 10, 2018.

Sona Ghahremani

August 2018 *Utility-driven Self-adaptation Employing Adaptation Rules for Large Dynamic Architectures*. GI Dagstuhl Seminar 18343 on Software Engineering for Intelligent and Autonomous Systems (SEfIAS), Schloss Dagstuhl, Germany, August 19, 2018.

September 2018 *Training Prediction Models for Rule-based Self-adaptive Systems*. 15th International Conference on Autonomic Computing (ICAC 2018), Trento, Italy, September 06, 2018.

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)

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)

10.2 Mitarbeit in Programmkomitees / Activities in Program Committees

Prof. Dr. Holger Giese

- International Workshop on Design and Verification Techniques for Mixed Critical Systems (DVMCS 2018)
Porto, Portugal, July 26-28, 2018, [↗ website](#)
- International Workshop on Petri Nets and Software Engineering (PNSE'18)
Bratislava, Slovakia, June 25-26, 2018, [↗ website](#)
- 2nd IEEE International Workshop on Verification and Validation of Adaptive Software Systems (VVASS @ QRS 2018)
Lisbon, Portugal, July 16-20, 2018, [↗ website](#)

- 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2018)
Prague, Czech Republic, August 29-31, 2018, [↗ website](#)
- Workshop on Microservices: Science and Engineering (MSE@STAF 2018)
Toulouse, France, June 25, 2018, [↗ website](#)
- 11th International Conference on Graph Transformation (ICGT 2018)
Toulouse, France, June 25 - 26, 2018, [↗ website](#)
- 41st German Conference on Artificial Intelligence (KI2018)
Berlin, Germany, September 24-28, 2018, [↗ website](#)
- 6th International Workshop on Software Engineering for Systems-of-Systems (SESoS'18)
Gothenburg, Sweden, May 29, 2018, [↗ website](#)
- 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'18)
Gothenburg, Sweden, May 27 - June 3, 2018, [↗ website](#)
- SASO 2018 - 12th IEEE International Conference on Self-Adaptive and Self-Organizing Systems
Trento, Italy,, September 3-7, 2018, [↗ website](#)
- Modellierung 2018 - Symposium by the German Informatics Society (Gesellschaft für Informatik e.V.) on Modeling
Braunschweig, Germany, February 21-23, 2018, [↗ website](#)
- WF-IOT 2018 - IEEE World Forum on Internet of Things
Singapore, February 05-08, 2018, [↗ website](#)
- IRC 2018 - 2nd IEEE International Conference on Robotic Computing
Laguna Hills, California, USA , January 31 - February 2, 2018, [↗ website](#)
- Bx 2018 Seventh International Workshop on Bidirectional Transformations
Nice, France, April 9-12, 2018, [↗ website](#)
- International Conference on Autonomic Computing
Trento, Italy, September 37, 2018, [↗ website](#)
- The 13th International Workshop on Models@run.time
Copenhagen, Denmark, October 14, 2018, [↗ website](#)
- The 1st Joint International Workshop on Models@run.time and Self-aware Computing Systems (MRT-SeAC)
Trento, Italy, September 03, 2018, [↗ website](#)
- SEAMS 2018 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems
Gothenburg, Sweden, May 28-29, 2018, [↗ website](#)

Dr. Leen Lambers

- International Conference on Model Transformation
Toulouse, France, June, 2018, [↗ website](#)
- Models and Evolution Workshop
Copenhagen, Denmark, October, 2018, [↗ website](#)

- 15th Workshop on Model-Driven Engineering, Verification and Validation (ModeVVA 2018)
Copenhagen, Denmark, October, 2018, [↗ website](#)

10.3 Organisationstätigkeiten / Organizational Activities

Prof. Dr. Holger Giese

- Management Committee Member for Germany at the ICT COST ACTION IC1404 “Multi-paradigm Modelling for Cyber-physical Systems” (MPM4CPS)
[↗ website](#)
- Reviewer for the accreditation procedure of the Master degree program „Digital Engineering“ (M.Sc.) at the Bauhaus Universität Weimar, Germany

Dr. Leen Lambers

- Co-Chair of the 11th International Conference on Graph Transformation (ICGT 2018) Toulouse, France, June 25-26, 2018, [↗ website](#)

10.4 Gutachtertätigkeiten / Reviewing Activities

10.4.1 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)

10.4.2 Zeitschriften und Magazine / Journals

Prof. Dr. Holger Giese

- ACM Transactions on Autonomous and Adaptive Systems (TAAS)
- ACM Transactions on Software Engineering and Methodology (TOSEM)
- IEEE Transactions on Software Engineering (TSE)
- Springer Journal Software Tools for Technology Transfer (STTT)
- Springer Journal Formal Aspects of Computing (FAC)
- Software Testing, Verification and Reliability
- Software and Systems Modeling (SoSyM)
- Theoretical Computer Science

Dr. Leen Lambers

- Journal of Logical and Algebraic Methods in Programming, Elsevier
- Journal of Software and Systems Modeling, Springer
- Journal of Theoretical Computer Science, Elsevier

10.4.3 Verschiedenes / Miscellaneous

Prof. Dr. Holger Giese

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

Dr. Leen Lambers

- Mitglied Berufungskommission W3-Professur “Scalable Data Engineering” und “Computational Statistics”, HPI