Jahresbericht 2019

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Digital Engineering gGmbH

Campus Griebnitzsee
Universität Potsdam

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Jahresbericht / Annual Report 2019

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

Leiter des Fachgebiets / Head
Prof. Dr. Holger Giese

Sekretariat / Secretary
Kerstin Miers

Senior Researcher
Dr. Leen Lambers

Postdocs
Dr. Maria Maximova
Dr. Sven Schneider

Wissenschaftliche Mitarbeiter / Research Assistants
Matthias Barkowsky, M.Sc.
Dipl.-Inform. Joachim Hänsel
Lucas Sakizloglou, M.Sc. (ab 01.04.2019)
Christian Zöllner, M.Sc.

PhD-Stipendiaten / Scholarship Holders
Christian Adriano, M.Sc.
Dipl.-Wirtsch.Inf. Thomas Brand
Johannes Dyck, M.Sc. (bis 28.02.2019)
Sona Ghahremani, M.Sc.
He Xu, M.Sc.
Lucas Sakizloglou, M.Sc. (bis 31.03.2019)

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

Studentische Hilfskräfte / Student Assistants

Julian Baumann	Maximilian Böther	Hendrik Bomhardt
Tim Cech	Henrik Guhl	Jan-Eric Hellenberg
Maximilian König	Jonas Kordt	Leon Masopust
Antonius Naumann	Felix Roth	Maximilian Schulze
Nils Strassenburg	Clara Uktar	Armin Wells
Simon Wietheger		

2 Lehrveranstaltungen / Courses

2.1 Vorlesungen / Lectures

Sommersemester / Summer term 2019

- Modellierung II

Wintersemester / Winter term 2019/2020

- Modellierungssprachen und Formalismen
- Modellgetriebene Softwareentwicklung
- Software Analysieren, Testen und Verifizieren

2.2 Übungen/Projekte / Exercises/Projects

Sommersemester / Summer term 2019

- Modellierung II

Wintersemester / Winter term 2019/2020

- Modellierungssprachen und Formalismen
- Software Testen, Analysieren und Verifizieren

2.3 Seminare / Seminars

Sommersemester / Summer term 2019

- Online Monitoring of Complex Conditions for Event-based Distributed Architectures
- Self-Adaptation in Micro-Service Architectures with Kubernetes

Wintersemester / Winter term 2019/2020

- Advanced Topics for Micro Services: From Data Streams to Online Monitoring
- Advanced Topics in Self-Adaptive Systems: Online Machine Learning

3 Betreuung von Studierenden und Dissertationen / Supervised Students and Dissertations

3.1 Betreuung von Bachelorprojekten / Supervised Bachelor projects

3.1.1 Ask Your Repository! - An infrastructure to categorize and retrieve project knowledge by combining voice conversational interfaces over project knowledge-bases enhanced by hybrid crowd-machine learning classifiers.

Zeitraum / Project Period: ab 10/2018 bis 07/2019
Kooperationspartner / Project Partner: HPI School of Design Thinking ↗ website

Motivation Design Thinking teams record their intermediate results by taking photos of whiteboards and prototypes thereby producing vast amounts of image data. These images are stored on multiple platforms by different team members and thus not quickly accessible for everyone when needed. The images rarely organized and if they are, searching big folder structures still takes time. Nevertheless, Design Thinking process is iterative and relies on using previous results.

Problem Statement If images are not easily accessible the process is slowed down or the knowledge contained in the images is lost because they won't be used again.

Solution The Bachelor Project team built Ask Your Repository, a web-application that reduces the effort of organizing the images and helps Design Thinkers finding them again quickly later on in the process. In order to do that our application uses tagging to make images searchable, enhanced by machine learning and an easy to use search interface with voice assistant integration.

Ansprechpartner / Contact: Christian Adriano, Christian Zöllner. ↗ website

3.2 Betreuung von Bachelorarbeiten / Supervised Bachelor's theses

[BA1] Luise Benkert. Optimizing search queries for the retrieval of tagged images through Natural Language Processing and prioritization, June 2019.

[BA2] Jascha Beste. A Sketch Based Image Retrieval System for Design Thinking artifacts, June 2019.

[BA3] Leonhard Hennicke. Detecting Design Thinking Methods from WhiteboardImages using Image Recognition and Crowdsourcing, September 2019.

[BA4] Adrian Steppat. Text recognition on whiteboard images, July 2019.

[BA5] Arne Zerndt. Supporting Iterative Development of Voice Interfaces using a Domain Specific Language, June 2019.

[BA6] Erik Ziegler. Implementing a recommendation system for single-user image collections, June 2019.

3.3 Betreuung von Masterarbeiten / Supervised Master's theses

[MA1] Stephan Detje. Architecture Runtime Model Maintenance for Microservice Systems. Master's thesis, Hasso Plattner Institute, University of Potsdam, Germany, April 2019.

3.4 Betreuung von Dissertationen / Supervised PhD theses

3.4.1 Abgeschlossene Dissertationen / Finished PhD theses

[D1] Johannes Dyck. *Verification of Graph Transformation Systems with k-Inductive Invariants*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 12 2019.

[D2] Maria Maximova. *Behavior and Confluence Analysis of $\mathcal{M}$-Adhesive Transformation Systems using $\mathcal{M}$-Functors*. PhD thesis, Technische Universität Berlin, 12 2019.

[D3] Sven Schneider. *Deterministic pushdown automata as specifications for discrete event supervisory control in Isabelle*. PhD thesis, Technische Universität Berlin, 12 2019.

3.4.2 Laufende Dissertationen / Running PhD theses

Christian Adriano: Causal models of Software Fault Understanding with an Application to Scale Source Code Inspection Tasks for Software Bugs

Matthias Barkowsky: Modular and Incremental Global Model Management

Thomas Brand: Generic Adaptive Monitoring with Architectural Runtime Models

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

Joachim Hänsel: Testing for Self-Adaptive Software Systems

Lucas Sakizloglou: Online Checking and Prediction of Violations of Temporal Graph Conditions

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

Christian Zöllner: Modeling and Simulation of Collaborations

4 Bearbeitete Forschungsthemen / Research Topics

4.1 Causal models of Software Fault Understanding with an Application to Scale Source Code Inspection Tasks for Software Bugs

Software programmers spend from 20% to 40% of their time searching for the causes of software failures. To alleviate that, debugging techniques were developed to reduce the search space from the entire program execution to a list of suspicious program statements. However, these debugging techniques assume "perfect fault understanding", i.e., that the programmer will always recognize the software fault among the list of suspicious program statements. Since inaccurate fault understanding inevitably happens, this causes programmers to waste time generating invalid bug fixes, which in turn undermine the trust on the debugging techniques.
We investigated fault understanding in the context of code inspection tasks that are focused at a few lines of code at a time. These are typical of inspecting codes for bugs during debugging. However, since these tasks are short-lived because they take a few seconds or minutes and reasonably self-contained, i.e., leave little to no traces in logs or versioning systems. Hence, the nature of these small tasks pose a challenge to investigate them.
Our approach was to investigate ways to capture performance attributes of these tasks, while at the same time allowing to scale to hundreds of tasks. We designed an experimentation platform that allows for: (1) recruitment and qualification of programmers, (2) automatic generation of tasks from a set of template questions, and (3) incremental distribution of tasks based on the outcome of previous tasks.
We evaluated our approach through a series of experiments with real software failures from popular Open Source Software 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 incrementally deciding which tasks to allocate and to whom, 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 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 incrementally updated 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 Verification of Graph Transformation Systems with k-Inductive Invariants

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 detaillierte 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 feeback 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 Online Checking and Prediction of Violations of Temporal Graph Conditions

Our everyday lives are increasingly dependent on software and the extent of dependency has also led to a greater demand for dependability. One of the aspects of dependability refers to checking whether expectations on software behavior hold during execution. These expectations can span from propositions about a given system state to the evolution of a state over time, that is, the temporal behavior of the system. An algorithmic approach to checking relies on an unambiguous language (typically, a variant of mathematical logic) first to capture software behavior in a model, and then to check whether every possible state of the model satisfies a formal formulation of the expectations. In practice however, checking the behavior of software powering complex timed systems is often intractable and in certain cases formal models cannot even be obtained, e.g., for software that powers human-in-the-loop systems with unpredictable behavior.
A checking approach that concerns only a single, given software execution has been pursued as a light-weight alternative. This approach does not require a model of the entire behavior and is therefore more suitable to software behind complex, timed systems. We introduce a variant of this alternative approach based on graphs. We employ typed, attributed graphs to represent a state, sequences of such graphs to represent software behavior over time, and a novel, highly expressive graph-based temporal logic to express expectations on sequences, which we name temporal graph conditions. We motivate the use of graphs as a means for representation by presenting real-world use-cases where, thanks to their inherent capturing of relationships, graphs leads to a more efficient checking than the state-of-the-art. Moreover, based on our logic, our checking approach can also predict the violations of conditions when no relevant change occurs in the execution. We implement our approach and evaluate it based on both real-world and synthetic data.

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

4.9 Modeling and Simulation of Collaborations

In future large-scale cyber-physical systems, the interconnection of autonomous systems via complex software and networking will result in massive systems of systems where a huge number of heterogenous systems collaborate and act together. In this research topic, we address the challenge of modeling relevant aspects of such systems of systems. Given the high demand for safety assurances for cyber-physical systems, the thorough analysis of systems and their models is obligatory. Besides verification and validation, we propose simulation as a means to identify and resolve potential safety risks and gain further insights into how the modeled systems act and collaborate in a large systems of systems context.

Ansprechpartner / Contact: Christian Zöllner

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 servive-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 - Universtitätsmedizin 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 werkzeugunterstützt 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 Strukturdynamik (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 Strukturdynamik bieten, die besondere Relevanz im Bereich von service-orientierten Echtzeitsystemen haben. Obwohl bereits limitierte Kombinationen aus probabilistischem Verhalten, Echtzeitverhalten und Strukturdynamik existieren, und auch substantielle Fortschritte bezüglich ihrer Analyse in den letzten Jahren gemacht wurden, fehlt noch immer eine komplette Kombination, welche alle geforderten Aspekte in einem Modell vereint.
Im Projekt ist deshalb geplant, die existierenden Modelle von zeitbehafteten Graphtransformationssystemen und probabilistischen Graphtransformationssystemen zu kombinieren und zu erweitern und passende Analysemöglichkeiten durch Integrieren von existierenden Werkzeugen bereitzustellen, welche die quantitative Analyse einer größeren Klasse von Systemen und ihrer Eigenschaften erlaubt als es durch die bisher existierenden Modelle möglich ist. Neben dem neuen formalen Modell, welches alle relevanten Aspekte abdeckt, wird eine probabilistische zeitbehaftete Spezifikationslogik, eine auf dem formalen Modell basierende, abstrakte QUANTUM-Modellierungssprache, welche durch Erweitern des SoaML UML-Profils direkt die Beschreibung von service-orientierten Echtzeitsystemen ermöglicht, sowie eine verwandte visuelle Spezifikationssprache für QUANTUM-Modelle entwickelt, um die Modellierungskonzepte and 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 untersützen. 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, Sven Schneider. ↗ 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, Integrationssichten, 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 Forschungskooperationen / Research Cooperations

6.1 Projektpartner aus der Wissenschaft / Project Partners from Research Institutions

COST Action IC1404 Multi-Paradigm Modelling for Cyber-Physical Systems (MPM4CPS)
Hans Vangheluwe, University of Antwerp (Belgium) and McGill University, Montréal (Canada)
Vasco Amaral, NOVA-LINCS FCT, Universidade Nova de Lisboa (Portugal)

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)

Modelling of Ecological Systems
Cédric Gaucherel (INRA — AMAP Laboratory, Montpellier)

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: Al-Ali Rima, Amrani Moussa, Bandyopadhyay Soumyadip, Borde Etienne, Buchs Didier, Barisic Ankica, Barros Fernando, Blouin Dominique, Erata Ferhat, Goulao Miguel, Iacono Mauro, Klikovits Stefan, Leon Florin, Navarro Eva, Pelliccione Patrizio, Taveter Kuldar, Tekinerdogan Bedir, Vanherpen Ken.

Dr. Leen Lambers hat in 2019 mit folgenden externen Kooperationspartnern gemeinsame Publikationen veröffentlicht: Loli Burgueno, Federico Cicozzi, Michalis Famelis, Reiko Heckel, Gerti Kappel, Sébastien Mosser, Fernando Orejas, Richard F. Paige, Alfonso Pierantonio, Arend Rensink, Rick Salay, Daniel Strüber, Gabriele Taentzer, Antonio Vallecillo, Manuel Wimmer.

Dr. Sven Schneider hat in 2019 mit dem externen Kooperationspartner Fernando Orejas eine gemeinsame Publikation veröffentlicht.

Sona Ghahremani hat in 2019 mit folgenden externen Kooperationspartnern gemeinsame Publikationen veröffentlicht: Mirko D'Angelo, Simos Gerasimou, Johannes Grohmann, Ingrid Nunes, Evangelos Pournaras, Sven Tomforde.

7 Publikationen / Publications

7.1 Zeitschriftenartikel / Journal Articles

[A1] Loli Burgue no, Federico Ciccozzi, Michalis Famelis, Gerti Kappel, Leen Lambers, S'ebastien Mosser, Richard F. Paige, Alfonso Pierantonio, Arend Rensink, Rick Salay, Gabriele Taentzer, Antonio Vallecillo, and Manuel Wimmer. Contents for a Model-Based Software Engineering Body of Knowledge. *Software and Systems Modeling*, 18(6):3193–3205, 2019.

[A2] Johannes Dyck, Holger Giese, and Leen Lambers. Automatic verification of behavior preservation at the transformation level for relational model transformation. *Software & Systems Modeling*, 18(5):2937–2972, 2019.

[A3] Reiko Heckel, Leen Lambers, and Maryam Ghaffari Saadat. Analysis of Graph Transformation Systems: Native vs Translation-based Techniques. *Electronic Proceedings in Theoretical Computer Science*, 309:1–22, December 2019.

[A4] Leen Lambers, Kristopher Born, Jens Kosiol, Daniel Strüber, and Gabriele Taentzer. Granularity of conflicts and dependencies in graph transformation systems: A two-dimensional approach. *Journal of Logical and Algebraic Methods in Programming*, 103:105–129, 2019.

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

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

[K1] Matthias Barkowsky and Holger Giese. Hybrid Search Plan Generation for Generalized Graph Pattern Matching. In Esther Guerra and Fernando Orejas, editors, *Graph Transformation - 12th International Conference, ICGT 2019, Held as Part of STAF 2019, Eindhoven, The Netherlands, July 15-16, 2019, Proceedings*, pages 212–229, 2019.

[K2] Thomas Brand and Holger Giese. Generic adaptive monitoring based on executed architecture runtime model queries and events. In *Proceedings of the 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, 2019.

[K3] Thomas Brand and Holger Giese. Modeling approach and evaluation criteria for adaptable architectural runtime model instances. In *Proceedings of the 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2019.

[K4] M. D'Angelo, S. Gerasimou, S. Ghahremani, J. Grohmann, I. Nunes, E. Pournaras, and S. Tomforde. On learning in collective self-adaptive systems: State of practice and a 3d framework. In *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 13–24, May 2019.

[K5] Johannes Erbel, Thomas Brand, Holger Giese, and Jens Grabowski. OCCI-compliant, fully causal-connected architecture runtime models supporting sensor management. In *Proceedings of the 14th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2019.

[K6] Sona Ghahremani and Holger Giese. Performance Evaluation for Self-Healing Systems: Current Practice & Open Issues. In *2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 116–119, Los Alamitos, CA, USA, June 2019. IEEE Computer Society.

[K7] Holger Giese. Software Engineering for Smart Cyber-Physical Systems: Challenges and Opportunities. In *Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference)*, 2019.

[K8] Holger Giese, Maria Maximova, Lucas Sakizloglou, and Sven Schneider. Metric Temporal Graph Logic over Typed Attributed Graphs. In *Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic April 6-11, 2019, Proceedings*, pages 282–298, 2019.

[K9] Leen Lambers, Jens Kosiol, Daniel Strüber, and Gabriele Taentzer. Exploring Conflict Reasons for Graph Transformation Systems. In Esther Guerra and Fernando Orejas, editors, *Graph Transformation - 12th International Conference, ICGT 2019*, volume 11629 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2019.

[K10] Leen Lambers, Daniel Strüber, Gabriele Taentzer, Kristopher Born, and Jevgenij Huebert. Multi-Granular Conflict and Dependency Analysis in Software Engineering based on Graph Transformation (Summary). In Steffen Becker, Ivan Bogicevic, Georg Herzwurm, and Stefan Wagner, editors, *Software Engineering and Software Management, SE/SWM 2019, Stuttgart Germany, February 18-22, 2019*, volume P-292 of *LNI*, pages 153–154. GI, 2019.

[K11] Sven Schneider, Leen Lambers, and Fernando Orejas. A Logic-Based Incremental Approach to Graph Repair. In Reiner Hähnle and Wil M. P. van der Aalst, editors, *Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic April 6-11, 2019, Proceedings*, volume 11424 of *Lecture Notes in Computer Science*, pages 151–167. Springer, 2019.

7.4 Bücher und Tagungsbände / Books and Proceedings

7.5 Technische Berichte / Technical Reports

[TR1] Rima Al-Ali, Moussa Amrani, Soumyadip Bandyopadhyay, Ankica Barisic, Fernando Barros, Dominique Blouin, Ferhat Erata, Holger Giese, Mauro Iacono, Stefan Klikovits, and al. et. *COST IC1404 WG1 Deliverable WG1.2: Framework to Relate / Combine Modeling Languages and Techniques*. January 2019.

[TR2] M. D'Angelo, S. Gerasimou, S. Ghahremani, J. Grohmann, I. Nunes, E. Pournaras, and S. Tomforde. Learning in Collective Autonomous Systems : In Software Engineering for Intelligent and Autonomous Systems report from the GI Dagstuhl Seminar 18343. *arXiv e-prints*, page arXiv:1904.01518, April 2019.

[TR3] Holger Giese, Maria Maximova, Lucas Sakizloglou, and Sven Schneider. Metric Temporal Graph Logic over Typed Attributed Graphs: An Extended Version. Technical Report 127, Hasso Plattner Institute at the University of Potsdam, 2019.

[TR4] Stefan Klikovits, Rima Al-Ali, Moussa Amrani, Ankica Barisic, Fernando Barros, Dominique Blouin, Etienne Borde, Didier Buchs, Holger Giese, Miguel Goulao, and al. et. *COST IC1404 WG1 Deliverable WG1.1: State-of-the-art on Current Formalisms used in Cyber-Physical Systems Development*. January 2019.

[TR5] Sven Schneider, Leen Lambers, and Fernando Orejas. A logic-based incremental approach to graph repair. Technical Report 126, Hasso Plattner Institute at the University of Potsdam, 2019.

8 Vorträge / Talks

8.1 Eingeladene Vorträge / Invited Talks

Prof. Dr. Holger Giese

January 2019 *Challenges for Engineering Smart Cyber-Physical Systems*. GI-Dagstuhl Seminar 19023 on Explainable Software for Cyber-Physical Systems (ES4CPS 2019), Schloss Dagstuhl, Germany, January 6, 2019.

Dr. Leen Lambers

July 2019 *Analysis of Graph Transformation Systems: Native vs Translation-Based Techniques*. Joint Presentation with Reiko Heckel, Workshop on Graph Computation Models, Eindhoven, The Netherlands, July 17, 2019.

Dr. Sven Schneider

Oktober 2019 *A Logic-Based Incremental Approach to Graph Repair*. University of Oldenburg, Oldenburg, Oktober 14, 2019.

Sona Ghahremani

November 2019 *Application of Machine Learning in Self-adaptive Systems*. SAP Innovation Center, Potsdam, November 27, 2019.

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

Dr. Leen Lambers

February 2019 *Multi-Granular Conflict and Dependency Analysis in Software Engineering based on Graph Transformation*. Software Engineering and Software Management, SE/SWM 2019, Stuttgart, Germany, February 22, 2019.

July 2019 *Exploring Conflict Reasons for Graph Transformation Systems*. International Conference on Graph Transformation, Eindhoven, The Netherlands, July 15, 2019.

Christian Adriano

May 2019 *Microtasking Fault Understanding*. DagStuhl Seminar, DFG Research Schools Meeting, Schloss Dagstuhl, Germany, May 05, 2019.

March 2019 *Exploring Fault Localization through Microtask Crowdsourcing*. DagStuhl Seminar, Research Methods in Software Engineering, Schloss Dagstuhl, Germany, March 31, 2019.

November 2019 *Application of Machine Learning in Self Adaptive Systems. Training Prediction Models for Rule Based Self Adaptation*. SAP Innovation Center, Potsdam, Germany, November 27, 2019.

October 2019 *Prediction models of Fault Understanding with an application to Microtask Crowdsourcing*. Research School Fall Retreat, Neuruppin, Germany, October 17, 2019.

December 2019 *Building Causal Inference Models to Predict Fault Understanding and Intervene in Code Inspection Tasks*. ForschungsKolleg HPI, Potsdam, Germany, December 19, 2019.

November 2019 *Explaining Software Failures with a Crowd of Programmers*. SAP Symposium, , Berlin, Germany, November 05, 2019.

March 2019 *Can a crowd locate the root-cause of a failure and suggest valid bug fixes?*. Future Soc Symposium, Potsdam, Germany, March 23, 2019.

Matthias Barkowsky

July 2019 *Hybrid Search Plan Generation for Generalized Graph Pattern Matching*. 12th International Conference on Graph Transformation (ICGT 2019), Eindhoven, The Netherlands, July 16, 2019.

Thomas Brand

June 2019 *Generic adaptive monitoring based on executed architecture runtime model queries and events*. 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2019), Umea, Sweden, June 17, 2019.

September 2019 *Modeling approach and evaluation criteria for adaptable architectural runtime model instances*. 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS 2019), Munich, Germany, September 20, 2019.

Sona Ghahremani

May 2019 *On learning in collective self-adaptive systems: state of practice and a 3D framework*. 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Montreal, Canada, May 25, 2019.

June 2019 *Performance Evaluation for Self-Healing Systems: Current Practice & Open Issues*. 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), Umea, Sweden, June 16, 2019.

He Xu

January 2019 *Run-time Verification for Self-adaptive System*. ForschungsKolleg HPI, Potsdam, Germany, January 16, 2019.

October 2019 *Combining design time and runtime verification for self-adaptive systems*. Research School Fall Retreat, Potsdam, Germany, October 17, 2019.

Lucas Sakizloglou

September 2019 *Efficient Runtime Monitoring of Temporal Conditions*. SAP Innovation Center Talks, Potsdam, Germany, September 05, 2019.

Dr. Sven Schneider

April 2019 *Metric Temporal Graph Logic over Typed Attributed Graphs*. Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS, Prague, Czech Republic, April 6, 2019.

April 2019 *A Logic-Based Incremental Approach to Graph Repair*. Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS, Prague, Czech Republic, April 6, 2019.

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 Werkzeuge 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

- 16th IEEE International Conference on Autonomic Computing (ICAC 2019)
  Umeå, Sweden, June 16-20, 2019, ↗ website
- 12th International Conference on Graph Transformation (ICGT 2019)
  Eindhoven, The Netherlands, July 15-16, 2019, ↗ website
- The Third IEEE International Conference on Robotic Computing (IRC 2019)
  Naples, Italy, February 25-27, 2019, ↗ website

- 12th Innovations in Software Engineering Conference (ISEC 2019)
  Pune, India, February 14-16, 2019, ↗website
- 42nd edition of the German Conference on Artificial Intelligence (KI 2019)
  Kassel, Germany, September 23-26, 2019, ↗website
- Modelling is going to become Programming (M2P 2019)
  Bled, Slovenia, September 8-11, 2019, ↗website
- 2nd International Workshop on Robotics Software Engineering (RoSE'19)
  Montréal, Canada, May 27, 2019, ↗website
- 13th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2019)
  Umeå, Sweden, June 16-20, 2019, ↗website
- 14th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2019)
  Montréal, Canada, May 25-26, 2019, ↗website
- 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'19)
  Montréal, Canada, May 25-31, 2019, ↗website
- International IEEE Workshop on Verification and Validation of Adaptive Software Systems (VVASS 2019)
  Sofia, Bulgaria, July 22-26, 2019, ↗website
- IEEE 5th World Forum on Internet of Things (WF-IoT 2019)
  Limerick, Ireland, April 15-18, 2019, ↗website

Dr. Leen Lambers

- International Conference on Model Transformation
  Eindhoven, The Netherlands, July 16-17, 2019, ↗website
- International Conference on Graph Transformation
  Eindhoven, The Netherlands, July 15-16, 2019, ↗website
- Workshop on Graph Computation Models
  Eindhoven, The Netherlands, July 17, 2019, ↗website
- EduSymp@Models 2019
  Munich, Germany, September 17, 2019, ↗website
- Models and Evolution Workshop
  Munich, Germany, September 16, 2019, ↗website
- Workshop on Model-Driven Engineering, Verification and Validation (ModeVVa 2019)
  Munich, Germany, September 17, 2019, ↗website
- Workshop on Bidirectional Transformations
  Philadelphia, USA, June 4, 2019, ↗website

Sona Ghahremani

- Workshop on Evaluations and Measurements in Self-aware Computing Systems at ICAC 2019
  Umea, Sweden, June 16-20, 2019, ↗ website
- International Conference on Autonomic Computing - Poster and Demo
  Umea, Sweden, June 16-20, 2019, ↗ 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

10.4 Gutachtertätigkeiten / Reviewing Activities

10.4.1 Forschungsprojekte / Research Projects

Prof. Dr. Holger Giese

- Deutsche Forschungsemeinschaft (DFG)
- Österreichische Forschungsförderungsgesellschaft (FFG)
- European Research Council (ERC)
- European Cooperation in Science and Technology (EU COST Action)

10.4.2 Zeitschriften und Magazine / Journals

Prof. Dr. Holger Giese

- Proceedings of the IEEE
- Science of Computer Programming (SCP)
- ACM Transactions on Autonomous and Adaptive Systems (TAAS)
- IEEE Transactions on Software Engineering (TSE)
- Springer Journal Software Tools for Technology Transfer (STTT)
- Software Testing, Verification and Reliability

Dr. Leen Lambers

- Compositionality Journal
- Journal of Software and Systems Modeling, Springer
- IEEE Transactions on Software Engineering

Dr. Sven Schneider

- International Journal on Software Tools for Technology Transfer, Springer