

Jahresbericht 2014

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Softwaresystemtechnik
an der Universität Potsdam

Jahresbericht 2014

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



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

<http://www.hpi.uni-potsdam.de/giese>

Inhaltsverzeichnis

1	Personelle Zusammensetzung	1
2	Lehrveranstaltungen	3
2.1	Vorlesungen	3
2.2	Übungen/Projekte	3
2.3	Seminare	3
3	Betreuung von Studierenden und Dissertationen	4
3.1	Betreuung von Bachelorprojekten	4
3.1.1	Bachelorprojekte (abgeschlossen in 2014)	4
3.1.2	Laufende Bachelorprojekte (Abschluss in 2015)	5
3.2	Betreuung von Bachelorarbeiten	5
3.3	Betreuung von Masterprojekten	6
3.3.1	Masterprojekte (abgeschlossen in 2014)	6
3.3.2	Laufende Masterprojekte (Abschluss in 2015)	6
3.4	Betreuung von Masterarbeiten	7
3.5	Betreuung von Dissertationen	8
3.5.1	Abgeschlossene Dissertationen	8
3.5.2	Laufende Dissertationen	8
4	Bearbeitete Forschungsthemen	9
4.1	Eine Modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme	9
4.2	Verifikation selbst-adaptiver service-orientierter Systeme	9
4.3	Effiziente Ausführung von Modell-Transformationen auf Basis von Tripel-Graph-Grammatiken	10
4.4	Megamodellierung der Entwicklung von Service Orientierten Enterprise Systemen	10
4.5	Erhebung und Validierung von Requirements durch Simulation und Animation . .	10
4.6	Modeling Collaborations in adaptive Systems of Systems	11
4.7	Efficient Analysis and Querying of Evolving Large-Scale Graph Data	11
4.8	Quantitative Analyse von Service-orientierten Echtzeitsystemen	12
4.9	Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten	12
4.10	Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems . . .	13
4.11	Utility-Driven Modularized MAPE-K loop architectures for Self-Adaptive Systems	13
4.12	Testing for Self-Adaptive Software Systems	14
5	Drittmittelprojekte	15
5.1	DFG – Korrekte Modelltransformationen (KorMoran) – Fortsetzungsprojekt . . .	15
5.2	Hasso Plattner Design Thinking Research Program – Connecting Designing and Engineering Activities I / II / III	15
5.3	Hasso Plattner Design Thinking Research Program – The Design Thinking Methodology at Work - Capturing and Understanding the Interplay of Methods and Techniques	16

6	Forschungskooperationen	18
6.1	Kooperationspartner aus der Wissenschaft	18
6.2	Kooperationspartner aus der Wirtschaft	18
7	Publikationen	19
7.1	Zeitschriftenartikel	19
7.2	Beiträge zu Büchern und Sammlungen	19
7.3	Begutachtete Konferenz- und Workshopartikel	19
7.4	Bücher und Tagungsbände	20
7.5	Technische Berichte	20
7.6	Miscellaneous	20
8	Vorträge	21
8.1	Eingeladene Vorträge	21
8.2	Vorträge auf Konferenzen und Workshops	21
9	Herausgeberschaft	22
9.1	Bücher und Tagungsbände	22
10	Web-Portale und -Services	22
10.1	Self-adaptive.org	22
10.2	MDELab.org	22
10.3	CPSLab.org	22
11	Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten	23
11.1	Mitgliedschaften	23
11.2	Mitarbeit in Programmkomitees	24
11.3	Organisation von Tagungen und Workshops	24
11.4	Gutachtertätigkeiten	25
11.4.1	Forschungsprojekte	25
11.4.2	Zeitschriften und Magazine	25

1 Personelle Zusammensetzung



**TODO:
Update!**

Leiter des Fachgebiets

Prof. Dr. Holger Giese

Sekretariat

Kerstin Miers

Gastprofessor

Prof. Dr. Jürgen Dingel

Postdocs

Dr. Dominique Blouin

Dr. Leen Lambers

Wissenschaftliche Mitarbeiter

Johannes Dyck, M.Sc. (ab 01.07.2014)

Dipl.-Inform. Joachim Hänsel

Dipl.-Wirtsch.Inf. Thomas Vogel

PhD-Stipendiaten

Thomas Beyhl, M.Sc.

Dipl.-Wirtsch.Inf. Thomas Brand

Johannes Dyck, M.Sc. (bis 30.06.2014)

Sona Ghahremani, M.Sc.
Sebastian Wätzoldt, M.Sc.

Studentische Hilfskräfte

Julian Risch
Christian Zöllner

Cathleen Ramson
Jens Hildebrand

Stefan Lehmann
Christian Nicolai

**TODO:
Hiwis?**

2 Lehrveranstaltungen

2.1 Vorlesungen

Sommersemester 2014

- Modellierung II
- Software Engineering for Self-Adaptive Systems

Wintersemester 2014/2015

- Modellgetriebene Softwareentwicklung
- Software Engineering for Embedded Systems

2.2 Übungen/Projekte

Sommersemester 2014

- Modellierung II
- Software Engineering for Self-Adaptive Systems

Wintersemester 2014/2015

- Modellgetriebene Softwareentwicklung
- Software Engineering for Embedded Systems

2.3 Seminare

Sommersemester 2014

- Automated Analysis of Formal Models

Wintersemester 2014/2015

- Formal Foundations of Graph Transformation Systems

3 Betreuung von Studierenden und Dissertationen

3.1 Betreuung von Bachelorprojekten

3.1.1 Bachelorprojekte (abgeschlossen in 2014)

- *Intelligente Organisation von Design Thinking Artefakten*

Betreuer: Prof. Dr. Holger Giese, Thomas Beyhl

Studenten: Axel Kroschk, Falco Dürsch, Johannes Koch, Tim Friedrich

Abstract: Die D-School Potsdam setzt für die Dokumentation ihrer Design Thinking Projekte verschiedene Softwaretools ein. Dabei ist es den angehenden Design Thinkern selbst überlassen, welche Softwaretools und –services sie verwenden. Diese Freiheit ist nötig, um die kreative Methodik des Design Thinking nicht negativ zu beeinflussen. In der Regel wählen die Design Thinker die Softwaretools und -services aus, mit denen sie am meisten vertraut sind und die sich am besten in die Arbeitsweise innerhalb des Design Thinking Projekts einbetten. Dieses Vorgehen führt zu einer Menge von Artefakten wie Fotos, Videos, Textdokumenten und Präsentationsfolien, die über die ausgewählten Softwaretools und –services verteilt abgelegt wurden. Daher wird die D-School Potsdam in Zukunft ein System einsetzen, das als virtuelles Whiteboard agiert und die Artefakten aus den unterschiedlichen Softwaretools und –services in einem eigenen Artefaktrepository aggregiert. Das virtuelle Whiteboard bietet dabei die Möglichkeit die aggregierten Artefakte manuell zu annotieren und in Beziehung zu setzen. Dabei kommen Techniken zum Einsatz, die den Design Thinkern bekannt und daher intuitiv anwendbar sind. Diese manuellen Aktivitäten führen zu einem zusätzlichen Aufwand für die Design Thinker bei der Dokumentation ihrer Projekte, der oft als störend empfunden wird.

Dieses Bachelorprojekt geht die Herausforderung an, einzelne Artefakte automatisch zu analysieren und Beziehungen zwischen den verschiedenen Artefakten automatisch abzuleiten. Neben rein digitalen Artefakten wie Textdokumenten und Präsentationsfolien erfassen digitale Artefakte oft auch analoge Artefakte wie Post-Its oder Post-It-Cluster in Form von Fotos. Während die rein digitalen Artefakte leicht maschinenlesbar sind muss digitalen Artefakten, die analoge Artefakte erfassen, auf eine einfache und verlässliche Art und Weise eine Bedeutung zugeordnet werden, sodass die Artefakte bei einer späteren Suche wiederauffindbar sind.

Dieses Bachelorprojekt bettet sich in unser Forschungsprojekt im Rahmen des Hasso Plattner Design Thinking Research Programms ein. Das Bachelorprojekt nimmt sich der Herausforderung an, die gewonnenen Design Thinking Artefakte durchsuchbar und auswertbar aufzubereiten. Dabei müssen die Artefakte innerhalb der Dokumentation mit Semantik versehen und in Bezug gesetzt werden. Idealerweise entsteht diese Anreicherung der Dokumentation durch die Design Thinker selbst als Nebenprodukt ihrer täglichen Arbeit und behindert die kreative Methodik des Design Thinking nicht. Durch Exploration der Daten sowie durch Beobachtungen und Befragungen der angehenden Design Thinker soll ermittelt werden, welche Möglichkeiten einer Datenanreicherung anwendbar sind, wie eine entsprechende Softwarelösung zur Unterstützung der Dokumentation durch Design Thinker aussehen kann und welche Schritte hinsichtlich der Dokumentation der Design Thinking Ergebnisse zeitsparend

automatisierbar sind. Die ermittelten Anforderungen und Use-Cases sollen als Ergebnis dieses Bachelorprojekts in ein System einfließen, das durch die Projektteilnehmer erstellt und evaluiert wird.

3.1.2 Laufende Bachelorprojekte (Abschluss in 2015)

- *Ein Framework für Graphdatenbanken auf Basis von Graphtransformationen und Multi-Core Architekturen*

Betreuer: Prof. Dr. Holger Giese, Thomas Beyhl, Johannes Dyck

Studenten: Lukas Faber, Henriette Dinger, Felix Montenegro, Matthias Barkowsky

Abstract: Graphdatenbanken gewinnen, z.B. in Form von sozialen und wissenschaftlichen Netzwerken, Mediendatenbanken und Inferenzsystemen, an Popularität. Im Gegensatz zu relationalen Datenbanken unterliegen die Inhalte von graphbasierten Datenbanken einer Graphstruktur und bieten spezialisierte Graphalgorithmen an, um Suchanfragen zu vereinfachen. Die Suchanfragen selbst können dabei durch Teilgraphen beschrieben werden nach denen in der graphbasierten Datenbank zur Beantwortung der Suchanfrage gesucht wird. Die Inhalte von graphbasierten Datenbanken müssen dabei in Form von Updates aktuell gehalten werden. Suchanfragen und Updates können dabei durch sogenannte Graphtransformationen umgesetzt werden.

Suchanfragen und Updates operieren auf beliebig großen graphbasierten Daten und können beliebig nebenläufig auftreten. Da Updates, im Gegensatz zu Suchanfragen, Seiteneffekte auf die Inhalte der graphbasierten Datenbank haben, müssen Konsistenz und Atomarität sichergestellt werden was eine parallele Abarbeitung von Suchanfragen und Updates erschwert. In diesem Bachelorprojekt wird zusammen mit dem SAP Innovation Center basierend auf der Theorie von Graphtransformationen eine Graphdatenbank konzipiert und prototypisch implementiert. Dabei soll besonderer Wert auf die Parallelisierung im Hinblick auf aktuelle Multi-Core Systeme gelegt werden während andere Aspekte wie z.B. Persistenz für den Prototypen nicht betrachtet werden sollen.

3.2 Betreuung von Bachelorarbeiten

[BA1] Falco Dürsch. Bidirectional Synchronization between Project Zoom and the Inference Engine, 2014.

[BA2] Tim Friedrich. Entwicklung eines Frameworks zur Erfassung von Metadaten-Extraktion aus Design Thinking Artefakten, 2014.

[BA3] Johannes Koch. Visualisieren und Durchsuchen von Design Thinking Dokumentationen, 2014.

[BA4] Axel Kroschk. Inferring New Knowledge into Design Thinking Documentations, 2014.

3.3 Betreuung von Masterprojekten

3.3.1 Masterprojekte (abgeschlossen in 2014)

- *Model-Based Adaptation for Embedded Systems*

Betreuer: Prof. Dr. Holger Giese, Sebastian Wätzoldt

Studenten: Sebastian Stamm, Uwe Hartmann, Rakesh Kumar Sah

Abstract: In the recent years, model-based solution for the development of complex embedded real-time systems have been introduced successfully in many industry scenarios. While model-based solutions today provide a multitude of concepts, methods and tools for coping with functional properties of embedded systems, such solutions rarely exist for supporting also non-functional or timing properties. The situation is even worse when dealing with systems that exhibit flexible behavior at runtime, e.g., when different systems or subsystems can take different roles or when the architecture of such systems needs to be changed (e.g., to react on context changes or failures).

Within this master project, we will investigate how to extend model-based solutions from the field of automotive systems to develop reliable embedded real-time systems that support flexible behavior with adaptation. Based on the capabilities provided by the existing solutions and the existing tool chain the students will investigate how to reflect non-functional and timing properties also at the model-level and how to preserve these properties within the implementation. Furthermore, different modeling concepts of adapting the behavior at runtime should be evaluated.

As an application example three Robotino robots will be used where the robots can take different roles and have to be able to react on changes of their context. Each robot has to fulfill several non-functional as well as timing constraints (e.g., avoid collisions with other robots).

The master project will consist of the following concrete elements: (1) An extension of an existing development tool chain that support also non-functional aspects, (2) a runtime framework realizing the higher-level concepts present in the models, and (3) a demonstrator where the software for the robots have been developed with the extended development tool chain on top of the runtime framework.

3.3.2 Laufende Masterprojekte (Abschluss in 2015)

- *An Extensible Simulation Environment for Large-Scale Federations of Autonomous Mobile Systems*

Betreuer: Prof. Dr. Holger Giese, Sebastian Wätzoldt, Thomas Vogel

Studenten: Christian Nicolai, Claus Steuer, Daniel Gimbatschki

Abstract: Cyber-Physical Systems (CPS) have started to emerge as a consequence of the general trend toward the integration of beforehand isolated systems into large-scale federations of distributed systems comprising software and physical entities. Moreover, each system as part of such a federation is autonomous in the sense that it is developed, operated, evolved, and governed independently from the other systems. However, they all interact with each other in a dynamic and open world, which causes individual systems to dynamically

join or leave the federation over time, the federation architecture to dynamically vary, and emergent behavior of the overall federation.

An example for such a CPS is a large-scale federation of autonomous robots as shown in the figure above. In contrast to our CPS lab, in which three robots are available, this project should consider large-scale federations with hundreds or even thousands of robots. Though the robots may act autonomously and therefore independently from each other, they may also interact to jointly perform a task such as collaboratively exploring an unknown territory. Thereby, the robot may sense their environment, store important information in a context model and share that information with other autonomous robots in the system. This is illustrated in the figure by links connecting individual robots such that the whole federation can be described by a graph with robots as nodes and collaborations between robots as links between the nodes. This requires scalable means to test and validate such systems through simulation that makes use of the graph structure of such federations. For instance, while an individual autonomous robot can be rather easily simulated in isolation, the federation behavior that emerges when thousands of robots collaborate can only be effectively simulated with a scalable solution.

The goal of the project is to design and develop an extensible simulation environment that enables the validation of types of systems such as a large-scale federation of autonomous robots. Thereby, the focus is set on a scalable and resource-efficient kernel of this environment that is able to simulate the collaboration of thousands of autonomous systems (robots) following a graph-transformation based approach. Considering the picture shown above, a federation of autonomous robots can be considered as a graph while the graph structure is continuously changing when robots join or leave the whole federation or individual collaborations with other robots. Therefore, the formalism of graph transformations is suitable for (simulating) such type of systems. Scalability should be addressed by leveraging incremental approaches and concurrency in simulating federations of autonomous systems and by investigating the use of modern multi-core hardware. This simulator kernel should allow the developer to analyze traces of the simulation, for instance, by logging the simulated behavior of individual autonomous systems and of the whole federation, and to monitor properties during the simulation, such as the convergence of the behavior of all robots toward a joint goal. Moreover, a visualization of the simulated behavior should be developed that supports focusing on the individual behavior of a single autonomous system and on the emergent behavior of the overall federation. Finally, the simulator kernel should be extensible with respect to further requirements of CPS. Examples for such requirements are timing aspects such as real-time constraints for embedded systems (robots) controlling physical entities or probability aspects such as uncertainties of the reliability concerning the autonomous systems (robots) and the communication among them. In the project, one extension of the kernel to support real-time behavior should be developed.

3.4 Betreuung von Masterarbeiten

- [MA1] Sebastian Stamm. Formal Modeling of Collaborations in Self-Adaptive Systems with Timed Automata. Master's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2014.

3.5 Betreuung von Dissertationen

3.5.1 Abgeschlossene Dissertationen

- [D1] Basil Becker. *Architectural modelling and verification of open service-oriented systems of systems*. PhD thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2014.
- [D2] Gregor Berg. *Virtual prototypes for the model-based elicitation and validation of collaborative scenarios*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 2014.
- [D3] Regina Hebig. *Evolution of model-driven engineering settings in practice*. PhD thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2014.
- [D4] Stephan Hildebrandt. *On the Performance and Conformance of Triple Graph Grammar Implementations*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, June 2014.

3.5.2 Laufende Dissertationen

Thomas Beyhl: Efficient Analysis and Querying of Evolving Large-Scale Graph Data

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

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

Sona Ghahremani: Utility-Driven Modularized MAPE-K loop architectures for Self-Adaptive Systems

Joachim Hänsel: Testing for Self-Adaptive Software Systems

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

Sebastian Wätzoldt: Modeling Collaborations in adaptive Systems of Systems

4 Bearbeitete Forschungsthemen

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

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

Ansprechpartner: Thomas Vogel

4.2 Verifikation selbst-adaptiver service-orientierter Systeme

Service-orientierte Architekturen werden häufig eingesetzt, um die stetig steigende Komplexität moderner Softwaresysteme weiterhin beherrschen zu können. Eines der dabei eingesetzten Konzepte ist die lose Bindung der einzelnen Teile des Systems, was dazu führt, dass erst zum Zeitpunkt der Ausführung bekannt ist, welche Komponenten des Systems miteinander interagieren. Gleichzeitig erlaubt die lose Kopplung auf leichte Art und Weise neue Services zu einem System hinzuzufügen. Durch diese Dynamik stellen Service-orientierte Architekturen hohe Anforderungen an die Verifikationstechniken, die genutzt werden können, um diese Systeme zu verifizieren. In meiner Promotion entwickle ich Verfahren, die es durch den gezielten Einsatz von Abstraktion und Verfeinerung der Verhaltensspezifikationen erlauben, solche Systeme zu verifizieren. Das entwickelte Verfahren ist dabei robust gegenüber neu hinzugefügten Services, wodurch es sich deutlich von bestehenden Verifikationsverfahren unterscheidet. Robust meint in diesem Zusammenhang, dass sich die Schritte für die Verifikation von Änderungen des Systems auf die geänderten Teile beschränken.

Des Weiteren entwickle ich eine Verifikationsmethode, die es erlaubt, zeitbehafte Systeme mit einem potentiell unendlichen Zustandsraum bezüglich Sicherheitseigenschaften (safety properties) zu verifizieren.

Ansprechpartner: Basil Becker

4.3 Effiziente Ausführung von Modell-Transformationen auf Basis von Tripel-Graph-Grammatiken

In der modell-getriebenen Softwareentwicklung spielen Modelltransformationen eine wichtige Rolle, um verschiedene Modelle eines Systems in einander zu transformieren und nach Änderungen konsistent zu halten. Dazu muss untersucht werden, wie Modelländerungen erfasst und verarbeitet werden können, und wie die Anzahl der Operationen auf dem Zielmodell einer Synchronisation minimiert werden können, um eine effiziente Ausführung zu gewährleisten.

Eine weitere Fragestellung ist, wie die Ausdrucksmächtigkeit von TGGs erhöht werden kann, ohne die Vorteile der formalen Absicherung von Graphtransformationen zu verlieren. Da TGGs im Rahmen des Modelltransformationssystems des Fachgebiets auf Story-Diagramme abgebildet werden, werden auch Möglichkeiten zur effizienten Ausführung solcher Story-Diagramme untersucht.

Ansprechpartner: Stephan Hildebrandt

4.4 Megamodellierung der Entwicklung von Service Orientierten Enterprise Systemen

Für die Entwicklung von Software können vielfältige Ansätze verwendet werden. In den einfachsten Fällen werden einzelne Programmiersprachen genutzt, wobei ein Kompilierungsschritt genügt um ein lauffähiges Programm zu erhalten. In komplexeren Fällen, wie zum Beispiel bei Modell getriebener Entwicklung (MDE), werden in mehreren Entwicklungsschritten viele unterschiedliche Repräsentationen des Zielsystems genutzt.

Besonders die Entwicklung von Service Orientierten Architekturen (SOA), bei denen Programmodule implizit über sprachunabhängige Schnittstellen gekoppelt sind, führt zu komplexen und verschiedenartigen Entwicklungsabläufen. Das macht es schwierig sicherzustellen, dass die Entwicklung von Service Orientierten Architekturen so produktiv und effizient wie möglich ist.

Megamodelle können genutzt werden um die Beziehungen zwischen Modellen, Sprachen, Spezifikationen und Tools zu erfassen. Es ist Ziel der Promotion eine Megamodellierungsmethodik zu entwickeln, die es erlaubt Ansätze für die Entwicklung von Service Orientierten Architekturen zu planen und zu dokumentieren. Auf Basis dieser Megamodelle und empirischen Studien soll es später möglich sein Best Practices zu identifizieren.

Ansprechpartner: Regina Hebig

4.5 Erhebung und Validierung von Requirements durch Simulation und Animation

Dieses Forschungsthema setzt sich mit dem Problem der Requirements Validierung auseinander. Während der Erhebung von Anforderungen haben Requirements Engineers die Wahl, auf welche Art und Weise sie die gewonnenen Erkenntnisse modellieren wollen. Obwohl es zahlreiche Notationen und Ansätze gibt, wird immer noch sehr häufig auf Natural Language (NL) zurückgegriffen. Obwohl NL zumeist nur aus einer Menge an einfachen Aussagen in Satzform besteht und daher nur schwer zu verwalten ist, zeichnet sich dieser Ansatz eine einfache Präsentation und hohe Verständlichkeit aus. Sobald komplexe Ansätze mit entsprechenden Notationen genutzt werden,

können modellierte Erkenntnisse nur schwer mit den Endnutzern validiert werden, da diese die eingesetzten Notationen nicht kennen und daher den Inhalt nicht verstehen. Um dieses Problem zu umgehen, beschäftigt sich dieses Dissertationsthema mit der interaktiven Simulation und Animation solcher Modelle, um sie zum einen den Endnutzern verständlich zu präsentieren und zum anderen interaktiv zu erweitern, d. h. um neue Erkenntnisse zu bereichern.

Ansprechpartner: Gregor Gabrysiak

4.6 Modeling Collaborations in adaptive Systems of Systems

An increasing demand on functionality and flexibility leads to an integration of beforehand isolated system solutions building a so-called *System of Systems* (SoS). Furthermore, the overall SoS should be adaptive to react on changing requirements and environmental conditions. Due SoS are composed of different independent systems that may join or leave the overall SoS at arbitrary point in times, the SoS structure varies during the systems lifetime and the overall SoS behavior emerges from the capabilities of the contained subsystems. In such complex system ensembles new demands of understanding the interaction among subsystems, the coupling of shared system knowledge and the influence of local adaptation strategies to the overall resulting system behavior arise.

In my research, I focus on modeling the interactions between system parts inside a SoS in so-called collaborations. Furthermore, on basis of a formal modeling language, I investigate the verification capabilities of the modeled collaborations to detect violations of the interaction specification, coupling of shared knowledge, and correct emergent behavior of the overall SoS. Beside the verification, I gain intention to simulation support for adaptive SoS, where the behavior is specified in form of graph transformation systems.

Ansprechpartner: Sebastian Wätzoldt

4.7 Efficient Analysis and Querying of Evolving Large-Scale Graph Data

Graph data is omnipresent in our everyday life. For example, web search engines rely on the graph constituted by websites and links between website. Social networks constitute a graph based on the users of the social network and their friendship relationships. Models and source code can be considered as abstract syntax graphs. Therefore, graphs include additional implicit knowledge that can be queried such as clusters of semantically related websites, users with similar interests in social networks, or knowledge about employed software design patterns and recommended software refactorings in abstract syntax graphs. However, querying this implicit knowledge on demand can be computationally expensive for large-scale graph data, especially when batch querying is employed. In this PhD thesis, I present an approach to incrementally maintain graph-based views that consist of pre-computed query answers. Since large-scale graph data evolves over time very frequently by adding, removing, and modifying nodes and edges in the graph, the views need to be updated regularly, either immediately or deferred. The contribution of this thesis is a) a modeling language that enables to exploit intermediate query results to pre-compute higher-level query results and b) an efficient and scalable incremental maintenance approach for graph-based views. Finally, several case studies with different graph data characteristics are presented to evaluate the approach.

Ansprechpartner: Thomas Beyhl

4.8 Quantitative Analyse von Service-orientierten Echtzeitsystemen

Eine der wichtigsten Herausforderungen in der Entwicklung von Service-orientierten Systemen ist die Vorhersage und die Zusicherung von nicht-funktionalen Eigenschaften, wie Ausfallsicherheit und Verfügbarkeit von zusammengesetzten, interorganisationellen Diensten. Diese Systeme sind oft charakterisiert durch eine Vielzahl von inhärenten Unsicherheiten, welche sowohl in der Modellierung als auch in der Analyse eine Rolle spielen. Ziel dieses Projektes ist es ein probabilistisches, Zeit-behaftetes Modell zu entwickeln, welches es ermöglicht quantitative Aussagen über nicht-funktionale Eigenschaften von Service-orientierten Echtzeitsystemen mittels formaler Methoden zu treffen. Als grundlegendes formales Modell für nicht-funktionale Eigenschaften werden sogenannte Interval Probabilistic Timed Automata (IPTA) benutzt. Dieses Modell besitzt sowohl ausreichende Ausdrucksstärke für eine realistische und modulare Spezifikation als auch geeignete formale Methoden zur Bestimmung von quantitativen Sicherheits- und Zuverlässigkeitseigenschaften. Als technisches Mittel für die quantitative Analyse wird probabilistisches Model Checking, speziell probabilistische Zeit-beschränkte Erreichbarkeitsanalyse und Bestimmung von Erwartungswerten für Kosten und Vergütungen eingesetzt. Um die quantitative Analyse mittels probabilistischem Model Checking durchzuführen, wurde eine Erweiterung des PRISM-Werkzeuges zur Modellierung und Analyse von IPTA entwickelt.

In diesem Projekt wurde eine Zusammenarbeit mit der Gruppe von Prof. Marta Kwiatkowska in Oxford initiiert und ein Antrag auf Sachbeihilfe bei der Deutschen Forschungsgemeinschaft (DFG) verfasst.

Ansprechpartner: Christian Krause

4.9 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: Johannes Dyck

4.10 Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

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

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

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

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

Ansprechpartner: Thomas Brand

4.11 Utility-Driven Modularized MAPE-K loop architectures for Self-Adaptive Systems

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

We peruse a hybrid adaptation policy in a modularized multi-concern self-adaptive system where each module is allowed to apply the type of adaptation strategy which fits better to its concern. The goal of the hybrid adaptation is to benefit from the strong points of each methodology and let them compensate for each others weaknesses. Regarding the challenges caused by modularization such

as the order in which modules need to be executed and dependencies among modules, we intend to apply a policy in which we avoid the probable unwanted interferences. We also propose benefiting from a utility-driven policy to obtain the optimal ordering scenarios for executing modules, for that purpose we need to assign utility values to each module and define a utility function which assigns a real-valued scalar representing the desirability of system configuration identifying how good that specific state or configuration is as opposed to others. The proposed approach allows to define a utility function for architectural runtime models, to specify the possible improvements or repairs of the architecture by means of rules, and finally to achieve the incremental triggering of the rules according to their impact on the utility. We were able to show that the suggested utility-driven rule-based adaptation works incrementally and that under certain assumptions it always results in the optimal improvement of the utility over time.

Ansprechpartner: Sona Ghahremani

4.12 Testing for Self-Adaptive Software Systems

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

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

Ansprechpartner: Joachim Hänsel

5 Drittmittelprojekte

5.1 DFG – Korrekte Modelltransformationen (KorMoran) – Fortsetzungsprojekt

Gefördert: ab 08/2013

Drittmittelgeber: DFG

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

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

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

Ansprechpartner: Holger Giese, Leen Lambers, Johannes Dyck

5.2 Hasso Plattner Design Thinking Research Program – Connecting Designing and Engineering Activities I / II / III

Gefördert: ab 10/2011

Drittmittelgeber: Hasso Plattner Design Thinking Research Program (HPDTRP)

Dies ist ein Projekt aus dem HPI - Stanford Design Thinking Research Program, einem Kooperationsprogramm zwischen der Stanford University School of Engineering und des Hasso-Plattner-Instituts.

Die verschiedenen Design Thinking Aktivitäten resultieren in einer Vielzahl analoger als auch digitaler Artefakte, die Arbeitszustände widerspiegeln und als Medium für die Kommunikation verwendet werden. Dabei enthalten sie Designentscheidungen, Beobachtungen und Erkenntnisse. Wenn es zum Engineering kommt und manchmal auch wenn Design Thinking Aktivitäten erneut durchgeführt werden, sind die Informationen, die durch die Artefakte mitgeliefert oder verwaltet werden nicht genug. Hinzu kommt, dass frühere Artefakte, deren Kontext, Abhängigkeiten zwischen den Artefakten, die Design Rationale und viele andere Details, die das Artefakt selbst nicht speichert, benötigt werden. Diese Informationen sind oft nur schwer oder gar nicht wiederherzustellen.

Im ersten Teil des Research Projekts stellen wir vor wie Design-Artefakte und deren Abhängigkeiten in einer kosten-effizienten Art und Weise organisiert werden sollten um das Extrahieren der benötigten Informationen für das Engineering als auch Design Thinking zu ermöglichen. Im zweiten Teil des Research Projekts untersuchen wir wie Traceability Informationen mit möglichst geringem Aufwand aber großen Nutzen erfasst werden können. Dazu untersuchen wir welche Anforderungen eine geeignete Dokumentationsplattform ermöglichen sollte. Im dritten Teil des Research Projekts untersuchen wir welche Informationen automatisch aus der erfassten Dokumentation geschlussfolgert werden können. Dazu wird eine geeignete Modellierungssprache und Ausführungsumgebung entworfen mit deren Hilfe solche automatischen Rückschlüsse modelliert und ausgeführt werden können.

Ansprechpartner: Holger Giese, Thomas Beyhl

5.3 Hasso Plattner Design Thinking Research Program – The Design Thinking Methodology at Work - Capturing and Understanding the Interplay of Methods and Techniques

Gefördert: ab 10/2014

Drittmittelgeber: Hasso Plattner Design Thinking Research Program (HPDTRP)

The methodology of Design Thinking (DT) suggests a repertoire of methods and techniques to solve wicked problems in terms of innovative solutions. The application of these methods and techniques leads to a concrete DT methodology at work. Which concrete methods and techniques have been employed is of special interest to stakeholders such as students, teachers, project managers and researchers. However, the DT methodology does not determine much concerning the order of applying methods and techniques. Furthermore, capturing this data is hard as the subjectively perceived and objectively employed methodology at work may differ and the capturing should not be perceived as obstructive. In this research proposal, we suggest to investigate how to capture and visualize employed DT methodologies at work to understand them in detail. For example, this includes capturing employed methodology phases, methods and techniques, as well as artifacts in which the outcome of certain activities is manifested. Visualizing the captured methodology at work enables us to compare them and investigate suspicious situations. We plan to conduct interviews to evaluate our capturing results and derived observations. We expect

that the outcome of our research enables us to compare DT methodologies at work to identify interesting commonalities and differences.

Ansprechpartner: Holger Giese, Thomas Beyhl

6 Forschungsk Kooperationen

6.1 Kooperationspartner aus der Wissenschaft

Jürgen Dingel (Queen's University, Kingston, Canada)
Mercator-Fellow im Rahmen der Fortsetzungsphase des KorMoran-Projekts

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

Paola Inverardi und Henry Muccini (Universität L'Aquila, Italien)
Analyse von Softwarearchitekturen

Wilhelm Schäfer (Universität Paderborn)
Mechatronic UML

Tingting Han und Marta Kwiatkowska (Universität Oxford)
Modellierung von zeitbehaftetem probabilistischem Verhalten

Christian Krause (SAP Innovation Center)
Graphdatenbanken

6.2 Kooperationspartner aus der Wirtschaft

D-LABS GmbH, Potsdam
Design Consulting für Softwareprodukte

dSpace GmbH, Paderborn
Automotives Software Engineering, Sicherheitsanalysen, Verifikation von Echtzeitverhalten

Hella KG Hueck & Co., Lippstadt
Automotives Software Engineering, Sicherheitsanalysen

SAP AG
Model-driven Language Engineering

SAP Deutschland AG & Co. KG, Walldorf
Erfassung von modelgetriebenen Entwicklungsansätzen in der Praxis

Capgemini
Erfassung von modelgetriebenen Entwicklungsansätzen in der Praxis

Ableton AG
Erfassung von modelgetriebenen Entwicklungsansätzen in der Praxis

VCat Consulting GmbH
Erfassung von modelgetriebenen Entwicklungsansätzen in der Praxis

Wasserwacht Berlin vom Deutschen Roten Kreuz
Anforderungsanalyse

7 Publikationen

7.1 Zeitschriftenartikel

- [A1] Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese. Software Engineering for Self-Adaptive Systems: Assurances (Dagstuhl Seminar 13511). *Dagstuhl Reports*, 3(12):67–96, 2014.
- [A2] Hartmut Ehrig, Ulrike Golas, Annegret Habel, Leen Lambers, and Fernando Orejas. M-Adhesive Transformation Systems with Nested Application Conditions, Part 1: Parallelism, Concurrency and Amalgamation. *Mathematical Structures in Computer Science*, 24(04), 2014.
- [A3] Holger Giese, Stephan Hildebrandt, and Leen Lambers. Bridging the gap between formal semantics and implementation of triple graph grammars. *Software and Systems Modeling*, 13(1):273–299, 2014.
- [A4] Thomas Vogel and Holger Giese. Model-Driven Engineering of Self-Adaptive Software with EUREMA. *ACM Trans. Auton. Adapt. Syst.*, 8(4):18:1–18:33, January 2014.

7.2 Beiträge zu Büchern und Sammlungen

- [S1] Amel Bennaceur, Robert France, Giordano Tamburrelli, Thomas Vogel, Pieter J. Mosterman, Walter Cazzola, Fábio M. Costa, Alfonso Pierantonio, Matthias Tichy, Mehmet Aksit, Pär Emmanuelson, Huang Gang, Nikolaos Georgantas, and David Redlich. Mechanisms for Leveraging Models at Runtime in Self-adaptive Software. In Nelly Bencomo, Robert France, Betty H.C. Cheng, and Uwe Assmann, editors, *Models@run.time*, volume 8378 of *Lecture Notes in Computer Science (LNCS)*, pages 19–46. Springer, 2014.
- [S2] Holger Giese, Nelly Bencomo, Liliana Pasquale, AndresJ. Ramirez, Paola Inverardi, Sebastian Wätzoldt, and Siobhan Clarke. Living with Uncertainty in the Age of Runtime Models. In Nelly Bencomo, Robert France, Betty H.C. Cheng, and Uwe Assmann, editors, *Models@run.time*, volume 8378 of *Lecture Notes in Computer Science*, pages 47–100. Springer International Publishing, 2014.

7.3 Begutachtete Konferenz- und Workshopartikel

- [K1] Anthony Anjorin, Alcino Cunha, Holger Giese, Frank Hermann, Arend Rensink, and Andy Schürr. BenchmarX. In K. Selçuk Candan, Sihem Amer-Yahia, Nicole Schweikardt, Vassilis Christophides, and Vincent Leroy, editors, *Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference (EDBT/ICDT 2014), Athens, Greece, March 28, 2014*, volume 1133 of *CEUR Workshop Proceedings*, pages 82–86. CEUR-WS.org, 2014.
- [K2] Joachim Hänsel. Model Based Test Case Generation with Metaheuristics for Networks of Timed Automata. In *Proceedings of the 7th International Workshop on Search-Based Software Testing, SBST 2014*, pages 31–34. ACM, 2014.

- [K3] Christian Krause, Matthias Tichy, and Holger Giese. Implementing Graph Transformations in the Bulk Synchronous Parallel Model. In Stefania Gnesi and Arend Rensink, editors, *17th International Conference on Fundamental Approaches to Software Engineering (FASE), Grenoble, France*, volume 8411 of *Lecture Notes in Computer Science (LNCS)*, pages 325–339. Springer, 7-10 June 2014.
- [K4] Axel Menning, Thomas Beyhl, Holger Giese, Ulrich Weinberg, and Claudia Nicolai. Introducing the LogCal: Template-Based Documentation Support for Educational Design Thinking Projects. In *International Conference on Engineering and Product Design Education*, 2014.
- [K5] Thomas Vogel and Holger Giese. On Unifying Development Models and Runtime Models. In *Proceedings of the 9th International Workshop on Models@run.time at the 17th IEEE/ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain*, volume 1270 of *CEUR Workshop Proceedings*, pages 5–10. CEUR-WS.org, September 2014.

7.4 Bücher und Tagungsbände

- [B1] Holger Giese and Barbara König, editors. *Graph Transformation, 7th International Conference, ICGT 2014, Held as Part of STAF 2014, York, UK, July 22-24, 2014. Proceedings*, volume 8571 of *Lecture Notes in Computer Science (LNCS)*. Springer, July 2014.

7.5 Technische Berichte

- [TR1] Johannes Dyck. Symbolic Representation and Constraint Reasoning in Invariant Checking. Technical Report 83, Proceedings of the 7th Ph.D. Retreat of the HPI Research School on Service-oriented Systems Engineering, Hasso Plattner Institute, University of Potsdam (Fall 2013 Workshop), 2014.
- [TR2] Thomas Vogel, Matthias Tichy, and Alessandra Gorla, editors. *Report from the GI Dagstuhl Seminar 14433: Software Engineering for Self-Adaptive Systems*, number 2014:02 in Research Reports in Software Engineering and Management. Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Sweden, December 2014.

7.6 Miscellaneous

- [M1] Johannes Dyck. Graph Transformation Systems and Verification with k-Inductive Invariants. In Proceedings of the 8th Joint Workshop of the German Research Training Groups in Computer Science, Dagstuhl 2014, Pro Business, Berlin, 2014. Extended Abstract.

8 Vorträge

TODO: Update!

8.1 Eingeladene Vorträge

Prof. Dr. Holger Giese

March 2013 *Extensions of Graph Transformation Systems for Timed, Continuous, and Probabilistic Behavior*. 12th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2013), Rome, Italy, March 24, 2013.

8.2 Vorträge auf Konferenzen und Workshops

Thomas Beyhl

March 2014 *Connecting Designing and Engineering Activities II*. 12. HPDTRP Community Building Workshop, Stanford University, CA, USA, March 24, 2014.

September 2014 *Introducing the LogCal: Template-Based Documentation Support for Educational Design Thinking Projects*. International Conference on Engineering and Product Design Education, Twente, Netherlands, September 4, 2014.

Johannes Dyck

April 2014 *Induction vs. k-Induction in Graph Transformation Systems*. Spring Workshop and Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, April 4, 2014.

June 2014 *Graph Transformation Systems and Verification with k-Inductive Invariants*. Joint Workshop of the German Research Training Groups in Computer Science, Dagstuhl, Germany, June 17, 2014.

Thomas Vogel

September 2014 *Model-Driven Engineering of Self-Adaptive Software with EUREMA*. GI Dagstuhl Seminar on Control Theory Meets Software Engineering, Seminar 14382 at Schloss Dagstuhl, Wadern, Germany, September 14-19, 2014.

9 Herausgeberschaft

9.1 Bücher und Tagungsbände

- [B1] Holger Giese and Barbara König, editors. *Graph Transformation, 7th International Conference, ICGT 2014, Held as Part of STAF 2014, York, UK, July 22-24, 2014. Proceedings*, volume 8571 of *Lecture Notes in Computer Science (LNCS)*. Springer, July 2014.

10 Web-Portale und -Services

10.1 Self-adaptive.org

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

10.2 MDELab.org

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

10.3 CPSLab.org

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

11 Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten

11.1 Mitgliedschaften

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)

Basil Becker

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

Gregor Gabrysiak

- Mitglied der IEEE
- Mitglied der IEEE Computer Society

Regina Hebig

- Mitglied der Association for Computing Machinery (ACM)
- Mitglied der folgenden Special Interest Groups: SIGSOFT

Thomas Vogel

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

Sebastian Wätzoldt

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

11.2 Mitarbeit in Programmkomitees

TODO: Update!

Prof. Dr. Holger Giese

- 16th International Conference on Fundamental Approaches to Software Engineering (FASE)
Rome, Italy, March 16-24, 2013, [↗ website](#)
- 16th International Conference on Model Driven Engineering Languages and Systems (MODELS)
– Foundations Track
Miami, Florida, USA, September 29 - October 4, 2013, [↗ website](#)
- European Conference on Modelling Foundations and Applications (ECMFA)
Montpellier, France, July 1-5, 2013, [↗ website](#)
- 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)
May 20-21, 2013, San Francisco, USA, [↗ website](#)
- 12th International Workshop on Foundations of Coordination Languages and Self Adaptation (FOCLASA)
Malaga, Spain, September 11, 2013, [↗ website](#)
- International Workshop on Modeling and Business Environments (ModBE)
Milano, Italy, June 24, 2013, [↗ website](#)
- 28th ACM Symposium on Applied Computing (SAC 2013) – 8th Track on Dependable and Adaptive Distributed Systems (DADS)
Coimbra, Portugal, March 18-22, 2013, [↗ website](#)

Dr. Leen Lambers

- 30th International Conference on Software Maintenance and Evolution
Victoria, Canada, September 28-October 3, 2014, [↗ website](#)
- 3rd International Workshop on the Verification of Model Transformation
York, UK, July 21, 2014, [↗ website](#)
- 7th International Conference on Graph Transformation
York, UK, July 22-24, 2014, [↗ website](#)

Thomas Vogel

- 9th International Workshop on Models@run.time (MRT 2014)
Valencia, Spain, September 30, 2014, [↗ website](#)

11.3 Organisation von Tagungen und Workshops

Prof. Dr. Holger Giese

- 7th International Conference on Graph Transformation (ICGT 2014)
York, UK, July 22-24, 2014, [↗ website](#)

- Workshop on Current trends, challenges, and solutions in Model-Driven Engineering
Potsdam, Germany, May 30, 2014, [↗ website](#)

Dr. Leen Lambers

- Publicity Chair of 7th International Conference on Graph Transformation (ICGT 2014)
York, UK, July 22-24, 2014, [↗ website](#)

Thomas Vogel

- GI Dagstuhl Seminar “Software Engineering for Self-Adaptive Systems”
Seminar 14433 at Schloss Dagstuhl, Wadern, Germany, October 19-24, 2014, [↗ website](#)
- Workshop on Current trends, challenges, and solutions in Model-Driven Engineering
Potsdam, Germany, May 30, 2014, [↗ website](#)

11.4 Gutachtertätigkeiten

11.4.1 Forschungsprojekte

Prof. Dr. Holger Giese

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

11.4.2 Zeitschriften und Magazine

Prof. Dr. Holger Giese

- Science of Computer Programming (Zeitschrift)
- Transactions on Software Engineering and Methodology (Zeitschrift)
- Formal Aspects of Computing (Zeitschrift)
- IEEE Computer (Magazin)
- IEEE Robotics and Automation (Magazine)
- IEEE Software
- IEEE Transactions on Control Systems Technology
- IEEE Transactions on Industrial Informatics

- IEEE Transactions on Software Engineering
- Information and Software Technology
- Journal of Systems and Software (Zeitschrift)
- Journal of Visual Languages and Computing (Zeitschrift)
- Requirements Engineering (Zeitschrift)
- Simulation: Transactions of the Society for Modeling and Simulation International (Zeitschrift)
- Software Quality Journal (Zeitschrift)
- Software and Systems Modeling (Zeitschrift)
- Journal of Software Engineering for Robotics (JOSER)
- International Journal on Software Tools for Technology Transfer (STTT)
- International Journal of Aerospace Engineering (IJAE)
- Information Systems (IS) (Zeitschrift)
- Concurrency and Computation: Practice and Experience (Zeitschrift)
- Mechatronics (Zeitschrift)
- Algorithms (Zeitschrift)

Dr. Leen Lambers

- IEEE Transactions on Software Engineering (Zeitschrift)

Thomas Vogel

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