

# Jahresbericht 2015

Prof. Dr. Holger Giese  
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für  
Softwaresystemtechnik  
an der Universität Potsdam



# Jahresbericht 2015

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

<b>1 Personelle Zusammensetzung</b>	<b>1</b>
<b>2 Lehrveranstaltungen</b>	<b>3</b>
2.1 Vorlesungen . . . . .	3
2.2 Übungen/Projekte . . . . .	3
2.3 Seminare . . . . .	3
<b>3 Betreuung von Studierenden und Dissertationen</b>	<b>4</b>
3.1 Betreuung von Bachelorprojekten . . . . .	4
3.1.1 Bachelorprojekte (abgeschlossen in 2015) . . . . .	4
3.2 Betreuung von Bachelorarbeiten . . . . .	4
3.3 Betreuung von Masterprojekten . . . . .	5
3.3.1 Masterprojekte (abgeschlossen in 2015) . . . . .	5
3.4 Betreuung von Dissertationen . . . . .	6
3.4.1 Laufende Dissertationen . . . . .	6
<b>4 Bearbeitete Forschungsthemen</b>	<b>7</b>
4.1 Eine Modellgetriebene Infrastruktur für selbst-adaptive service-orientierte Systeme	7
4.2 Modeling Collaborations in adaptive Systems of Systems . . . . .	7
4.3 Efficient Analysis and Querying of Evolving Large-Scale Graph Data . . . . .	8
4.4 Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten	8
4.5 Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems . . .	9
4.6 Utility-Driven Modularized MAPE-K loop architectures for Self-Adaptive Systems	9
4.7 Testing for Self-Adaptive Software Systems . . . . .	10
<b>5 Drittmittelprojekte</b>	<b>11</b>
5.1 DFG – Korrekte Modelltransformationen (KorMoran) – Fortsetzungsprojekt . . .	11
5.2 Hasso Plattner Design Thinking Research Program – The Design Thinking Methodology at Work - Capturing and Understanding the Interplay of Methods and Techniques . . . . .	11
<b>6 Forschungsk Kooperationen</b>	<b>13</b>
6.1 Kooperationspartner aus der Wissenschaft . . . . .	13
6.2 Kooperationspartner aus der Wirtschaft . . . . .	13
<b>7 Publikationen</b>	<b>15</b>
7.1 Zeitschriftenartikel . . . . .	15
7.2 Beiträge zu Büchern und Sammlungen . . . . .	15
7.3 Begutachtete Konferenz- und Workshopartikel . . . . .	15
7.4 Technische Berichte . . . . .	16
7.5 Miscellaneous . . . . .	16

---

<b>8</b>	<b>Vorträge</b>	<b>17</b>
8.1	Eingeladene Vorträge . . . . .	17
8.2	Vorträge auf Konferenzen und Workshops . . . . .	17
<b>9</b>	<b>Web-Portale und -Services</b>	<b>19</b>
9.1	Self-adaptive.org . . . . .	19
9.2	MDELab.org . . . . .	19
9.3	CPSLab.org . . . . .	19
<b>10</b>	<b>Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten</b>	<b>20</b>
10.1	Mitgliedschaften . . . . .	20
10.2	Mitarbeit in Programmkomitees . . . . .	20
10.3	Organisation von Tagungen und Workshops . . . . .	22
10.4	Gutachtertätigkeiten . . . . .	22
10.4.1	Forschungsprojekte . . . . .	22
10.4.2	Zeitschriften und Magazine . . . . .	22

## 1 Personelle Zusammensetzung



### **Leiter des Fachgebiets**

Prof. Dr. Holger Giese

### **Sekretariat**

Kerstin Miers

### **Senior Researcher**

Dr. Leen Lambers (ab 01.07.2015)

### **Postdocs**

Dr. Dominique Blouin

Dr. Leen Lambers (bis 30.06.2015)

### **Wissenschaftliche Mitarbeiter**

Thomas Beyhl, M.Sc. (ab 01.04.2015)

Johannes Dyck, M.Sc.

Dipl.-Inform. Joachim Hänsel

Dipl.-Wirtsch.Inf. Thomas Vogel

Sebastian Wätzoldt, M.Sc.

**PhD-Stipendiaten**

Thomas Beyhl, M.Sc. (bis 31.03.2015)

Dipl.-Wirtsch.Inf. Thomas Brand

Sona Ghahremani, M.Sc.

**Studentische Hilfskräfte**

Matthias Barkowsky

Micheal Fabian

Tobias Knösche

Marianne Thieffry

Tobias Dürschmid

Robert Fruth

Christian Nicolai

Christopher Weyand

Pascal Führlich

Paul Geppert

Lukas Pirl

Christian Zöllner



## **2 Lehrveranstaltungen**

### **2.1 Vorlesungen**

#### **Sommersemester 2015**

- Modellierung II
- Software Engineering for Self-Adaptive Systems

#### **Wintersemester 2015/2016**

- Modellierung I
- Softwarequalität
- Testen, Verifizieren und Analysieren von Software

### **2.2 Übungen/Projekte**

#### **Sommersemester 2015**

- Modellierung II
- Software Engineering for Self-Adaptive Systems

#### **Wintersemester 2015/2016**

- Modellierung I
- Softwarequalität
- Testen, Verifizieren und Analysieren von Software

### **2.3 Seminare**

#### **Wintersemester 2015/2016**

- Self-Awareness and Self-Adaptive Software

### 3 Betreuung von Studierenden und Dissertationen

#### 3.1 Betreuung von Bachelorprojekten

##### 3.1.1 Bachelorprojekte (abgeschlossen 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] Matthias Barkowsky. Implementierung gleichzeitiger Lese- und Schreibzugriffe auf Graphdaten mithilfe von Versionen, 2015. Bachelor Thesis.
- [BA2] Henriette Dinger. Unterstützung verschiedener Synchronisationsstrategien und Graphtransformationsengines in einem Framework für Graphdatenbanken, 2015. Bachelor Thesis.
- [BA3] Lukas Faber. Development of a Benchmarking Framework and Implementation of Reference Strategies for Evaluation of Synchronisation Strategies for Graph Transformations, 2015. Bachelor Thesis.
- [BA4] Felix Montenegro. Synchronisation von Graphtransformationen durch gegenseitigen Abschluss überschneidender Suchanfragen, 2015. Bachelor Thesis.

### 3.3 Betreuung von Masterprojekten

#### 3.3.1 Masterprojekte (abgeschlossen 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 Dissertationen

#### 3.4.1 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 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**

### 4.3 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**

### 4.4 Graphtransformationssysteme und Invariant-Checking mit k-induktiven Invarianten

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

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

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

**Ansprechpartner: Johannes Dyck**

## 4.5 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**

## 4.6 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**

#### **4.7 Testing for Self-Adaptive Software Systems**

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

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

**Ansprechpartner: Joachim Hänsel**



## 5 Drittmittelprojekte

### 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 – 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

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

Jürgen Dingel (Queen's University, Kanada)

Semantikbewahrung für Modelltransformationen

### 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

SAP AG

Runtime Data-Driven Software Evolution in Enterprise Software Ecosystems

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] Regina Hebig and Holger Giese. On the complex nature of mde evolution and its impact on changeability. *Software & Systems Modeling*, pages 1–24, 2015.

### 7.2 Beiträge zu Büchern und Sammlungen

- [S1] Thomas Beyhl and Holger Giese. Connecting Designing and Engineering Activities II. In Hasso Plattner, Christoph Meinel, and Larry Leifer, editors, *Design Thinking Research - Building Innovators*, pages 211–239. Springer International Publishing, 2015.
- [S2] Thomas Beyhl and Holger Giese. Connecting Designing and Engineering Activities III. In *Design Thinking Research - Making Design Thinking Foundational*, pages 265–290. Springer, 2015.

### 7.3 Begutachtete Konferenz- und Workshopartikel

- [K1] Thomas Beyhl and Holger Giese. Traceability Recovery for Innovation Processes. In *Proceedings of the 8th International Symposium on Software and Systems Traceability*, pages 22–28. IEEE, 2015.
- [K2] Dominique Blouin, Gilberto Ochoa-Ruiz, Yvan Eustache, and Jean-Philippe Diguët. KaoLin: A system-level AADL tool for FPGA design reuse, upgrade and migration. In *2015 NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2015, Montreal, QC, Canada, June 15-18, 2015*, pages 1–8, 2015.
- [K3] Johannes Dyck and Holger Giese. Inductive Invariant Checking with Partial Negative Application Conditions. In Francesco Parisi-Presicce and Bernhard Westfechtel, editors, *Graph Transformation*, volume 9151 of *Lecture Notes in Computer Science*, pages 237–253. Springer International Publishing, 2015.
- [K4] Johannes Dyck, Holger Giese, Leen Lambers, Sebastian Schlesinger, and Sabine Glesner. Towards the Automatic Verification of Behavior Preservation at the Transformation Level for Operational Model Transformations. In *Fourth Workshop on the Analysis of Model Transformations*, CEUR Workshop Proceedings, 2015.
- [K5] Antonio Filieri, Martina Maggio, Konstantinos Angelopoulos, Nicolas D'Ippolito, Ilias Gerostathopoulos, Andreas Berndt Hempel, Henry Hoffmann, Pooyan Jamshidi, Evangelia Kalyvianaki, Cristian Klein, Filip Krikava, Sasa Misailovic, Alessandro Vittorio Papadopoulos, Suprio Ray, Amir M. Sharifloo, Stepan Shevtsov, Mateusz Ujma, and Thomas Vogel. Software Engineering meets Control Theory. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS '15*, pages 71–82. IEEE, 2015.

- [K6] Holger Giese, Thomas Vogel, and Sebastian Wätzoldt. Towards Smart Systems of Systems. In Mehdi Dastani and Marjan Sirjani, editors, *Proceedings of the 6th International Conference on Fundamentals of Software Engineering (FSEN '15)*, volume 9392 of *Lecture Notes in Computer Science (LNCS)*, pages 1–29. Springer, 2015. (invited paper).
- [K7] Joachim Hänsel, Thomas Vogel, and Holger Giese. A Testing Scheme for Self-Adaptive Software Systems with Architectural Runtime Models. In *Proceedings of the 2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW '15*, pages 134–139. IEEE, 2015.
- [K8] Christian Krause, Matthias Tichy, and Holger Giese. Implementing Graph Transformations in the Bulk Synchronous Parallel Model. In *Software Engineering & Management 2015, Multikonferenz der GI-Fachbereiche Softwaretechnik (SWT) und Wirtschaftsinformatik (WI), FA WI-MAW, 17. März - 20. März 2015, Dresden, Germany*, pages 99–100, 2015.
- [K9] Sebastian Wätzoldt and Holger Giese. Modeling Collaborations in Adaptive Systems of Systems. In *Proceedings of the European Conference on Software Architecture Workshops, ECSAW*. ACM, 2015.

#### 7.4 Technische Berichte

- [TR1] Thomas Beyhl and Holger Giese. Efficient and Scalable Graph View Maintenance for Deductive Graph Databases based on Generalized Discrimination Networks. Technical report, Hasso Plattner Institute at the University of Potsdam, 2015. Technical Report No. 99.
- [TR2] Johannes Dyck and Holger Giese. Inductive Invariant Checking with Partial Negative Application Conditions. Technical Report 98, Hasso Plattner Institute, University of Potsdam, Potsdam, Germany, 2015.
- [TR3] Sebastian Wätzoldt and Holger Giese. Modeling Collaborations in Self-Adaptive Systems of Systems—Terms, Characteristics, Requirements, and Scenarios. Technical Report 96, Hasso Plattner Institute at the University of Potsdam, April 2015.

#### 7.5 Miscellaneous

- [M1] Sona Ghahremani. Utility-Driven Modularized MAPE-K Loop Architectures for Self-Adaptive Systems. In *Proceedings of the Joint Workshop of the German Research Training Groups in Computer Science*, 2015, June 2015. Extended Abstract.

## 8 Vorträge

### 8.1 Eingeladene Vorträge

#### Prof. Dr. Holger Giese

April 2015 *Towards Smart Systems of Systems*. 6th International Conference on Fundamentals of Software Engineering (FSEN 2015), Tehran, Iran, April 24, 2015.

May 2015 *Model-Driven Engineering for Cyber-Physical Systems*. Kolloquium, Department Elektrotechnik und Informatik, Universität Siegen, Siegen, Germany, May 8, 2015.

#### Thomas Vogel

August 2015 *Model-Driven Engineering of Self-Adaptive Software*. CS Colloquium at the University of Cape Town, Cape Town, South Africa, August 19, 2015.

### 8.2 Vorträge auf Konferenzen und Workshops

#### Prof. Dr. Holger Giese

January 2015 *Software Engineering for Self-Adaptive Systems & Self-Aware Computing*. Dagstuhl-Seminar 15041 on Model-driven Algorithms and Architectures for Self-Aware Computing Systems, Schloss Dagstuhl, Wadern, Germany, January 20, 2015.

November 2015 *Invariant Checking for Graph Transformation: Applications & Open Challenges*. Dagstuhl Seminar 15451 on Verification of Evolving Graph Structures, Schloss Dagstuhl, Wadern, Germany, November 5, 2015.

#### Dr. Leen Lambers

November 2015 *Invariant Checking for Graph Transformation: Applications & Open Challenges*. Dagstuhl Seminar 15451 on Verification of Evolving Graph Structures, Schloss Dagstuhl, Wadern, Germany, November 5, 2015.

#### Thomas Beyhl

March 2015 *The Design Thinking Methodology at Work*. 14th HPDTRP Community Building Workshop, Stanford, USA, March 12, 2015.

May 2015 *Traceability Recovery for Innovation Processes*. 8th International Symposium on Software and Systems Traceability (SSN 2015), Florenze, Italy, May 17, 2015.

September 2015 *The Design Thinking Methodology at Work*. 15th HPDTRP Community Building Workshop, Potsdam, Germany, September 9, 2015.

**Johannes Dyck**

July 2015 *Inductive Invariant Checking with Partial Negative Application Conditions*. 8th International Conference on Graph Transformation (ICGT 2015), L'Aquila, Italy, July 23, 2015.

September 2015 *Towards the Automatic Verification of Behavior Preservation at the Transformation Level for Operational Model Transformations*. 4th Workshop on the Analysis of Model Transformations (AMT 2015), Ottawa, Canada, September 28, 2015.

**Joachim Hänsel**

September 2015 *A Testing Scheme for Self-Adaptive Software Systems with Architectural Runtime Models*. 2nd Workshop on Quality Assurance for Self-Adaptive, Self-Organising Systems (QA4SASO 2015), Cambridge, USA, September 21, 2015.



## 9 Web-Portale und -Services

### 9.1 Self-adaptive.org

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

### 9.2 MDELab.org

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

### 9.3 CPSLab.org

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

## 10 Mitgliedschaften, Programmkomitees und Gutachtertätigkeiten

### 10.1 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)

#### 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)

### 10.2 Mitarbeit in Programmkomitees

#### Prof. Dr. Holger Giese

- Graphs as Models (GAM 2015)  
London, UK, April 11-12, 2015, [↗ website](#)
- 10th Dependable and Adaptive Distributed Systems Track of the 30th ACM Symposium on Applied Computing (DADS - SAC 2015)  
Salamanca, Spain, April 13-17, 2015, [↗ website](#)
- 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2016)  
Austin, USA, May 16-17, 2015, [↗ website](#)

- 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems (SE-sCPS 2015)  
Florenze, Italy, May 16-24, 2015, [↗ website](#)
- 3rd International Workshop on Software Engineering for Systems-of-Systems (SESoS 2015)  
Florenze, Italy, May 17, 2015, [↗ website](#)
- International Workshop on Petri Nets and Software Engineering (PNSE 2015)  
Brussels, Belgium, June 22-23, 2015, [↗ website](#)
- 8th International Conference on Model Transformation (ICMT 2015)  
L'Aquila, Italy, July 20-21, 2015, [↗ website](#)
- 8th International Conference on Graph Transformation (ICGT 2015)  
L'Aquila, Italy, July 21-23, 2015, [↗ website](#)
- 4th International Workshop on the Verification of Model Transformation (VOLT 2015)  
L'Aquila, Italy, July 23, 2015, [↗ website](#)
- 4th International Workshop on Bidirectional Transformations (Bx 2015)  
L'Aquila, Italy, July 24, 2015, [↗ website](#)
- 7th International Workshop on Software Engineering for Resilient Systems (SERENE 2015)  
Paris, France, September 7-8, 2015, [↗ website](#)
- 2nd Workshop on Quality Assurance of Self-Adaptive, Self-Organising Systems (QA4SASO 2015)  
Cambridge, USA, September 21, 2015, [↗ website](#)
- International Workshop on Modeling in Automotive Software Engineering (MASE 2015)  
Ottawa, Canada, September 27, 2015, [↗ website](#)
- 4th Workshop on the Analysis of Model Transformation (AMT 2015)  
Ottawa, Canada, September 28, 2015, [↗ website](#)
- 10th International Workshop on Models@run.time (MRT 2015)  
Ottawa, Canada, September 29, 2015, [↗ website](#)
- 9th International Workshop on Multi-Paradigm Modeling (MPM 2015)  
Ottawa, Canada, September 29, 2015, [↗ website](#)
- 2nd International Workshop on Event Modeling and Processing in Business Process Management (EMoV 2015)  
Stockholm, Sweden, October 19-22, 2015, [↗ website](#)
- 8th International Conference on Software Language Engineering (SLE 2015)  
Pittsburg, USA, October 26-27, 2015, [↗ website](#)
- 2nd World Forum on Internet of Things (WF-IOT 2015)  
Milan, Italy, December 14-16, 2015, [↗ website](#)

#### **Dr. Leen Lambers**

- 8th International Conference on Graph Transformation (ICGT 2015)  
L'Aquila, Italy, July 21-23, 2015, [↗ website](#)
- 4th International Workshop on the Verification of Model Transformation (VOLT 2015)  
L'Aquila, Italy, July 23, 2015, [↗ website](#)

- 12th Workshop on Model-Driven Engineering, Verification and Validation (ModeVVA 2015)  
Ottawa, Canada, September 29, 2015, [↗ website](#)

### **Thomas Vogel**

- Artifact Evaluation Committee of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)  
Firenze, Italy, May 18–19, 2015, [↗ website](#)
- 1st Workshop on Control Theory for Software Engineering (CTSE 2015), co-located with ESEC/FSE 2015.  
Bergamo, Italy, August 31, 2015, [↗ website](#)
- 10th International Workshop on Models@run.time (MRT 2015)  
Ottawa, Canada, September 29, 2015, [↗ website](#)

## **10.3 Organisation von Tagungen und Workshops**

### **Prof. Dr. Holger Giese**

- 11th Dagstuhl Workshop: Model-Based Development of Embedded Systems (MBEES)  
Schloss Dagstuhl, Wadern, Germany, March 22-25, 2015, [↗ website](#)

## **10.4 Gutachtertätigkeiten**

### **10.4.1 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

### **10.4.2 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)
- The Computer Journal, Oxford University Press

**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)