Jahresbericht 2013

Prof. Dr. Holger Giese
Fachgebiet Systemanalyse und Modellierung

Hasso-Plattner-Institut für
Softwaresystemtechnik
an der Universität Potsdam

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Jahresbericht 2013

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

Leiter des Fachgebiets
Prof. Dr. Holger Giese

Sekretariat
Kerstin Miers

Gastprofessor
Prof. Dr. Jürgen Dingel

Postdocs
Dr. Christian Krause
Dr. Leen Lambers

Wissenschaftliche Mitarbeiter
Dipl.-Wirtsch.Inf. Thomas Vogel

PhD-Stipendiaten
Dipl.-Inform. Basil Becker
Thomas Beyhl, M.Sc.
Johannes Dyck, M.Sc.
Gregor Gabrysiak, M.Sc.
Regina Hebig, M.Sc.
Stephan Hildebrandt, M.Sc.
Dipl.-Inform. Stefan Neumann
Sebastian Wätzoldt, M.Sc.

Studentische Hilfskräfte

Falk Benke
Stefanie Birth
Josefine Harzmann
Peter Haucke
Manuel Hegner
Jens Hildebrandt
Sven Köhler
Christoph Kühnl
Stefan Lehmann
Alexander Lüders
Christoph Matthies
Björn Müller
Edgar Näther
Helena Niesche
Dominic Petrick
Lukas Pirl
Stephanie Platz
Cathleen Ramson
Julian Risch
Ingo Richter
Christoph Sterz
Henrik Steudel
Martin Zabel
Dmitry Zakharov
Christian Zöllner

2 Lehrveranstaltungen

2.1 Vorlesungen

Sommersemester 2013

- Modellierung II
- Software Adaption

Wintersemester 2013/2014

- Modellierung I
- Software Engineering for Embedded Systems

2.2 Übungen/Projekte

Sommersemester 2013

- Modellierung II
- Software Adaption

Wintersemester 2013/2014

- Modellierung I
- Software Engineering for Embedded Systems

2.3 Seminare

Sommersemester 2013

- Advanced MDE: Model Management
- Automated Analysis of Formal Models

Wintersemester 2013/2014

- Automated Analysis of Formal Models

3 Betreuung von Studierenden und Dissertationen

3.1 Betreuung von Bachelorprojekten

3.1.1 Bachelorprojekte (abgeschlossen in 2013)

- *From Creative Ideas to Well-Founded Engineering*
  Betreuer: Prof. Dr. Holger Giese, Thomas Beyhl, Gregor Gabrysiak
  Studenten: Anita Dieckhoff, Tom Bocklisch, Dominic Braeunlein, Tom Herold, Norman Rzepka, Thomas Werkmeister
  Abstract: Softwareentwickler sind nicht dafür bekannt ihre Arbeitsergebnisse gerne zu dokumentieren. Bei Design Thinkern, die eine innovative Idee für neuartige Softwareprodukte mit Hilfe der Design Thinking Methode entwickelt haben, ist das ähnlich. Das Dokumentieren von Ideen, Anforderungen, Alternativen und Lösungen kostet Design Thinker Zeit, die sie selbst lieber in kreative Lösungen einfließen lassen. Jedoch ist gerade die Dokumentation der Design Thinking Ergebnisse als auch der Weg zur Lösung essentiell. Nur dadurch erfahren Softwareentwickler was sie lösen sollen, wie sie testen können, ob dies gelungen ist und können im Fall von (techn.) Einschränkungen gut-informierte Entscheidungen treffen. Fehlende Dokumentation hingegen führt zum Verlust wertvoller Informationen bei der Übergabe von innovativen Ideen der Design Thinker an die Softwareentwickler. Zum Zeitpunkt der Übergabe treffen die informale Welt des Design Thinking und die formale Welt der Softwareentwicklung aufeinander. Diese Lücke gilt es zu verkleinern. Hinzu kommt, dass es oft besser ist Dinge sofort zu dokumentieren als erst Tage oder gar Wochen später. Daher muss eine Möglichkeit zur Dokumentation von Design Thinking Ergebnissen geschaffen werden, die den Design Thinking Prozess möglichst wenig beeinträchtigt und einen Mehrwert für Design Thinker selbst schafft, sodass sie gerne dokumentieren.

  Design Thinker präsentieren oft Konzepte und Features in unterschiedlichsten Formen von Prototypen je nach dem was der Kunde wünscht. Die Adressaten sind dabei nicht die Ingenieure, die die Konzepte und Features umsetzen, sondern der Auftraggeber selbst. Die dahinterstehenden Anforderungen können durch fehlende Dokumentation verloren gehen. Diese Beobachtung ist Gegenstand unseres Forschungsprojekts Connecting Designing and Engineering Activities, das im Rahmen des Hasso Plattner Design Thinking Research Programms stattfindet. Unser Bachelorprojekt bettet sich in dieses Forschungsprojekt ein. Das Bachelorprojekt nimmt sich der Herausforderung an, Design Thinker zu motivieren und vor Augen zu führen warum die Dokumentation ihrer Ideen essentiell für die spätere Umsetzung und Markteinführung ist, sowie die Dokumentation zu erleichtern. Idealerweise entsteht die Dokumentation durch die Design Thinker als Nebenprodukt ihrer täglichen Arbeit und behindert die kreative Methodik des Design Thinking nicht. Durch Beobachtungen und Befragungen soll ermittelt werden 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 Projekteilnehmer erstellt und evaluiert wird.

  Erste Beobachtungen und Interviews mit, sowie Erfahrungen der HPI School of Design

Thinking haben gezeigt, dass eine Art digitale Kommunikationsplattform, wie z.B. Social Networks, sowie dazugehörige mobile Apps eine mögliche Lösung der beschriebenen Herausforderung darstellen. Mit einer solchen Plattform wären Design Thinker in der Lage, entsprechende Dokumentation als Nebenprodukt täglicher Kommunikation zu sammeln und zusätzlich mit Meta-Informationen anzureichern. Durch zusätzliche Verarbeitungsschritte könnten die erfassten Daten verwendet werden, um Dinge wie Empfehlungen für Beiträge, Aussagen über die Vollständigkeit der Dokumentation sowie die Generierung einer finalen Dokumentation zu ermöglichen. Die Erstellung oder Adaption einer solchen Kommunikationsplattform steht entsprechend der durch die Projektteilnehmer ermittelten Anforderungen im Fokus des Projekts.

3.1.2 Laufende Bachelorprojekte (Abschluss 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 maschinen-lesbar 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 widerauffindbar 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 Dokumentati-

on 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 Projekteilnehmer erstellt und evaluiert wird.

3.2 Betreuung von Bachelorarbeiten

[BA1] Tom Bocklisch. Eine Architektur für ein ereignisgesteuertes, webbasiertes Backend für Project-Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[BA2] Dominic Bräunlein. Generierung und Bereitstellung von semantischen Thumbnails aus heterogenen Daten für Project-Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[BA3] Anita Dieckhof. Layout-Funktionalität für interaktive Graphen für Project Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[BA4] Tom Herold. Kontextsensitiver Assistent für interaktive Graphen für Project Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[BA5] Norman Rzepka. Entwicklung einer web-basierten Ereignis-gesteuerten Clientanwendungs-Architektur für Project Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[BA6] Thomas Werkmeister. Erweiterbare Backend-Architektur zur Datensammlung aus heterogenen Quellen für Project-Zoom. Bachelor's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

3.3 Betreuung von Masterprojekten

3.3.1 Masterprojekte (abgeschlossen in 2013)

- *Iterative Development and Execution of Consistency-Preserving Rule-Based Refactorings*
  Betreuer: Prof. Dr. Holger Giese, Dr. Leen Lambers, Basil Becker, Johannes Dyck, Stephan Hildebrandt
  Studenten: Ekaterina Gavrilova, Josefine Harzmann, Hauke Klement, Michael Wolowyk
  Abstract: Refactorings sind eine weit verbreitete Technik zur Verbesserung der Lesbarkeit, Komplexität, Wartbarkeit oder Erweiterbarkeit eines Systems oder Modells durch strukturelle Änderungen. Dabei sollen diese Änderungen nicht nur das Verhalten, sondern insbesondere auch die Konsistenz des geänderten Systems bewahren. Insbesondere sollte ein Refactoring

beispielsweise ein Softwaresystem nur derartig verändern, dass die Bedingungen der verwendeten Sprache zur Sicherstellung der Wohlgeformtheit auch nach der Ausführung des Refactorings gewahrt bleiben.

Ob ein konkretes Refactoring die Konsistenz bewahrt, kann zur Laufzeit für ein konkretes System festgestellt werden. Allerdings liegt die Entscheidung über den Umgang mit einer Konsistenzverletzung dann beim Benutzer, also Anwender des Refactorings, der diese Entscheidung möglicherweise nicht ohne genaue Kenntnis des Refactorings und dessen interner Funktionsweise treffen kann. Um stattdessen diese Entscheidung auf die Ebene der Refactoring-Entwickler zu verschieben, beschäftigt sich das Projekt mit einer Entwicklungsumgebung für Refactorings, in der Konsistenzverletzungen bereits zur Entwicklungszeit festgestellt und behoben werden können. Umgekehrt kann der Anwender dadurch Konsistenzverletzungen zur Laufzeit ausschließen.

Die im Projekt verwendeten Werkzeuge und Technologien basieren auf der Entwicklungsumgebung Eclipse und dem Eclipse Modeling Framework (EMF). Dabei werden die Bedingungen an die Wohlgeformtheit eines Systems mit Hilfe von EMF als Modelle auf der Basis von Metamodellen der jeweiligen Sprache dargestellt. Um Konsistenzbewahrung formal zu verifizieren, werden die Konzepte der Graphbedingungen und Graphregeln auf besagte Modelle beziehungsweise auf die Refactorings angewendet. Ein Werkzeug zur Verifikation induktiver Invarianten, das auf Graphbedingungen und Graphregeln basiert, sichert dann die Konsistenzerhaltung zu oder liefert anderenfalls aussagekräftige Gegenbeispiele. Auf Basis dieser Gegenbeispiele ist der Entwickler in der Lage, durch iteratives Vorgehen schließlich ein konsistenzerhaltendes Refactoring zu entwickeln. Für ein solches Refactoring gilt dann, dass es für jedes konsistente Ausgangssystem bei Anwendung stets ein konsistentes Ergebnissystem liefert.

Zur Ausführung der Refactorings wie auch für eine zusätzliche Überprüfung der Konsistenz des Ausgangssystems zur Laufzeit werden Story-Diagramme verwendet. Sowohl für die Prüfung eines Refactorings auf Konsistenzerhaltung als auch für dessen Ausführung sind daher Modelltransformationen erforderlich. Zum einen müssen das Refactoring und die Wohlgeformtheitsbedingungen, die auf dem Metamodell der jeweiligen Sprache basieren, in ein für das Verifikationstool verständliches Format übertragen werden. Zum anderen ist eine Modelltransformation von einem Refactoring und zusätzlichen Konsistenzprüfungen in ein ausführbares Story-Diagramm erforderlich. Als Grundlage der in der Entwicklungsumgebung verwendeten Modelle ist weiterhin Metamodellierung ein zentraler Bestandteil des Projekts.

Abschließend soll eine Evaluierung als Teil des Projekts erfolgen. Insbesondere soll als konkretes Beispiel JaMoPP (Java Model Parser and Printer) verwendet werden, um aus Java-Programmen Modelle zu erzeugen, die auf einem durch JaMoPP bereitgestellten Metamodell für Java basieren. Dadurch können gängige und weit verbreitete Java-Refactorings auf Konsistenzbewahrung überprüft werden. Da der Ansatz allerdings generischer Natur ist, kann die zu entwickelnde Entwicklungsumgebung für alle Sprachen verwendet werden, sofern entsprechende Metamodelle der Sprache auf Basis von EMF zur Verfügung stehen und sich die Refactorings und Bedingungen an die Wohlgeformtheit als Graphregeln und Graphbedingungen darstellen lassen.

3.3.2 Laufende Masterprojekte (Abschluss 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.

3.4 Betreuung von Masterarbeiten

[MA1] Martin Hanysz. Integrating Offline and Online Adaptations of Self-Adaptive Software Systems. Master's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[MA2] Josefine Harzmann. Capturing and Presenting Design Rationales for Prototyping. Master's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

[MA3] Thomas Stening. Implementierung und Evaluation eines Storydiagramminterpreters für eingebettete Systeme. Master's thesis, Hasso-Plattner-Institut für Softwaresystemtechnik, Universität Potsdam, 2013.

3.5 Betreuung von Dissertationen

3.5.1 Abgeschlossene Dissertationen

[D1] Stefan Neumann. *Modular Timing Analysis of Component-Based Real-Time Embedded Systems*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 2013.

[D2] Andreas Seibel. *Traceability and model management with executable and dynamic hierarchical megamodels*. PhD thesis, Hasso-Plattner-Institut fur Softwaresystemtechnik, Universität Potsdam, 2013.

3.5.2 Laufende Dissertationen

Basil Becker: Modellierung und Verifikation selbst-adaptiver service-orientierter Systeme

Thomas Beyhl: Traceability zur Erfassung von Designrationalen und als Hilfsmittel für Versionierung

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

Gregor Gabrysiak: Erhebung und Validierung von Requirements durch Simulation und Animation

Regina Hebig: Megamodeling the Development of Service Oriented Enterprise Systems

Stephan Hildebrandt: Effiziente Ausführung und Fehleranalyse von Modelltransformationen und -synchronisationen

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

Sebastian Wätzoldt: Reconfiguration and Adaptation of Cyber-Physical 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 unterstützten 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, zeitbehaftete 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 Effiziente Wartung von Modellen unter der Verwendung von Megamodellen und fortgeschrittene Anwendungen von Megamodellen

Modellgetriebene Softwareentwicklung leidet unter der stetig steigenden Komplexität von heutigen Softwaresystemen. Wegen diesem Problem ist vor einigen Jahren der Forschungsbereich der Megamodellierung entstanden. Heutige Ansätze zur Megamodellierung sind schon weit fortgeschritten. Dennoch fehlt aktuellem State-of-the-Art noch Unterstützung zur automatisierten und effizienten Wartung von Modellen mittels Megamodellen. Zusätzlich wurde bis jetzt noch nicht alle Möglichkeiten durch die Anwendung von Megamodellen vergegenwärtigt. Diese Forschungsarbeit arbeitet an der Verbesserung der Automatisierung und effizienter Wartung sowie das Finden neuer Anwendungsgebiete für Megamodelle.

Ansprechpartner: Andreas Seibel

4.5 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

4.6 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 validert 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

4.7 Modellierung, Analyse und Bewertung nicht-funktionaler Eigenschaften in komponentenbasierten eingebetteten Realzeitsystemen

Das Forschungsthema beschäftigt sich mit der Analyse von Eigenschaften verteilter, eingebetteter Realzeitsysteme, wie diese beispielsweise im Automobil zum Einsatz kommen.
Dabei wird der Fokus auf die Analyse nicht-funktionaler Eigenschaften gelegt, wie etwa im Fall zeitlicher-Anforderungen bei eingebetteten Systemen. Die steigende Komplexität der Systeme macht es zunehmend schwieriger diese Systeme zu entwickeln, da gerade nicht-funktionale systemweite Eigenschaften mit den bisherigen Verfahren erst sehr spät im Entwicklungsprozess analysiert werden können. Es wird nach einem Ansatz gesucht, welcher es erlaubt, frühzeitig im Entwicklungsprozess Aussagen über entsprechende Eigenschaften des Gesamtsystems zu treffen.

Ansprechpartner: Stefan Neumann

4.8 Rekonfiguration und Adaption von Cyber-Physical Systems

Ein Großteil heutiger Software befindet sich in kleinen eingebetteten Systemen. Diese sind meist vor dem Nutzer verborgen und realisieren Steuerungs-, Überwachungs- oder Regelungsaufgaben. Zusätzlich sind diese Systeme durch eine eingeschränkte Ressourcennutzung und weiterer nicht funktionaler Eigenschaften, wie harte Echtzeit Anforderungen oder begrenzte Speicherkapazität, gekennzeichnet. Cyber-Physical Systems sind eine neue Generation dieser eingebetteten Systeme,

die durch eine immer stärkere Vernetzung und der Interaktion mit der physikalischen Umgebung entstehen.
Dieses Forschungsthema befasst sich mit der dynamischen Anpassung des Verhaltens von eingebetteten und Cyber-Physical Systems. Bestehende Ansätze zur Rekonfiguration oder Adaption des Verhaltens berücksichtigen die geforderten Echtzeiteigenschaften nur ungenügend oder fordern unrealistische Annahmen für die Software.
Eine Evaluierung wird anhand des AUTOSAR Standards vorgenommen, der größtenteils im Automobilbereich eingesetzt wird. Zusätzlich werden die Forschungsergebnisse anhand eines Roboterlabores evaluiert. Dieses besteht aus drei mobilen Robotern und einer variablen Umgebung, die ein Produktionsszenario simuliert.

Ansprechpartner: Sebastian Wätzoldt

4.9 Traceability zur Erfassung von Designrationalen und als Hilfsmittel für Versionierung

Traceability ist Gegenstand aktueller Forschung und ermöglicht durch die Erzeugung von Traceability Informationen u.a. Aussagen über den Einfluss von Artefaktänderungen, Erfüllung von Systemanforderungen, Entwicklungsfortschritt und Designrationalen. Um diese Aussagen zuverlässig treffen zu können müssen die Traceability Informationen komplett und korrekt sein. Ansätze zur Erzeugung dieser Traceability Informationen sind oft technologie- oder sprachen-spezifisch. Darüber hinaus ändern sich Traceability Informationen über die Zeit und sind daher Gegenstand von Wartung und Versionierung.
In diesem Forschungsthema geht es darum wie Traceability-Ansätze kombiniert werden können um deren Vollständigkeit und Korrektheit zu erhöhen, wie die erhaltenen Traceability Informationen versioniert werden können und wie sie im Kontext der Methodik des Design Thinking sowie der Modellgetriebenen Softwareentwicklung genutzt werden können.
Im Rahmen der Methodik des Design Thinking wird evaluiert wie Traceability Informationen und deren Versionierung zur Erfassung und Auffindung von Designrationalen genutzt werden können. Im Kontext der Modellgetriebenen Softwareentwicklung wird evaluiert wie Traceability Informationen kollaborative Entwicklungsszenarien unterstützen können.

Ansprechpartner: Thomas Beyhl

4.10 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 entwickelen, 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 Zuverlässlichkeitseigenschaften. 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 eine Antrag auf Sachbeihilfe bei der Deutschen Forschungsgemeinschaft (DFG) verfasst.

Ansprechpartner: Christian Krause

4.11 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 detaillierte Aussage über die Gültigkeit der zu beweisenden Eigenschaften getroffen werden. Beispielsweise könnte eine Eigenschaft als induktive Invariante zurückgewiesen werden, weil die Eigenschaft nach einem Zustandsübergang aus einem Zustand verletzt wird, der wiederum nur aus einem anderen verbotenen Zustand erreichbar ist. Durch die Untersuchung eines längeren Pfades wird die Zahl der Gegenbeispiele, die auf derartigen nicht korrekt erreichbaren Zuständen basieren, reduziert. Ein weiterer Punkt im Rahmen des Themas ist die Ausdrucksmächtigkeit des Ansatzes und die potentielle Erweiterung derselben.

Ansprechpartner: Johannes Dyck

5 Drittmittelprojekte

5.1 DFG – Korrekte Modelltransormationen (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

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

6 Forschungskooperationen

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

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 Enwicklungsansätzen in der Praxis

Capgemini
Erfassung von modelgetriebenen Enwicklungsansätzen in der Praxis

Ableton AG
Erfassung von modelgetriebenen Enwicklungsansätzen in der Praxis

VCat Consulting GmbH
Erfassung von modelgetriebenen Enwicklungsansätzen in der Praxis

Wasserwacht Berlin vom Deutschen Roten Kreuz
Anforderungsanalyse

7 Publikationen

7.1 Zeitschriftenartikel

[A1] Andrew Fish and Leen Lambers. Special Issue on Graph Transformation and Visual Modeling Techniques: Guest Editors' introduction. *Journal of Visual Languages & Computing*, 24(6):419–420, 2013.

7.2 Beiträge zu Büchern und Sammlungen

[S1] Jesper Andersson, Luciano Baresi, Nelly Bencomo, Rogério de Lemos, Alessandra Gorla, Paola Inverardi, and Thomas Vogel. Software Engineering Processes for Self-Adaptive Systems. In Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*, pages 51–75. Springer, January 2013.

[S2] Thomas Beyhl, Gregor Berg, and Holger Giese. Connecting Designing and Engineering Activities. In Hasso Plattner, Christoph Meinel, and Larry Leifer, editors, *Design Thinking Research - Building Innovation Eco-Systems*. Springer Heidelberg New York Dordrecht London, September 2013.

[S3] Rogério de Lemos, Holger Giese, HausiA. Müller, Mary Shaw, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Gabriel Tamura, NorhaM. Villegas, Thomas Vogel, Danny Weyns, Luciano Baresi, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Ron Desmarais, Schahram Dustdar, Gregor Engels, Kurt Geihs, Karl Goeschka, Alessandra Gorla, Vincenzo Grassi, Paola Inverardi, Gabor Karsai, Jeff Kramer, Antónia Lopes, Jeff Magee, Sam Malek, Serge Mankovskii, Raffaela Mirandola, John Mylopoulos, Oscar Nierstrasz, Mauro Pezzè, Christian Prehofer, Wilhelm Schäfer, Rick Schlichting, Dennis B. Smith, Joao P. Sousa, Ladan Tahvildari, Kenny Wong, and Jochen Wuttke. Software Engineering for Self-Adaptive Systems: A second Research Roadmap. In Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*, pages 1–32. Springer, January 2013.

[S4] Holger Giese and Wilhelm Schäfer. Model-Driven Development of Safe Self-Optimizing Mechatronic Systems with MechatronicUML. In Javier Camara, Rogério de Lemos, Carlo Ghezzi, and Antónia Lopes, editors, *Assurances for Self-Adaptive Systems*, volume 7740 of *Lecture Notes in Computer Science (LNCS)*, pages 152–186. Springer, January 2013.

[S5] Gabriel Tamura, NorhaM. Villegas, HausiA. Müller, JoãoPedro Sousa, Basil Becker, Gabor Karsai, Serge Mankovskii, Mauro Pezzè, Wilhelm Schäfer, Ladan Tahvildari, and Kenny Wong. Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems. In Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*, pages 108–132. Springer, January 2013.

[S6] Danny Weyns, Bradley Schmerl, Vincenzo Grassi, Sam Malek, Raffaela Mirandola, Christian Prehofer, Jochen Wuttke, Jesper Andersson, Holger Giese, and Karl Goeschka. On Patterns for Decentralized Control in Self-Adaptive Systems. In Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*, pages 76–107. Springer, January 2013.

7.3 Begutachtete Konferenz- und Workshopartikel

[K1] Thomas Beyhl, Gregor Berg, and Holger Giese. Towards Documentation Support for Educational Design Thinking Projects. In *Proceedings of EPDE 2013, the 15th International Conference on Engineering and Product Design Education*, pages 408–413, 2013.

[K2] Thomas Beyhl, Gregor Berg, and Holger Giese. Why Innovation Processes Need to Support Traceability. In *International Workshop on Traceability in Emerging Forms of Software Engineering*, pages 1–4, San Francisco, 2013. IEEE.

[K3] Thomas Beyhl, Regina Hebig, and Holger Giese. A Model Management Framework for Maintaining Traceability Links. In Stefan Wagner and Horst Lichter, editors, *Software Engineering 2013 Workshopband*, volume P-215 of *Lecture Notes in Informatics (LNI)*, pages 453–457, Aachen, February 2013. Gesellschaft für Informatik (GI).

[K4] Dave Binkley, Dawn Lawrie, Emily Hill, Janet Burge, Ian Harris, Regina Hebig, Oliver Keszöcze, Karl Reed, and John Slankas. Task Driven Software Summarization. In *ERA Track at 29th IEEE International Conference on Software Maintenance (ICSM)*, Eindhoven, The Netherlands, 2013. accepted.

[K5] Gregor Gabrysiak, Daniel Eichler, Regina Hebig, and Holger Giese. Enabling Domain Experts to Modify Formal Models via a Natural Language Representation Consistently. In *Proc. of the First ICSE 2013 Workshop on Natural Language Analysis in Software Engineering*, NaturaLiSE'13, 25 May 2013.

[K6] Gregor Gabrysiak, Lukas Pirl, Regina Hebig, and Holger Giese. Cooperating with a Non-governmental Organization to Teach Gathering and Implementation of Requirements. In *Proc. of the 26th Conference on Software Engineering Education and Training*, CSEE&T'13, May 2013.

[K7] Tingting Han, Christian Krause, Marta Kwiatkowska, and Holger Giese. Modal Specifications for Probabilistic Timed Systems. In *Proc. of the 11th International Workshop on Quantitative Aspects of Programming Languages and Systems (QAPL 2013)*, Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2013.

[K8] Regina Hebig, Holger Giese, Florian Stallmann, and Andreas Seibel. On the Complex Nature of MDE Evolution. In Ana Moreira and Bernhard Schaetz, editors, *Model Driven Engineering Languages and Systems, 16th International Conference, MODELS 2013*, LNCS, Miami, USA, 20 September - 4 October 2013. Springer.

[K9] Stephan Hildebrandt, Leen Lambers, and Holger Giese. Complete Specification Coverage in Automatically Generated Conformance Test Cases for TGG Implementations. In Keith Duddy and Gerti Kappel, editors, *Theory and Practice of Model Transformations*, volume 7909 of *Lecture Notes in Computer Science*, pages 174–188. Springer Berlin Heidelberg, 2013.

[K10] Stephan Hildebrandt, Leen Lambers, Holger Giese, Jan Rieke, Joel Greenyer, Wilhelm Schäfer, Marius Lauder, Anthony Anjorin, and Andy Schürr. A Survey of Triple Graph Grammar Tools. In *Bidirectional Transformations*, volume 57, pages 1–18. EC-EASST, 2013.

[K11] Christian Krause, Johannes Dyck, and Holger Giese. Metamodel-Specific Coupled Evolution Based on Dynamically Typed Graph Transformations. In Keith Duddy and Gerti Kappel, editors, *Theory and Practice of Model Transformations, 6th International Conference, ICMT 2013, Budapest, Hungary, June 18-19, 2013. Proceedings*, volume 7909 of *Lecture Notes in Computer Science*, pages 76–91. Springer / Heidelberg, 2013.

[K12] Ralf Teusner, Gregor Gabrysiak, Stefan Richter, and Stefan Kleff. Interactive Strategy-Based Validation of Behavioral Models. In *Proc. of the 12th International Workshop on Graph Transformation and Visual Modeling Techniques*, GT-VMT'13, 2013.

7.4 Bücher und Tagungsbände

[B1] Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors. *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*. Springer, January 2013.

[B2] Andrew Fish and Leen Lambers. Special Issue on Graph Transformation and Visual Modeling Techniques: Guest Editors' introduction. *Journal of Visual Languages & Computing*, 24(6):419–420, 2013.

7.5 Technische Berichte

[TR1] Basil Becker and Holger Giese. Modeling and Verifying Dynamic Evolving Service-Oriented Architectures. Technical Report 75, Hasso Plattner Institute at the University of Potsdam, 2013.

[TR2] Stefan Neumann and Holger Giese. Scalable Compatibility for Real-Time Embedded Components using Language-Progressive TIOA. Technical Report 65, Hasso-Plattner Institute, 2013.

[TR3] Thomas Vogel and Holger Giese. Model-Driven Engineering of Adaptation Engines for Self-Adaptive Software: Executable Runtime Megamodels. Technical Report 66, Hasso Plattner Institute at the University of Potsdam, Germany, April 2013.

7.6 Miscellaneous

[M1] Johannes Dyck. Graph Transformation Systems and Invariant Checking with k-Inductive Invariants. In Proceedings of the Joint Workshop of the German Research Training Groups in Computer Science, Dagstuhl 2013, Pro Business, Berlin, 2013. Extended Abstract.

8 Vorträge

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.

Dr. Leen Lambers

March 2013 *Introduction to Graph Transformation and Static Analysis Techniques*. Blackboard seminar, LSI departmen, UPC Barcelona, Spain, March 13, 2013.

May 2013 *Correct Model Transformations*. Formal Methods and Tools, Computer Science Department, University of Twente, The Netherlands, May 16, 2013.

8.2 Vorträge auf Konferenzen und Workshops

Thomas Beyhl

September 2013 *Towards Documentation Support for Educational Design Thinking Projects*. International Conference on Engineering and Product Design Education 2013, Dublin, Irland, September 5, 2013.

Mai 2013 *Why Innovation Processes Need to Support Traceability*. International Workshop on Traceability in Emerging Forms of Software Engineering, San Francisco, USA, Mai 19, 2013.

April 2013 *Connecting Designing and Engineering Activities*. 10. HPDTRP Community Building Workshop, Stanford University, CA, USA, April 20, 2013.

February 2013 *A Model Management Framework for Maintaining Traceability Links* . Software Engineering 2013 Traceability Workshop - Nutzen und Praxis, Aachen, Deutschland, February 26, 2013.

Johannes Dyck

April 2013 *Verification with k-Inductive Invariants*. Spring Workshop and Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, April 12, 2013.

June 2013 *Metamodel-Specific Coupled Evolution Based on Dynamically Typed Graph Transformations*. 6th International Conference on Model Transformations (ICMT), Budapest, Hungary, June 18, 2013.

October 2013 *Improving Graph-Based Invariant Checking with Constraint Reasoning*. Fall Workshop and Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, October 17-18, 2013.

December 2013 *Formal Verification with Inductive Invariants*. HPI Symposium, Walldorf, Germany, December 16, 2013.

Leen Lambers

February 2013 *Automated Reasoning for Graph Properties and its Applications to Graph Database Design and Analysis*. First Workshop on Graph-based Technologies and Applications (Graph-TA), UPC Barcelona, Spain, February 19, 2013.

Thomas Vogel

December 2013 *Modeling Self-Adaptive Software*. Dagstuhl Seminar on Software Engineering for Self-Adaptive Systems: Assurances, Seminar 13511 at Schloss Dagstuhl, Wadern, Germany, December 15-19, 2013.

Sebastian Wätzoldt

April 2013 *Adaptation in Cyber-Physical Systems*. Spring Workshop and Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, April 12, 2013.

April 2013 *Adaptation in Cyber-Physical Systems*. Workshop for ICT4D and Service-Oriented Computing at the University of Cape Town, Cape Town, South Africa, April 15, 2013.

May 2013 *Adaptation in Cyber-Physical Systems*. Annual DFG Research Workshop, Dagstuhl, Germany, May 27, 2013.

October 2013 *Adaptation in Cyber-Physical Systems*. Fall Workshop and Ph.D. Retreat of the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, October 18, 2013.

December 2013 *Collaboration in Cyber-Physical Systems*. Doctoral Seminar at the HPI Research School on Service-Oriented Systems Engineering, Potsdam, Germany, December 04, 2013.

9 Herausgeberschaft

9.1 Bücher und Tagungsbände

[B1] Rogério de Lemos, Holger Giese, HausiA. Müller, and Mary Shaw, editors. *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*. Springer, January 2013.

[B2] Andrew Fish and Leen Lambers. Special Issue on Graph Transformation and Visual Modeling Techniques: Guest Editors' introduction. *Journal of Visual Languages & Computing*, 24(6):419–420, 2013.

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 Werkzeuge 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

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

- 29th IEEE International Conference on Software Maintenance (ICSM)
  Eindhoven, The Netherlands, September 22-28, 2013, ↗ website
- 6th International Conference on Model Transformation (ICMT)
  Budapest, Ungary, June 18-19, 2013, ↗ website

11.3 Organisation von Tagungen und Workshops

Prof. Dr. Holger Giese

- Workshop Co-Chair of the International Conference on Software Engineering (ICSE)
  San Francisco, USA, May 17-27, 2013, ↗ website
- Dagstuhl Seminar on Software Engineering for Self-Adaptive Systems: Assurances
  Schloss Dagstuhl, Wadern, Deutschland, December 15-19, 2013, ↗ website
- 9. Dagstuhl-Workshop: Modellbasierte Entwicklung eingebetteter Systeme (MBEES)
  Schloss Dagstuhl, Wadern, Deutschland, April 24-26, 2013, ↗ website

Dr. Leen Lambers

- 2nd International Workshop on the Verification of Model Transformation (VOLT)
  Budapest, Ungary, June 17, 2013, ↗ website

11.4 Gutachtertätigkeiten

11.4.1 Forschungsprojekte

Prof. Dr. Holger Giese

- European Research Council (ERC)
- European Union Seventh Framework Programme (EU FP7)
- Deutsche Forschungsemeinschaft (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

- Journal of Visual Languages and Computing, Elsevier (Zeitschrift)
- Journal of Software and Systems Modeling, Springer (Zeitschrift)
- Journal of Science of Computer Programming, Elsevier (Zeitschrift)

Thomas Vogel

- 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)