

# Praxis der Forschung – Sommersemester 2017

## Teilnehmende Arbeitsgruppen im Sommersemester 2017

- IAR Asfour, Arbeitsgruppe für Hochperformante Humanoide Technologien (H<sup>2</sup>T)
- IAR Stiefelhagen, Computer Vision for Human-Computer Interaction Lab
- IPD Reussner, Software Design and Quality (SDQ)
- ITI Beckert, Anwendungsorientierte formale Verifikation
- TM Beigl, Pervasive Computing Systems / TECO

## Kontakt bei allgemeinen Fragen zu „Praxis der Forschung“:

- Michael Kirsten (ITI Beckert), [kirsten@kit.edu](mailto:kirsten@kit.edu), +49 721 608-45648
- Sarah Grebing (ITI Beckert), [sarah.grebing@kit.edu](mailto:sarah.grebing@kit.edu), +49 721 608 45253

## Termine:

- Anmeldung bis 03.05.2017 beim jeweiligen Betreuer + per Mail bei Michael Kirsten + im ILIAS Kurs **Praxis der Forschung (Methoden, 1. Semester) SoSe 2017**  
[https://ilias.studium.kit.edu/goto.php?target=crs\\_663734](https://ilias.studium.kit.edu/goto.php?target=crs_663734)  
Bitte Thema und Betreuer, sowie Matrikelnummer bei der Anmeldung angeben.
- **1. Methodische Veranstaltung:**  
„Kickoff für das SoSe 2017 und Literaturrecherche“  
04.05.2017, 11:30 – 13:15 Uhr in Raum 010, Geb. 50.34

## Ausgeschriebene Themen im SoSe 2017

Praxis der Forschung – Sommersemester 2017.....	1
Active Perception: Physically Plausible Scene Understanding.....	1
Assistenzsystem zur Entwicklung humanoider Roboter.....	2
Pixelwise Depth Estimation from Monocular Images.....	2
Inkrementelle Modellsichten.....	2
Kopplung von modularen Simulationen und Analysen.....	3
Security Design Patterns applied to CoCoME.....	3
Formale Verifikation objektorientierter Software für Produktionsanlagen.....	4
Interaktionskonzepte für benutzbare Beweissysteme.....	4
Mirror mirror on the wall, tell me what I'm looking for.....	5
IoT Inertialsensorystem.....	5

## Active Perception: Physically Plausible Scene Understanding

Active perception enables robots to explore partially or completely unknown environments autonomously using physical interaction. The forceful physical interaction with the environment generates new sensory input, which would not have happened without it. By incorporating this new information, the robot is able to improve scene understanding and to build richer object representations. This process relies on visual and haptic sensors to perceive the environment and the objects in it. However, in order to interpret the signals and gain information about the scene, prior knowledge of what object relations constitute a physically plausible description of the environment is still needed. Since the physical interaction with the environment is a central part of active perception, physical plausibility allows to further constrain the usually strongly under constrained problem of scene understanding.

The student working on this project should learn how to integrate physical plausibility into an (active) perception pipeline. By implementing and evaluating a component for scene understanding for the real robot, the student shows the improvements gained compared to standard approaches.

The result of the project should be a system for physically plausible scene understanding, which will be integrated and evaluated on a real robot using our robot software framework ArmarX. Applicants must be interested in active perception and should possess good programming skills in C++.

Kontakt / Betreuer:

Fabian Paus (IAR Asfour)

[fabian.paus@kit.edu](mailto:fabian.paus@kit.edu)

## Assistenzsystem zur Entwicklung humanoider Roboter

Aktuell findet die Entwicklung humanoider Roboter hauptsächlich in Forschungs-einrichtungen statt. Da sich noch keine Referenzplattformen oder feste Standards zur Entwicklung durchgesetzt haben, gibt es zum Bau eines humanoiden Roboters meist nur zwei Möglichkeiten, die auch miteinander kombiniert werden können: Zum einen können Roboter komplett neu entwickelt werden. Dies erfordert nicht nur viele Ressourcen, sondern ist auch sehr herausfordernd, da die Neuentwicklung eines humanoiden Roboters eine komplexe, interdisziplinäre Aufgabe ist. Eine zweite Möglichkeit besteht darin, existierende Roboterkomponenten zu einem neuen System zu verbinden. Doch da die Entwicklung von Robotern eine sehr heterogene Aufgabe ist, ist es schwierig sicherzustellen, dass Roboter aus verschiedensten Einzelkomponenten die Anforderungen, die an sie gestellt werden, auch erfüllen. Eine Möglichkeit um die Entwicklung humanoider Roboter in beiden Fällen zu vereinfachen ist die Entwicklung eines Assistenzsystems.

Ziel der Arbeit ist die Implementierung eines Programms, das basierend auf den Anforderungen des Nutzers die Entwicklung eines humanoiden Roboters unterstützt, indem es eine Kinematik und passende Komponenten empfiehlt. Um dieses Ziel zu erreichen muss zunächst untersucht werden, wie Wissen zu existierenden Robotersystemen sinnvoll kategorisiert und verwendet werden kann. Dies kann durch die Erstellung einer Ontologie erfolgen. In einem zweiten Schritt muss dieses Wissen dann in einem Programm nutzbar gemacht werden, das dem Nutzer Fragen stellt, um die genauen Anforderungen zu ermitteln. Das Programm soll dann eine Kinematik und Komponenten empfehlen. Die Empfehlungen werden idealerweise durch Beispiele existierender Systeme aus der Literatur gestützt.

Voraussetzungen für die Durchführung der Arbeit sind fundierte Kenntnisse in C++ oder einer anderen OO-Sprache (z.B. Java, C#). Des Weiteren sollte Interesse an der Entwicklung von Wissensdatenbanken und humanoiden Robotern bestehen.

Kontakt / Betreuer:

Samuel Rader (IAR Asfour)

[samuel.rader@kit.edu](mailto:samuel.rader@kit.edu)

## Pixelwise Depth Estimation from Monocular Images

Kontakt / Betreuer:

Monica Haurilet (IAR Stiefelhagen)

[haurilet@kit.edu](mailto:haurilet@kit.edu)

## Inkrementelle Modellsichten

Moderne Software-Systeme weisen eine hohe Komplexität und eine umfangreiche Größe auf. Daher werden in der Entwicklung solcher Systeme mehrere Sprachen und Modelle verwendet, um unterschiedliche Gesichtspunkte und Abstraktionsebenen der Systeme zu beschreiben. Beispielsweise können Softwaresysteme unter anderem mit Komponentenmodellen der Architektur, Performance-Modellen zur Analyse der nichtfunktionalen Eigenschaften oder mit der Implementierung im Code dargestellt werden. Obwohl all diesen Artefakten unterschiedliche Konzepte und Formalismen zugrunde liegen, beschreiben sie dasselbe System aus unterschiedlichen Blickwinkeln. Zwischen diesen Modellen kommt es häufig zu semantischen Überlappungen, beispielsweise sind Komponenten aus der Architektur auch in der Implementierung in Form von Klassen wieder

zu finden. Um ein vollständiges Bild einer Komponente zu erfassen, müssen Informationen daher aus verschiedenen Modellen aggregiert werden in sog. Sichten. Die Entwicklung dieser Sichten wird dabei von Sprachen wie beispielsweise dem am Lehrstuhl Reussner entwickelten ModelJoin unterstützt. Ein grundsätzliches Problem hierbei ist es jedoch, dass sich während des Softwareentwicklungsprozesses die zugrundeliegenden Modelle der Architektur, der Performance-Modelle und der Implementierung ändern. Da diese Änderungen potentiell Auswirkungen auf die Sichten haben, müssen diese neu generiert werden. Für große Softwaresysteme ist die ständige Neugenerierung dieser Sichten aber zeit- und rechenintensiv, gerade weil mit jeder Änderung nur kleine Teile des Systems geändert werden. Es ist daher lohnenswert, möglichst große Teile der bestehenden Sicht zu belassen und nur die Teile zu aktualisieren, die tatsächlich von einer Änderung betroffen sind. Man spricht hierbei von einer inkrementellen Ausführung der Sicht.

Im Rahmen der Projektgruppe soll untersucht werden, inwiefern neue Forschungsansätze zu impliziter Inkrementalität dazu geeignet sind, eine solche inkrementelle Ausführung von Sichten automatisch aus der Spezifikation eines Sichttyps abzuleiten. Dazu soll ein System entwickelt werden, dass mit der Sprache ModelJoin spezifizierte flexible Sichten automatisch inkrementelle ausführen kann. Hierbei kann auf das ebenfalls am Lehrstuhl Reussner entwickelte inkrementelle Ausführungssystem NMF Expressions zurückgegriffen werden. Die Korrektheit des Systems soll durch Fallstudien belegt und deren Nützlichkeit durch Messung des Laufzeitverhaltens und des Speicherverbrauchs belegt werden. Studierende werden dabei durch Mitarbeiter des Lehrstuhls Reussner betreut.

Kontakt / Betreuer: Erik Burger (IPD Reussner) [burger@kit.edu](mailto:burger@kit.edu)  
Georg Hinkel (IPD Reussner) [georg.hinkel@kit.edu](mailto:georg.hinkel@kit.edu)

## Kopplung von modularen Simulationen und Analysen

Kontakt / Betreuer: Robert Heinrich (IPD Reussner) [robert.heinrich@kit.edu](mailto:robert.heinrich@kit.edu)

## Security Design Patterns applied to CoCoME

Moderne und komplexe Enterprise-Software-Systeme durchlaufen im Laufe der Zeit viele Änderungen mit unterschiedlichen Folgen für die Sicherheit des Systems. Um diese Folgen schnell und präzise erkennen sowie darauf reagieren zu können, sollte Sicherheit bereits in früheren Entwicklungsphasen und danach kontinuierlich betrachtet werden. Dazu müssen auf Architekturebene Modellierungsmöglichkeiten geschaffen werden, um die sicherheitsrelevante Entwurfsentscheidungen mit dazugehörigen Annahmen an den Systemkontext, die Laufzeitkonfiguration sowie Nutzungsprofile zu dokumentieren. Die als eine logische Verbindung zwischen Bedrohungen und Sicherheitsmuster agierende Annahmen werden eingesetzt, um die Validität der Sicherheitsaspekte zu überprüfen.

Die so entstandenen Sichten und Modelle können anschließend in den verschiedenen Phasen zur Analyse der Sicherheit genutzt werden, u.a. auf sicherheitsrelevante Änderungen während der Software-Evolution. Im Rahmen der Forschungsgruppe soll ermittelt werden, wie formale Dokumentation von Annahmen und deren Analyse bzgl. Sicherheitsaspekte genutzt werden können, um Sicherheitszustand einer Enterprise-Anwendung während der Software-Evolution zu prüfen. Dabei sollen insbesondere folgende Punkte betrachtet werden:

- Untersuchung unterschiedlicher Sicherheitsaspekte für konkrete Anwendungsdomäne- und Szenarien (Webanwendungen, Betriebssysteme, eingebettete Software, etc.)
- Ableitung der möglichen Sicherheitslücken, Bedrohungen sowie dazugehörige Annahmen aus Antipatterns und Misuse-Cases
- Abbildung von Laufzeitkonfiguration und deren Beschreibung während der Entwurfsphase

Der entwickelte Ansatz soll zur Evaluation in der PCM-Workbench umgesetzt werden. Als Fallstudiensystem

wird CoCoMe genutzt. Es handelt sich dabei um eine Cloud-basierte, verteilte Enterprise-Anwendung, für die verschiedene State-of-The-Art Evolutionsszenarien definiert wurden.

Kontakt/Betreuer:

Emre Taşpolatoğlu (IPD Reussner)  
Robert Heinrich (IPD Reussner)

[tasspolat@fzi.de](mailto:tasspolat@fzi.de)  
[heinrich@kit.edu](mailto:heinrich@kit.edu)

## Formale Verifikation objektorientierter Software für Produktionsanlagen

Moderne Produktionsanlagen sind häufig mehrere Jahrzehnte lang im Einsatz und müssen immer wieder gewartet, optimiert oder erweitert werden – auch die Software muss an geänderte Anforderungen angepasst werden. Dabei können sich leicht Fehler in den Code einschleichen. Wir forschen deshalb an Methoden und Techniken zur Regressionsverifikation, bei der überprüft wird, ob eine neue Version der Software sich ähnlich wie die alte verhält. Anders als beim Regressionstesten werden dabei nicht einzelne Läufe betrachtet, sondern durch formale Analyse “alle möglichen Läufe” symbolisch auf einmal. Während Objektorientierung in Bereichen wie Anwendungs- und Server-Software längst nicht mehr wegzudenken ist, beginnt der Aufstieg der Objektorientierung in der hardwarenahen Disziplin der Anlagensoftware gerade erst langsam.

Ihre Aufgabe im Rahmen dieses Praxis-der-Forschung-Projektes wird es sein, einen Ansatz zu erarbeiten, mit dem Regressionsverifikation von objektorientierter Software für speicherprogrammierbare Steuerungen (SPS) möglich ist. Dabei können und sollen Sie auf ein existierendes Rahmenwerk für eine imperative, nicht objektorientierte Sprache aufbauen, und es um die entsprechenden benötigten neuen Konzepte zur Behandlung der Objektorientierung erweitern. Im Zentrum Ihrer Arbeit steht die logische Modellierung der objektorientierten Sprachmerkmale. Die Modellierung muss hinreichend präzise und effizient sein, um Regressionsverifikation zu ermöglichen.

Die Forschung an diesem Projekt ist sehr forschungsrelevant und Sie werden Gelegenheit erhalten, Ihre Ergebnisse auch den Projektpartnern unseres DFG-Projektes von der TU München vorzustellen.

Ressourcen

- Bernhard Beckert, Mattias Ulbrich, Birgit Vogel-Heuser, Alexander Weigl: “Regression Verification for Programmable Logic Controller Software”, 17th International Conference on Formal Engineering Methods, ICFEM 2015, Paris, France, November 3–5, 2015, Proceedings, LNCS 9407, 2015. Seiten 234-251
- Ulf Schünemann: “Programming PLCs with an Object-Oriented Approach”, atp international, Nr. 2.2007 Februar 2007, Seiten 59–63.

Kontakt / Betreuer:

Alexander Weigl (ITI Beckert) [weigl@kit.edu](mailto:weigl@kit.edu)  
Mattias Ulbrich (ITI Beckert) [ulbrich@kit.edu](mailto:ulbrich@kit.edu)

## Interaktionskonzepte für benutzbare Beweissysteme

Interaktives Programmbeweisen ist eine kognitiv anspruchsvolle Aufgabe. Selbst die Unterstützung durch automatisierte Werkzeuge ändert bisher daran nicht sehr viel. Neben dem mathematischen Kernproblem existieren auch Abhängigkeiten zwischen Programmaufrufen, die die Aufgabe komplexer gestalten. Aktuelle Werkzeuge haben verschiedene Interaktionskonzepte zum Beweisen integriert, bspw. Point-und-Click Interaktion oder skriptbasierte Interaktion.

Ziel dieses Projektes ist es, zu erforschen wie verschiedene State-of-the-Art Interaktionskonzepte für das interaktive Beweisen in einer Oberfläche für ein einfaches Programmbeweis-System kombiniert werden können, um die User Experience zu erhöhen. Neben den Interaktionskonzepten ist eine weitere Fragestellung, wie Abhängigkeiten zwischen Beweis- und Programmartefakten geeignet dargestellt werden können, sodass sie den Nutzer während der Bearbeitung der Beweisaufgabe zum richtigen Zeitpunkt zur Verfügung stehen. In diesem Projekt ist der Fokus auf der grafischen Benutzeroberfläche. Es sollen die verschiedenen, bisher vorhandenen Interaktionskonzepte in einer GUI für ein einfache Programmbeweissystem nach dem Stand-der-Kunst der User

Experience kombiniert werden und mittels Nutzertests evaluiert werden, in wie weit sich die Kombination der Konzepte auf die Bearbeitung der Beweisaufgabe auswirkt. Es kann auf einer bestehenden Grundimplementierung eines einfachen Beweissystems aufgebaut werden.

Anforderungen:

- Spaß an der Entwicklung von GUIs und Interaktionskonzepten
- Wissen im Bereich Mensch-Maschine-Interaktion (HCI) und Usability-Evaluierungsmethoden
- Sehr gute Java-Kenntnisse
- Von Vorteil: Vorlesung Formale Systeme

Kontakt / Betreuer:

Sarah Grebing (ITI Beckert) [grebing@ira.uka.de](mailto:grebing@ira.uka.de)  
Andrea Schankin (TM Beigl) [schankin@teco.edu](mailto:schankin@teco.edu)  
Mattias Ulbrich (ITI Beckert) [ulbrich@kit.edu](mailto:ulbrich@kit.edu)

## Mirror mirror on the wall, tell me what I'm looking for

(Ontology-based user modelling, and behavior and personality analysis for predicting users' intentions)

Keywords: Human-Machine-Interaction (HMI), User Modeling, Behavior and Personality Analysis, Personalization, Semantics, Proactivity.

In order to achieve higher levels of performance and quality of service, today's systems and applications interacting with users place great value on some kind of intelligence that makes them among other capable of learning and thereby highly adaptable to their users, their personal needs and their environment. To autonomously acquire and model such context-aware knowledge and reason about it, various artificial intelligent based technologies have been applied, like machine and statistical learning, pattern recognition and semantic-based knowledge graphs to mention a few, with considerably satisfying results as far as human-machine-interaction is concerned.

However, there is still space for improvement. One way to build better systems would be to give them the feature of proactivity on the basis of which such systems would be able to provide users the right service, at the right place and always at the right time even before the user consciously asks for it. To accomplish such a thing and be in other words able to predict the intention of a person, a more sophisticated user model is required. One that takes also the personality of the user in consideration.

Within the scope of Praxis der Forschung, it shall be on the one hand investigated whether an autonomously, un- or semi supervised acquisition, representation and analysis of the user's behavior and personality is possible in the first place. On the other hand, it shall be examined whether and to what degree does this knowledge of the user's personality provide significant support to an intelligent system in predicting his intentions. The Hypothesis shall be tested and evaluated, inter alia by conducting a user study, within a concrete use case.

Requirements:

Applicant should be generally interested in A.I., have good Java and/or Python programming skills and some first experience in programming semantic web applications (RDF, JSON-LD). Additional experience in machine learning and OWL are major plus points.

Language: German or English

Contact / Supervisor:

Antonios (Toni) Karatzoglou (TM Beigl) [antonios@teco.edu](mailto:antonios@teco.edu)

## IoT Inertialsensorsystem

Internet-der-Dinge

1. System: Miniaturisierung, BLE, M4-Prozessor, eingebettete Software
2. Communication: Multi-Hop DTN
3. IoT: Autonome, Self-X Cloud DB (Influx TSDB-basiert)
4. Analytics: Genaue Inertialbewegungsberechnung trotz verrauschter Signale (Autocalibration)

Kontakt / Betreuer:

Michael Beigl (TM Beigl) [michael.beigl@kit.edu](mailto:michael.beigl@kit.edu)