Praxis der Forschung – Wintersemester 2018/19

Teilnehmende Arbeitsgruppen im WiSe 2018/19

- IAR Prof. Asfour, Hochperfomante Humanoide Technologien (H²T)
- ITI Prof. Beckert, Anwendungsorientierte Formale Verifikation
- TM Prof. Beigl, Pervasive Computing Systems (PCS) / TECO
- IAR Prof. Hanebeck, Intelligent Sensor-Actuator-Systems (ISAS)
- IPD Jun.-Prof. Koziolek, Architecture-driven Requirements Engineering (ARE)
- IAR Prof. Kröger, Intelligente Prozessautomation und Robotik (IPR)
- IPD Prof. Reussner, Software Design and Quality (SDQ)
- TM Prof. Zitterbart, Telematik

Kontakt bei allgemeinen Fragen zu „Praxis der Forschung“:

- Michael Kirsten, ITI Prof. Beckert, kirsten@kit.edu, +49 721 608 45648
- Sarah Grebing, ITI Prof. Beckert, sarah.grebing@kit.edu, +49 721 608 45253
- Erik Pescara, TM Prof. Beigl, pescara@teco.edu, +49 721 608 41704

Termine:

- Anmeldung bis 22.10.2018 bei jeweiligen Betreuern + per Mail bei Michael Kirsten + im ILIAS Kurs **Praxis der Forschung (1. Semester) WiSe 2018/19**
  https://ilias.studium.kit.edu/goto.php?target=crs_880067
  Bitte Name, Thema und Matrikelnummer bei der Anmeldung angeben.
- **Erste Methodische Veranstaltung:**
  „Kickoff WiSe 2018/19 und Literaturrecherche“, 25.10.2018, 13:30 - 15:30 Uhr in R. 010, Geb. 50.34

Ausgeschriebene Themen im WiSe 2018/19

Ein Expertensystem zur Entwicklung von Roboterhänden und Prothesen

Die Entwicklung von intelligenten Roboterhänden und Handprothesen nach menschlichem Vorbild stellt Roboteringenieure vor große Herausforderungen. Sowohl die Aktuierung als auch die Sensorisierung von hoch integrierten Roboterhänden muss sorgfältig geplant werden, um passende Komponenten auszuwählen. Das H²T hat bereits verschiedenste Hände für die humanoiden Roboter der ARMAR-Familie und Handprothesen entwickelt. Um das Design zukünftiger Systeme zu unterstützen, soll der Entwicklungsprozess systematisiert und teilweise auch automatisiert werden.
Ziel dieser Projektgruppe ist die Realisierung eines Expertensystems, das Entwickler von Roboterhänden und Handprothesen unterstützt. Basierend auf Anforderungen wie Kraft, taktiler Sensorik, möglichen Griffen und menschlichen Proportionen soll das System passende Lösungen vorschlagen. Als Basis dient ein Expertensystem-Framework, das auf eine ontologische Wissensbasis mit Wissen zu Teilkomponenten und Anforderungen zurückgreift und Lösungen für mechatronische Systeme in einem Bottom-Up-Ansatz generiert. Das Expertenwissen zu Roboterhänden und Handprothesen wird durch Roboterhand-Entwickler zur Verfügung gestellt. Zur Bearbeitung der Arbeit sind gute bis sehr gute Java-Kenntnisse nötig. Geboten werden eine intensive, persönliche Betreuung und die Möglichkeit zur Mitarbeit an aktuellen Forschungsgebieten. Darüber hinaus bietet dieses hoch interdisziplinäre Thema einen Einblick in verschiedene Teilgebiete der Roboterhandentwicklung und Prothetik.

Kontakt / Betreuung: Samuel Rader (IAR Asfour) samuel.rader@kit.edu

Niederdimensionale Repräsentation dynamischer Greifbewegungen

Menschliche Greifbewegungen beruhen auf einer hochkomplexen Ansteuerung der 21 Bewegungsfreiheitsgrade der Hand. Die Generierung solcher Bewegungen für Roboterhände oder Prothesen ist daher herausfordernd und wird häufig über eine niederdimensionale Repräsentation des Griffes gelöst. Dabei werden vorhandene Kopplungen zwischen den menschlichen Gelenken, sogenannte Synergien, genutzt, um den Griff mit einer geringeren Zahl von Parametern zu beschreiben.
Aufgabe dieser Projektgruppe ist es, die Synergien zur Beschreibung einer statischen Griffpose auf den dynamischen Greifvorgang zu erweitern. Dazu soll die Greifbewegung in sinnvolle Abschnitte unterteilt werden, welche im Anschluss auf ihre Charakteristika abhängig von geplanten Grifftyp und dem zu greifenden Objekt untersucht werden. Die Greifbewegung soll im Anschluss mit Hilfe von Methoden des maschinellen Lernens generalisiert und in eine niederdimensionale Darstellung gebracht werden, welche sich für die Steuerung menschenähnlicher Greifbewegungen auf Roboterhänden eignet. Zur Bearbeitung der Arbeit sind gute bis sehr gute Programmierkenntnisse (Python) erforderlich. Grundkenntnisse in Robotik und maschinellem Lernen sind vorteilhaft, aber nicht zwingend erforderlich. Geboten werden eine intensive, persönliche Betreuung und die Möglichkeit zur Mitarbeit an aktuellen Forschungsgebieten.

Kontakt / Betreuung: Julia Starke (IAR Asfour) julia.starke@kit.edu

Geometric and Semantic Scene Reconstruction for SLAM Based on Non-parametric Learning

Simultaneous Localization and Mapping (SLAM) denotes the technique of ego-motion tracking and constructing or updating a map of unknown surroundings at the same time. It plays a central role in a variety of application scenarios, such as autonomous driving, virtual/augmented reality etc. In order to endow better autonomy and interaction capability for mobile robots, techniques such as vision-based semantic understanding are required and have become an active research field in recent years. Though the deep learning-based methods have shown impressive results for image processing such as semantic segmentation, there is still much potential to exploit for making robotic vision more robust, efficient and semantically accurate.

In this project, novel non-parametric learning methods will be investigated and developed for robotic perception in the context of SLAM. In particular, we focus on learning methods proposed on the Riemannian manifolds such that better tracking and mapping performance can be realized both geometrically and semantically. More specifically, the project is composed of the following **work packages:**

- Literature review of existing vision-based semantic SLAM frameworks and non-parametric learning based on Riemannian geometry, especially techniques from geometric deep learning and Gaussian process.
- Development and implementation of a proper vision-based 3D mapping method, based on which a learning framework for semantic understanding is then proposed.
- Development of a SLAM framework and deployment on a real robotic platform mounted with visual sensors, e.g., LiDAR and stereo-camera.
- Evaluation based on real-world dataset and experiments in multiple application scenarios, e.g., 3D object detection, large-scale tracking and semantic reconstruction, etc.

Prerequisites:

- Highly self-motivated and willing to take on challenges.
- Having solid coding skill with C++/Python and good mathematical foundations.
- Experience with deep learning and computer vision is a plus.

At least one joint publication is planned as one of the goals of the project. The topic will be tailored individually in the initial meeting.

Kontakt / Betreuung: Kailai Li (IAR Hanebeck) kailai.li@kit.edu

Kamerabasierte Erfassung des Menschen mittels Deep Learning für sichere Mensch-Roboter-Kooperation

Im Bereich der Robotik kommen Anwendungen, bei denen Mensch und Roboter kooperieren, vermehrt zum Einsatz. Der Gewährleistung der Sicherheit des Menschen wird dabei hohe Bedeutung beigemessen. In diesem Kontext der Sicherheit wird grundsätzlich zwischen Security (Angriffssicherheit) und Safety (Betriebssicherheit) unterschieden. Ziel des vorliegenden Projektes ist die Realisierung des Letzteren unter Einbeziehung von neuronalen Netzen. Zur Abschätzung der Position des Menschen kann beispielsweise ein Kamerasystem verwendet werden.

Aufgabenstellung:

- Recherchen über sichere Mensch-Erfassung in der Mensch-Roboter-Kooperation: eingesetzte Sensortypen (Kameras, Tiefensensoren, Kinect, etc.), deterministische Algorithmen vs. Machine-Learning-Verfahren, relevante Normen und Standards, …
- Untersuchung des DensePose Algorithmus zur Eignung hinsichtlich der Sicherheit in der Mensch-Roboter-Kooperation
- Entwicklung einer systematischen Methode zur Einordnung von Situationen als “kritisch”, abhängig von Körperhaltung des Menschen und Status des Roboters
- Evaluation des angewandten Algorithmus und Methode im Hinblick auf die Performance sowie der Robustheit.

Erwünschte Qualifikationen:

- Grundlegende Kenntnisse über Robotik, Bildverarbeitung oder Machine Learning
- Erfahrung mit Python oder anderen Programmiersprachen

Es *erwarten* Sie spannende und herausfordernde Aufgaben, Einblick in aktuelle Forschungsprojekte im Bereich Safety, ML sowie eine angenehme Arbeitsatmosphäre. Zeitnahe und engagierte Betreuung sorgen für gutes Gelingen.

Kontakt / Betreuung: Dr.-Ing. Christoph Ledermann (IAR Kröger) christoph.ledermann@kit.edu
Woo-Jeong Baek (IAR Kröger) baek@kit.edu
Yongzhou Zhang (IAR Kröger) yongzhou.zhang@kit.edu

Continuous Integration of Performance Models and Frameworks for Data Analysis

Background. Performance models of software systems can be used to anticipate how a system will react to changes in its structure, usage or environment. It is, however, expensive to build accurate prediction models manually. At the Architecture-Driven Requirements Engineering Group (ARE), we are working on an approach for the automated extraction of parameterized models from source code that are updated when the underlying source code or the system's environment changes. This process is then tightly integrated with deployment and operation (DevOps) to parametrize the models based on dynamic analysis and ensure that accurate models are available at all times during the software development process.
Problem. Currently, our approach is focused on extracting performance models from source code and targeted source code instrumentation and monitoring of the system for updating and calibrating those models. This extraction does not differentiate between communication inside the system and the usage of "external" components such as libraries, middleware, frameworks or external services. This can be disadvantageous, e.g., when the monitoring of those services cannot be done by source code instrumentation, because the code is not available. Plus, external components can have specific monitoring interfaces that can be used for monitoring and calibration instead of additional monitoring overhead.
Goals. For the aforementioned reasons, we aim to enrich the model extraction to allow the detection of calls to external components which are then transformed into appropriate representations in the performance model. They need to be handled differently when deciding which parts of the system to instrument or monitor, and how to use the resulting measurements for calibration. Your task is to devise a flexible method for this incorporation and evaluate it using an example system.
Prerequisites.

- Solid Java coding skills and knowledge object-orientation
- Interest in software engineering research and methods and model-driven software engineering
- Additional knowledge in software architecture and software design are helpful, but not mandatory.

Kontakt / Betreuung:	Manar Mazkatli (IPD Koziolek)	manar.mazkatli@kit.edu
	Dominik Werle (IPD Koziolek)	dominik.werle@kit.edu
	Yves Schneider (IPD Koziolek)	yves.schneider@kit.edu

Multi Actor Behaviour and Dataflow Modelling for Dynamic Privacy

(Thema bereits vergeben.)
Motivation. In current architecture descriptions the user behaviour is reduced to the interaction of user groups with the system. This allows getting more easily performance prediction. However in privacy analysis the interaction and dataflow between different users of the system is relevant. Therefore an integration of a multi-actor behaviour modelling might be needed. In Industry 4.0 or IoT environments a high dynamic data-exchange exist. To support this data-exchange, the data-flow of the system needs to be modelled.
Task. First a literature research about different multi-actor behaviour modelling and data-flow modelling tools should be done. Based on the research an existing behaviour model should be adapted to support multi-actors. This should be integrated into an existing data-flow model. In the end a privacy analysis for the new modelling tool should be created.
We offer:

- Latests modelling approaches based on Eclipse EMF
- Strong ties to current research project Trust 4.0
- Good working environment and extensive mentoring

Kontakt / Betreuung:	Dr. Robert Heinrich (IPD Reussner)	robert.heinrich@kit.edu
	Maximilian Walter (IPD Reussner)	maximilian.walter@kit.edu

Bislicing – Slicing for Relational Verification

The problem whether two programs are equivalent is of great interest in the daily practice of software development—especially in order to support evolving software systems. We developed reve, a tool that proves the equivalence of two C programs with the same behaviour on a local function level. This leads to the next challenge: the scalability on full software projects.
Lightweight analysis techniques provide *Program Dependence Graphs* (PDGs) that capture all dependencies between statements within one program. We can use the well-known theoretical result that two equivalent programs have isomorphic PDGs in order to rapidly check whether certain parts of the two analyzed programs are equivalent. This would allow reve to focus on the more difficult program parts. We call this process of excluding equivalent parts (this result is taken from the PDG-analysis) of the two programs for the equivalence verification *bi-slicing*.
The focus of this thesis should be a theoretical concept of bi-slicing for case equivalence checking. Implementation and evaluation are also on the agenda, but subordinate.

Kontakt/Betreuung:	Mihai Herda (ITI Beckert)	herda@kit.edu
	Dr. Mattias Ulbrich (ITI Beckert)	ulbrich@kit.edu

Entwicklung eines formalen Fairnessmodells für Datenverkehr

Spätestens seit der Entwicklung digitaler sozialer Netzwerke durchdringt das Internet in zunehmendem Maße unsere Gesellschaft. Hier stellt sich allerdings spätestens seit der Debatte zum Thema „Netzneutralität“ die Frage, wie dies so umsetzbar ist, dass kein Teilnehmer benachteiligt wird bzw. alle Teilnehmer fair behandelt werden. Allerdings steht hier zuerst die Frage, was der Begriff „fair“ in diesem Kontext überhaupt bedeutet.
Am Lehrstuhl für Angewandte Formale Verifikation existieren hierzu bereits erste Vorarbeiten, in denen wir mittels Begriffen aus der Subdomäne „Fair Allocation Theory“ aus der Sozialwahltheorie Routing formalisiert und Fairness-Begriffe aufgestellt haben. Interessant wäre es hier, dieses Modell im Rahmen einer Projektgruppe weiterzuentwickeln, sodass es sich auf reale Szenarien anwenden lässt und die Fairnessbegriffe hierauf anzuwenden.

Kontakt/Betreuung:	Michael Kirsten (ITI Beckert)	kirsten@kit.edu

Hyper Test Tables

Hyper properties became very popular in the last years, because of their expressive power. The core of hyper properties is the possibility to (uni-versal and existential) quantified over program traces. For example, this enables the specification of refinement (“forall runs in the old software revision, exists a run in the new revision“) or:
“Hyperproperties can express security policies, such as secure information flow and service level agreements, that trace properties cannot.“ (Clarkson and Schneider in Hyperproperties. JCS 18. 2010.)
The goal is a table-based specification language, that (a) supports hyper properties and (b) is decidable by state-of-the-art tools (model checker or SMT solver).
Your task is to understand the current work of generalized test tables, HyperLTL and HyperCTL. You define the syntax and sematic of the specification language and implement a decision procedure for proving the conformance of reactive system to your specification.

Kontakt/Betreuung:	Alexander Weigl (ITI Beckert)	weigl@kit.edu

Program Synthesis from Generalised Test Tables

Background. Automated production systems, such as industrial plants and assembly lines, are usually driven by Programmable Logic Controllers (PLCs). These computing devices are specially tailored to controlling automated production systems in mission- and safety-critical realtime environments. They are worthy goal for formal methods.
Generalised Test Tables are a table-based specification developed at the chair of Prof. Beckert. They arise from the concrete test table and preserve their comprehensibility although extended expressiveness. Program synthesis is the generation of software that adheres a given specification. In contrast to formal verification, where a given software is checked for conformance against a specification.
Goal & Task. In this thesis, we want to develop a practicable method for the synthesis of PLC software from a set of Generalised Test Tables.
Your Profile. Programming skills in Java/Kotlin are required. Furthermore, you should be interested in programming languages, formal methods, and theory of infinite games and automaton. You should have completed the Formal Methods (Formale Systeme) Course at KIT or equivalent.

Kontakt/Betreuung: Alexander Weigl (ITI Beckert) weigl@kit.edu

Property-Directed Reachability for Regression Verification

Since 2007, IC3 and property-directed reachability (PDR) became de-facto standard in the domain of symbolic model checking. Both approaches are decision procedures to verify that a given invariant holds for the modelled system. With Regression Verification, we can prove that two given systems with the same behaviour are functionally equivalent (minus the intended changes). In our case we apply Regression Verification in the field of automated production systems to ensure the well-functioning during software evolution.
The goal of this thesis is the transfer of current PDR/IC3 approaches to software model checking in order to outperform current regression verification implementations. The idea is to exploit the main assumption behind regression verification, which is that both software versions have a high degree of similar structures. As a benchmark scenario we use the Pick-and-Place-Unit from TUM.
Your task is to understand the current State-of-the-Art of PDR and IC3; adapt the ideas into a novel approach, and perform the benchmarks.

Kontakt/Betreuung: Alexander Weigl (ITI Beckert) weigl@kit.edu

Relational Debugging for Scalable Algorithms

In contrast to functional properties, relational properties are universally quantified over multiple program runs. This allows the specification of complex properties. For example: (1) the absence of information flow from confidential input to public output, (2) the equivalence of two programs under the same input, or (3) the numerical stability in scalable algorithms with floating points. If a relational property is violated, the developer needs investigation tools to find the reason. For functional properties, the tools of choice are debuggers, that allow a coarse or fine-grained stepping through a program run and inspection of the internal variable assignments. For relational properties, the developer needs to be able to step through multiple program runs simultaneously.
The goal of this thesis is to develop a full-featured debugger for relational properties applicable for *real* programming languages and *real-sized* software! We need to develop and deploy several features to handle the increased complexity of simultaneously debugging, like relational synchronization points, relational invariants, or user annotations. Your task is to develop a relational debugger. This includes concepts of (1) embedding of user annotation and specification, (2) integration of (semi-)formal methods to aid the user, (3) visualization and user interaction. This project should result into a working prototype.

Kontakt/Betreuung: Alexander Weigl (ITI Beckert) weigl@kit.edu

Entwicklung eines kompakten Piezoaktortreibers

Das TECO entwickelt Anwendungen zu taktiler Informationsübertragung, wobei derzeit Vibrationsmotoren zum Einsatz kommen. Vibrationsmotoren haben allerdings den Nachteil, dass sie eine hohe Baugröße aufweisen und Frequenz und Amplitude nicht unabhängig voneinander verstellbar sind. Ebenso ist die Wellenform (Sinus) technologiebedingt vorgegeben. Für Forschungszwecke sollen Piezoaktoren zum Einsatz kommen, die diese Nachteile nicht aufweisen, in ihrer Ansteuerung aber komplizierter sind.
Ziel des Projekts ist die Entwicklung eines kompakten vom Smartphone aus steuerbaren Piezoaktortreibers. Als Basis dient ein bereits entwickelter Piezoaktortreiber, der aber nur vom Computer aus steuerbar ist und sich in einem 19‘‘ Rack befindet. Im ersten Teil des Projekts soll eine verkleinerte Version des Treibers entwickelt werden. Anschließend wird eine Android App entwickelt, die den Treiber über die Smartphone-Klinkenbuchse ansteuert. Das Projekt kann von 2 Studierenden bearbeitet werden. Zur Bearbeitung sind Kenntnisse in Elektrotechnik (insb. Schaltungsdesign), Android und Java nötig.

Kontakt / Betreuung: Jan Formanek (TM Beigl) formanek@teco.edu

Quellcodeverständnis und API-Usability

Das Entwickeln von Software erfordert das Verstehen von Quellcode, ob beim Entwickeln neuer Software, bei der Wartung bestehender Software oder beim Integrieren neuer Funktionalitäten. Dabei kann das Lesen und Verstehen bereits vorhandenen Codes sowie des Stils und der Intention des ursprünglichen Entwicklers länger dauern als das eigentliche Integrieren der Funktion. Zu erforschen, wie Entwickler Quellcode verstehen, kann zu verbessertem Quellcode führen, der es zukünftigen Entwicklern erleichtert, vorhandene Software zu pflegen und zu erweitern. Das ist insbesondere für die Entwicklung von APIs von Bedeutung. Obwohl es Empfehlungen und Konventionen zum Schreiben von Quellcode gibt (z.B. Java-Konventionen oder Clean Code), liegen bislang nur wenige empirische Befunde vor, die diese Empfehlungen prüfen.
Ziel des „Praxis-der-Forschung-Projekt“ ist daher die empirische Untersuchung, wie Programmierer Quellcode verstehen. Dazu sollen empirische Studien geplant, durchgeführt und ausgewertet werden. Je nach Interesse kann entweder eher auf die Erforschung von kognitiven Prozessen (Aufmerksamkeit oder Gedächtnis) oder auf die Usability von APIs fokussiert werden. An diesem Thema können 1-2 Studierende entweder an Einzelprojekten oder im 2er Team arbeiten.
Voraussetzungen:

- Programmiererfahrung (Java, Python)
- Interesse an der Mensch-Computer-Interaktion (HCI)
- Bereitschaft, Nutzerstudien zu planen, durchzuführen und auszuwerten

Kontakt / Betreuung: Dr. Andrea Schankin (TM Beigl) andrea.schankin@kit.edu

Machine Learning for Active Network Defense

Description. Machine learning – especially deep reinforcement learning – is currently one of the most prominent research areas in computer science. Bleeding edge prototypes continuously claim new records in various fields: CMU’s Libratus beat four of the most professional poker players while Google’s AlphaGo Zero succeeded against the world’s best chess programs (after only four hours of self-training). Countless examples from other domains underline the importance of machine learning, from personal assistants to self-driving cars and smart health care.
Given their adaptability and high processing speed it comes as no surprise that machine learning is an emerging trend in the fast-paced cat-and-mouse game that is computer security – prominently demonstrated by the Mayham AI, which competed against human security experts in the DEFCON 24 capture the flag challenge. While autonomous computer systems have taken first steps in the art of cyber defense, their full integration into the landscape of network security remains a challenge.

Assignment. Current approaches to predict adversarial behavior either focus on single-stage attack scenarios or provide a coarse-grained model to determine the next stage of a multi-stage attack. Our primary goal is to extend existing solutions to anticipate adversarial behavior in a more fine-grained way. For this, we want to determine the precise parameterization of various attacks in ongoing multi-stage attack scenarios in advance (e.g., the next type of attack, its timing, its source and the targeted services). To achieve this, the project will cover the following tasks:

- Data acquisition and analysis of multi-stage attack scenarios
- Application of different machine learning techniques (HMMs, ANNs, Bayesian networks)
- Design and implementation of fine-grained prediction mechanisms
- The details of this topic can be discussed individually. Just contact us at the initial PdF meeting (poster session) or send as a mail (see below).

Prerequisites:

- A background or strong interest in machine learning
- Basic knowledge of communication networks (Telematik, EiR)
- Familiarity with at least one programming language.

Kontakt / Betreuung:	Hauke Heseding (TM Zitterbart)	hauke.heseding@kit.edu
	Robert Bauer (TM Zitterbart)	robert.bauer@kit.edu