Bachelor & Master Projects and Theses
Chair for Software and Systems Engineering

Prof. Dr. Stefan Leue

University of Konstanz
Chair for Software Engineering

Stefan.Leue@uni-konstanz.de
http://www.sen.inf.uni-konstanz.de

Summer Term 2019
We offer

- ... integration in state of the art research projects
- ... if applicable, collaboration with our industrial partners
- ... interesting real-life applications and problems as well as more foundational work
Scope of Projects/Theses

- **Project (Bachelor and Master)**
  - 1 Semester
  - 9 ECTS (270h work)

- **Thesis (Bachelor)**
  - 3-6 Months ("1 Semester")
  - 12 (Thesis) + 3 (Colloquium) ECTS (450h work)

- **Thesis (Master)**
  - 6 Months (1 Semester)
  - 30 (Thesis + Colloquium) ECTS (900h work)
Formal Requirements

- **Before you start your work**
  - Written proposal (≈ 1 page) containing (ideally by **May 3, 2019**)
    - the *topic* you want to choose
    - how well you *match the prerequisites*
    - *why you are suited* for the task.
  - Schedule for the project/thesis
    - What will be achieved at which point in time.

- **During your preparation of the thesis or the project work**
  - Regular consultation with your supervisor.
Deliverables

- Any models/code/data/binaries you created for the project.
- Report of ≥ 20 pages (Projects, Bachelor-Thesis) or ≥ 40 pages (Master-Thesis)
  - Discussing state of the art
  - Stating the problem
  - Presenting your approach and results
  - Critical discussion
Members

- Prof. Dr. Stefan Leue
  - Email: Stefan.Leue@uni.kn.de, Room: PZ 902

- Martin Kölbl
  - Email: Martin.Koelbl@uni.kn.de, Room: PZ 912

- NN

- Dr. Georgiana Caltais
  - Email: Georgiana.Caltais@uni.kn.de, Room: PZ 912
Areas of Research Interest

- **Formal Modeling and Analysis**
  - model checking
  - SMT solving

- **System Safety and Analysis**
  - causality in systems
  - causality checking / QuantUM
  - analysis and repair for timed system models
  - safety and verification of machine-learning based systems

- **Applications**
  - automotive software architectures
  - functional safety of automotive software (ISO 26262)
  - safety of autonomous driving (SOTIF ISO PAS 21448)
  - legal tech
  - collective behavior
Projects and Theses
System Safety and Analysis
Problem Statement

- Causality Checking in QuantUM relies on trace computation
  - bottleneck: memory
- LTSmin / PINS is interface to symbolic model checking engine
  - hope: more memory efficient

Project Tasks

- integrate QuantUM/Causality Checking into LTSmin / PINS
- direct logic encoding of causality checking in symbolic BDD data structures [MP/MT]

Prerequisites

- programming, discrete structures
- for [MP/MT]: (symbolic) model checking an advantage
Causality Checking for Programs [MP/MT]

- Programs
  - the assignment of certain values to variables can cause a program to crash
  - which variable assignments and which values are causal for a program failure?

- Tasks
  - selection of a program analysis framework (other than testing)
    - for instance, symbolic execution, static analysis
  - development of a causality notion for program executions
  - prototype implementation and case study

- Prerequisites
  - good understanding of logic, program semantics, foundations of computing
Formal Modeling and Analysis
Constraint Relaxation for Timed Systems [MP/MT]

- **Motivation**
  - real-time constraints can be unnecessarily tight

- **Analysis to Relax Time Constraints**
  - encode timed system (trace)
    - SMT2, horn clauses
  - encode analysis candidates (e.g. \( x \leq 1 + \Delta d \))
  - check for weaker constraints (e.g. \( \Delta d = 2 \))
    - Craig interpolation, SMT2
  - is the functionality of the system changed?

- **Related Work**
  - analysis of parametric timed systems

- **Requirements**
  - advanced model checking
  - advantageous: course on SMT solving
Repair for Parametric Timed Automata [BP/BT] [MP/MT]

◆ Parametric Timed Automata
  ▶ literature survey
    – starting points
      • Minimal-Time Synthesis for Parametric Timed Automata
        É. André et al., 2019
      • What’s decidable about parametric timed automata?
        É. André, 2015
    – tools? case studies?
  ▶ relationship to repair analysis
    – based on SMT solving
    – tool TarTar @UniKN
    – paper
      • M. Kölbl, S. Leue, T. Wies,
        Clock Bound Repair for Timed Traces,

◆ Prerequisites
  ▶ ideally, Advanced Model Checking, but not mandatory...
From Design to Code
... filling the gap...

SysML State-Machine

 ATL Transformation

How much of the SysML State-Machines can we handle?
To what model-checkers can we translate?
Are the transformations correct?

♦ Co-supervised by Dr. Georgiana Caltais
UML-RT / Papyrus Real Time to Promela / PINS [MP/MT]

- **Model Transformation**
  - UML-RT models edited in Papyrus Real Time (EMF)
  - understand semantics of state machine diagrams, inter-object communication and meta-models
  - define transformation rules in ATL model transformation framework
  - implementation and case studies

- **Benefits**
  - exposure to practically very relevant model based design language
  - interest in Papyrus community

- **Prerequisites**
  - Software Engineering
  - interest in semantics and model transformation
Model Transformation to LTSmin / PINS
- SysML models edited in Papyrus Real Time (EMF)
- understand semantics of state machine diagrams, inter-object communication and meta-models
- define transformation rules in ATL model transformation framework
- implementation and case studies

Benefits
- exposure to practically very relevant model based design language
- interest in Papyrus community

Prerequisites
- Software Engineering
- interest in semantics and model transformation
Run-Time Causality Checking [MP/MT]

♦ **Runtime Verification**
  ‣ observe and assess running system
  ‣ often: monitoring

♦ **Run-Time Causality Checking**
  ‣ observe system behavior
  ‣ detect occurrence of events at run-time as causal for undesired system behavior
  ‣ learn for the future

♦ **Tasks**
  ‣ study various run-time verification approaches, in particular run-time model checking
  ‣ analyze, what causality can mean in this context
  ‣ adapt causality checking

♦ **Preerequisites**
  ‣ one model checking course
  ‣ advantageous: machine learning, data mining
Causality Checking for Deadlock Properties [BP/BT] [MP/MT]

- **Deadlocks**
  - circular wait, no more progress
- **Causality Checking for Deadlocks**
  - deadlock is reachability [BP/BT]
    - adopt causality checking to deadlock
    - implement in SpinJa
  - consider different deadlocks
    - extend the algorithmics and implementation of causality checking to multiple deadlocked states
  - implementation and case studies
- **Prerequisites**
  - preferably, one of the model checking courses
  - good programming skills
Causality Checking Concurrent Code [MP / MT]

Microsoft Research

♦ CHESS
  ‣ tool for finding and producing bugs in concurrent programs (C,C++,C#)

♦ Task
  ‣ transfer the idea of Causality Checking to testing / code analysis
  ‣ apply it to the CHESS tool / environment
  ‣ implement prototype
  ‣ perform case studies
Applications
Joint work with Prof. Rüdiger Wilhelmi, Dept. of Law, Univ. KN

- **Motivation**
  - sale contracts for companies need to be self-consistent
  - vulnerable to inconsistencies because of references and dependencies
  - aim: find inconsistencies automatically

- **Approach (LegalTech)**
  - transform contract to a logical model
  - check for semantic and syntactic inconsistencies using logic reasoning tools

- **Requirements**
  - basic logic
  - good command of German language

---

A. Object of Agreement
Person A sells Person B the company Bakery GmbH.

B Sale Conditions
§1 Purchase Price
Person B pays 50,000€ within 4 weeks after the sale date.

... §5 Warranties
Person A ensures the company has a capacity to bake 10,000 pretzels a day.

§6 Claims
In case the company has not the capacity ensured in §5, Person B has to send a written claim within 2 weeks after the sale date.

§7 Legal Consequences
Person A has to remove any claim within 4 weeks else the price defined in §1 is reduced.
Analysis Methods for SOTIF [MP/MT]

- Safety of the Intended Function
  - ISO PAS 21448 SOTIF
  - how to ensure safety of an autonomous vehicle in the presence of malfunctioning machine learning components

- Objective
  - analysis, where semi-formal and formal modeling and analysis can be used in SOTIF
  - approaches to be considered
    - e.g., probabilistic contracts

- Prerequisites
  - interest in machine learning
  - ideally combined with seminar "Formal Verification and Testing of Deep Neural Networks"

Figure 6: Screen shot of the simulator in interactive mode. See Section 7.1 for explanation of the performance metrics. The green area on the left is unknown because of the viewpoint transformation. The highlighted wide rectangle below the horizon is the area which is sent to the CNN.

Explaining Faults with Machine Learning [MP/MT]

- Explaining Faults with Machine Learning
  - Model Checking: thousands of counterexamples with hundreds of events
  - Use Machine Learning (ML) to learn & explain counterexamples
    - Learn compact representation of counterexamples
    - Extract patterns

- Goals
  - Tool to explain a set of counterexamples of Model Checker (SPIN)
    - Implementation in Python
    - Decision Tree Learning (scikit learn)
  - Case study on selected Promela Models

- Requirements
  - Strong programming skills (e.g. C++, Java, Python)
  - Ability to work independently
  - Basic Machine Learning knowledge
Interested? Contact...

... either one of us any time!

- Prof. Dr. Stefan Leue
  - Email: Stefan.Leue@uni.kn, Room: PZ 902
- Martin Kölbl
  - Email: Martin.Koelbl@uni.kn, Room: PZ 912
Questions