1 Background

Embedded systems are present in nearly every aspects of our daily life — from smartphones to networks of cooperating controller in modern aircrafts. The growing complexity of today’s embedded systems applications results in a dramatic increase of the effort to verify these systems. This applies in particular to the situation where these systems perform safety-related or security-related functions: already now, the verification and validation costs for embedded systems of the highest criticality are higher than the development costs. The V&V costs are expected to increase even further when autonomous robots, vehicles, ships, and trains are deployed more widely.

In computer science and systems engineering, two approaches to overcome this complexity problem are investigated.

- The **correctness by construction** approach applies HW/SW co-design with the objective to create embedded systems that are *a priori* free of errors, by proving each development step — from specifications to implementations — to be a correct refinement of the previous step.

- The **automated verification** approach aims at *a posteriori* verification with a high degree of automation. For both testing and analyses, methods are applied that are capable of uncovering every potential error in designs, software
code, and in the integrated HW/SW system, as long as certain hypotheses are fulfilled.

2 Project TEAMOD

2.1 General Project Objectives

In this project, the automated verification approach is studied, because it is closer to industrial applicability than the former, while still presenting enough scientific challenges for the next 20 years. The methods to be investigated are

- bounded model checking and
- model-based testing.

Bounded model checking investigates correctness properties in the vicinity of a given model or software program state, exploring only a bounded number of execution steps. This has the advantage, that the method can always be applied for bug finding, even when the model complexity would make it infeasible to represent all model states in memory or to perform a comprehensive mathematical verification. Combining bounded model checking with inductive reasoning, it is again possible to obtain globally valid correctness results.

In model-based testing, test cases and concrete test data are generated automatically from models specifying the expected behaviour of the system under test. For these test data, executable test procedures can be generated and executed in an automated way as well. This shifts the test engineers’ focus from test data calculation and test procedure programming to modelling and “thinking” about what the system under test should really do.

In recent years, several researchers have pointed out that the underlying methods for bounded model checking and model-based testing have many similarities. Therefore the combined investigation of these approaches is a very active research field. Also from the practical application perspective, it is advantageous to combine bounded model checking and model-based testing in a single tool: If model-based testing techniques help to create and execute tests based on a model in a fully automated way, it is mandatory to verify that this model is correct and complete. For this purpose, bounded model checking can be used in an efficient way.

It should be emphasised that the project focus is on embedded systems verification. Therefore the methods, techniques and tools studied and developed in this project are not typical software testing methods you might know from Java programming and application of the JUnit tool. Instead, our typical systems under test are integrated HW/SW systems developed to monitor or control components
like interlocking systems or other safety-related applications. Software is programmed in C or C++. Only for the master sub-project “Graphical Interfaces for Model-based Testing” described below, programming is performed in Java within the Eclipse plugin framework.

The project consists of two parts, each part with the usual 2-semester duration.

- The **bachelor project** starts in winter semester 2018/19.
- The **master project** starts in winter semester 2019/20.

For the master project, new students may join the project who did not participate in the bachelor project phase. It is not mandatory to participate in both project phases.

### 2.2 Bachelor Project

The bachelor project is structured into three sub-projects:

- **Methods.** This sub-project aims at acquiring the necessary skills for model checking and model-based testing of embedded control systems. All participants are expected to contribute by giving presentations related to the underlying methods and the practical algorithms used in this field.

- **System Under Test Development.** As system under test, a small railway control system will be developed, running on a platform like Raspberry Pi, using the Linux operating system. The interlocking system is then used to control a Märklin railway system equipped with trains running on different — sometimes conflicting — routes. The control system consists of two parts:
  
  1. **Interlocking System (Route Controllers).** The interlocking system is responsible for reserving routes for trains, setting the appropriate points and signals, and ensuring that no conflicting routes that might lead to train collisions are allocated at the same time.
  2. **Ceiling Speed Monitor (Automated Train Protection).** This component is part of the train’s on-board software. It compares actual train speed to the maximal speed allowed at the current location and issues warnings or performs automated emergency brake intervention, if the speed limit is violated.

- **Modelling, Bounded Model Checking, and Model-based Testing.** A model of the expected interlocking system behaviour is developed. To this end, the SysML modelling language is used. This model is verified using bounded model checking. A hardware-in-the-loop testing environment
is created for testing the interlocking system controller using the HW/SW integration tests automatically create from the model.

Participants are expected to contribute to all three sub-projects. It is mandatory to participate in the TEAMOD meetings every Friday (4 hours), contribute to the preparation of the project day demonstrations, and to participate in the project day.

2.3 Master Project

The master project is structured into the following sub-projects.

- **Algorithms.** New algorithms for bounded model checking and model-based testing are developed, and existing ones are improved. The underlying theory is studied, and the implemented algorithms are integrated into the verification and testing environment. This sub-project does not require any knowledge from the bachelor project, it is suitable for new members joining the master project. The sub-project is also suitable for participants with a special interest in theoretical computer science.

- **Autonomous Train Movements.** The control system is extended to allow for fully autonomous trains movements, such that the presence of train engine drivers is no longer required.

- **Safety Monitor Development.** A safety monitor is developed which checks the interlocking controller during real operation and enforces a safe state (all signals on HALT) if an unsafe controller command is detected. To this end, a code generator is developed which creates the monitor code automatically from safety properties (these are specific formulas described by temporal logic). The underlying theory of so-called property checking and property testing is studied.

- **Graphical Interfaces for Scenario-based Testing.** The efficient derivation of tests from models needs to be supported by intelligent graphical user interfaces. In particular, this applies to so-called scenario-based testing, where automated test generation is guided by expert knowledge. In this sub-project, new graphical interaction paradigms are developed for combining this expert knowledge with the existing test model specifying the expected behaviour of the system under test. An existing tool interface programmed as Eclipse plugin is extended to demonstrate the effectiveness of these new interface paradigms.
Participants are expected to contribute to one of the sub-projects. It is mandatory to participate in the TEAMOD meetings every Friday (4 hours), contribute to the preparation of the project day demonstrations, and to participate in the project day.

2.4 Project Language

Depending on the project participants’ preferences, the project language will be English or German.

3 Project Infrastructure

A project room is available. PC servers and an experimental platform for programming embedded controllers (Raspberry Pi or similar technology) acting as systems under test for the purpose of hardware-in-the-loop testing are available. Modelling tools for model-based testing, as well as a model-based test automation and model checking system that can be extended by novel algorithms during the project are available. A demonstration platform for railway control systems is available.

The project will be supervised by Jan Peleska and Wen-ling Huang.

4 Recommended Lectures

1. Summer semester 2018 – optional – in preparation for the project

   - Test Automation (ME-706.04) – 2+2 hours lecture and tutorial
   - Theory of reactive systems (MB-699.03) – 2+2 hours lecture and tutorial [introduces into the foundations of model checking]
   - Real-time Operating Systems Development (03-ME-702.04) – 2+2 hours lecture and tutorial, introducing the practical development of real-time operating systems

   This is recommended for participants who wish to prepare for the project already during the summer semester.

2. Winter semester 2018/2019

   - Test Automation (ME-706.04) – 2+2 hours lecture and tutorial [highly recommended for those participants who did not take this lecture in the summer semester]
• Systems of high quality, safety, and security (MB-700.31) – 2+2 hours lecture and tutorial
• Operating systems (BB-702.01) – 2+2 hours lecture and tutorial

3. **Summer semester 2019**

• Theory of reactive systems (MB-699.03) – 2+2 hours lecture and tutorial
• Real-time Operating Systems Development (03-ME-702.04) – 2+2 hours lecture and tutorial, introducing the practical development of real-time operating systems [recommended as preparation for the master project]

4. **Winter semester 2019/2020** – start of master project

• Test Automation (ME-706.04) – 2+2 hours lecture and tutorial [highly recommended for those participants who entered the master project but did not take part in the bachelor project]
• Specification of Embedded Systems (ME-702.03) – 2+2 hours lecture and tutorial [this is highly recommended for those participants who wish to specialise on the development of new test modelling formalisms]

5 **Bachelor and Master Theses and More**

The topics covered in this project present a wide range for bachelor and master theses, from simple to sophisticated. The fields of model-based testing and model checking, while being well-understood for test objectives of medium complexity, offer a wide variety of research challenges, giving the opportunity to work on exciting doctoral and habilitation theses.