

# Performance Estimation of the Speed and Distance Monitoring

Alexander Nitsch      Benjamin Beichler

December 20, 2013

## Abstract

This document reports the verification activities of the University of Rostock. SystemC (performance analysis), executable models: SysML modelling and code generation, Scade modeling and code generation.

### 0.0.1 Verification of the Speed and Distance Monitoring

This section reports the verification activities of the *Speed and Distance Monitoring* with model based simulation and virtual prototyping. The first activity pursues the goal of formalizing the specification in the form of an executable model. This model provides a performance estimation at an early stage of system level design and adduces evidence what hardware resources will be needed for the future OBU. Secondly, finding and reporting of unclarities, inconsistencies, incompleteness and errors while implementing the specification by using tools of the openETCS toolchain (Papyrus/SysML, SCADE). Furthermore, we are developing a method of SystemC code generation from abstract and domain specific SysML models. Finally we want to demonstrate the efficiency of model based simulation after transformation from SysML to SystemC models.

The activity is described in the Verification and Validation Plan section *5.3.10 Verification with Model-Based Simulation* [?] To sum up, we develop application models from the specification of the *Speed and Distance Monitoring*, generate test scenarios and use the inherent simulation environment of SystemC to do performance and scheduling analysis.

**Object of verification** Speed and Distance Monitoring ERTMS function baseline 3. The system under test is the implemented SystemC code which is described here: [https://github.com/openETCS/model-evaluation/tree/master/model/SystemC\\_TWT\\_UR0/3.13\\_Speed\\_and\\_distance\\_monitoring](https://github.com/openETCS/model-evaluation/tree/master/model/SystemC_TWT_UR0/3.13_Speed_and_distance_monitoring). It describes the Speed and Distance Monitoring at the Software phase.

**Available specification** The specification is described in the SUBSET-026[?] chap 3.13 [?]. It describes the realization of both TI and DMI commands by

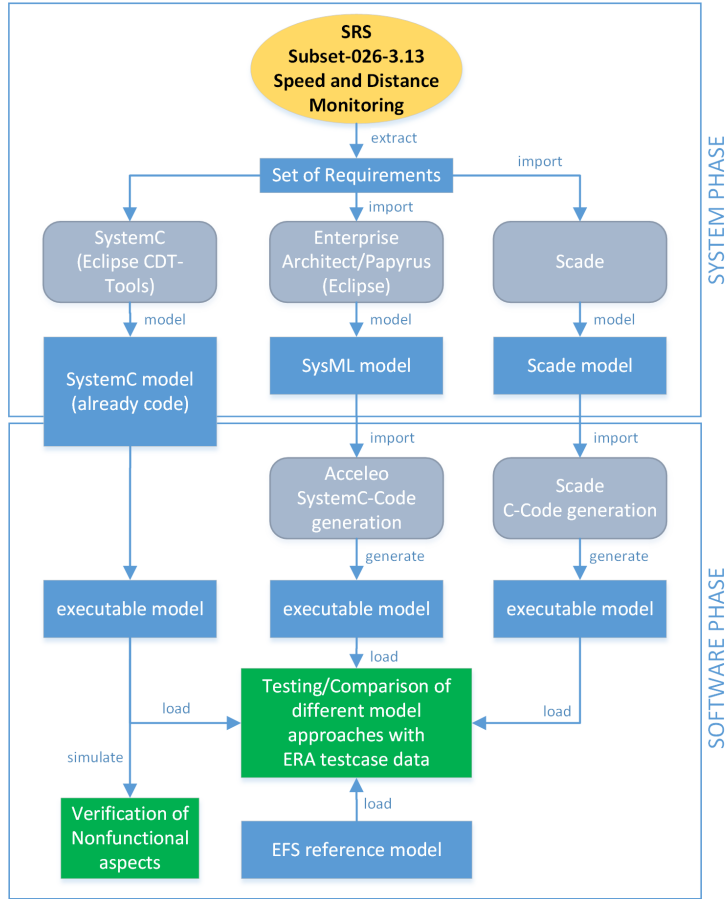


Figure 1: University of Rostock VnV Approach

calculating several modules with inputs from train side, track side and odometry. The University of Rostock focuses on the calculation of parts, which are used for safety critical cases using emergency brakes. This especially includes the EBD (emergency brake deceleration) curves.

### Detailed verification plan

**Goals** On the one hand we are testing extra-functional aspects such as performance and scheduling analyses to give evidence which hardware system is sufficient to meet the system requirements. We want to discover which hardware resources (e.g. number of processors) will be needed for the OBU. This is important to avoid excessive delays to ensure adequate response times in critical

situations. Therefore the University of Rostock will do model based simulation using the inherent simulation environment of SystemC.

On the other hand we use the existing SystemC model to check against a reference model, such as EFS braking curve model. Furthermore we use different tools and means to build additional system models for comparison and verifying behavior.

**Method/Approach** Figure 1 depicts our methodology. Three models will be created from the SRS specification: one SystemC model that is directly implemented into executable code (finished), one SysML model that will be used to produce SystemC code to be executable (not finished) and one SCADE model that will be used to produce C code (not finished). For model verification we use test cases and data provided by WP4 and use the ERA excel datasheet as reference implementation. The created test models will contain behavioral representation of the *Speed and Distance Monitoring* such as state machines, activity diagrams and sequence diagrams. There will be a link to the specification requirements to meet the needs of traceability in terms of verification activities.

**Means** The Artifacts are produced as follow:

- SystemC code which is directly implemented.
- One Test model is the SystemC model. It's code will be generated/transformed by Acceleo from abstract SysML models designed with Enterprise Architect and/or Papyrus.
- C code is produced by Scade using Scade System.
- Executable code is compiled with GNU-C-Compiler `gcc`.
- Reference model delivered by ERTMS Formal Specs.
- Testdata and testcases are provided by ERTMS Formal Specs using the ERA excel data sheet.

**Results** Verification of extra-functional aspects is successfully done for the first iteration of application models. The provided results consist of recommendations on how hardware resources shall be allocated to the future OBU. The modeling activities are still in progress.

**Summary** What we have done:

1. Created an executable model in SystemC.
2. Ran simulations on single, dual and multi core virtual prototypes.
3. Architecture SysML model of the *Speed and Distance Monitoring*.
4. Architecture Scade model of the *Speed and Distance Monitoring*.

5. Defined a reduced set of parameters for calculating braking Curves especially EBD curves.

The next step:

1. Finishing model activities on SysML and Scade.
2. Developing a model transformation/code generation from SysML models.
3. Defining an exchanges interfaces between different model approaches.
4. Run the tests.

**Conclusions/Lessons learned** From the first results, we see that SysML is a very powerful graphical modeling language but to perform code generation it is necessary to have restrictions to it. We will have a domain specific SysML profile to get reliable results from code generation.

**Future Activities** Simulink as a modeling tool is in our interest because there is a bridge to Scade. Simulink provides code generation for hardware description languages such as VHDL. That enables new hardware test scenarios.