# Test generation of the Management of the Radio Communication

Cécile Braunstein Uwe Steinke

December 20, 2013

**Abstract**

This document reports the tests generated for the management of the radio communication function verification. The tests are produced using Model based testing, thus a test model has been created from the specification. Then, the produced tests are applied to a C code generated from a SCADE model.

Since SCADE uses a verified compiler, the verification objectives are to ensure the completeness of the model with regards to the specification (requirement coverage) and to track the possible ambiguities within the specification that leads to a non-uniform implementation of the system under test and the test model.

### 0.0.1 Verification of the Management of the Radio Communication

This section reports the verification activity of SCADE-MoRC. The goal of this activity is, first, to establish the compliance of the SCADE model to the SRS description via testing. Secondly, we want to track the ambiguities within the specification. Finally we want to demonstrate the efficiency of model based testing using the RT-tester tool for system integration testing.

The activity is described in the Verification and Validation Plan section *6.1.2.7 System Integration Testing (Uni Bremen/DLR)* [1] In short, we develop a test model from the specification, generate tests and use the code generated from SCADE to perform software-in-loop testing.

**Object of verification** MoRC ERTMS function baseline 3.

The system under test (e.g. the verification object) is the C code generated from a SCADE model and described here `https://github.com/openETCS/model-evaluation/tree/master/model/SCADE_Siemens/Subset_026_Chapt_3.5_ManagementOfRadioCommunication/Generated_C_Code`. It describes the Management of the radio communication at the software phase.

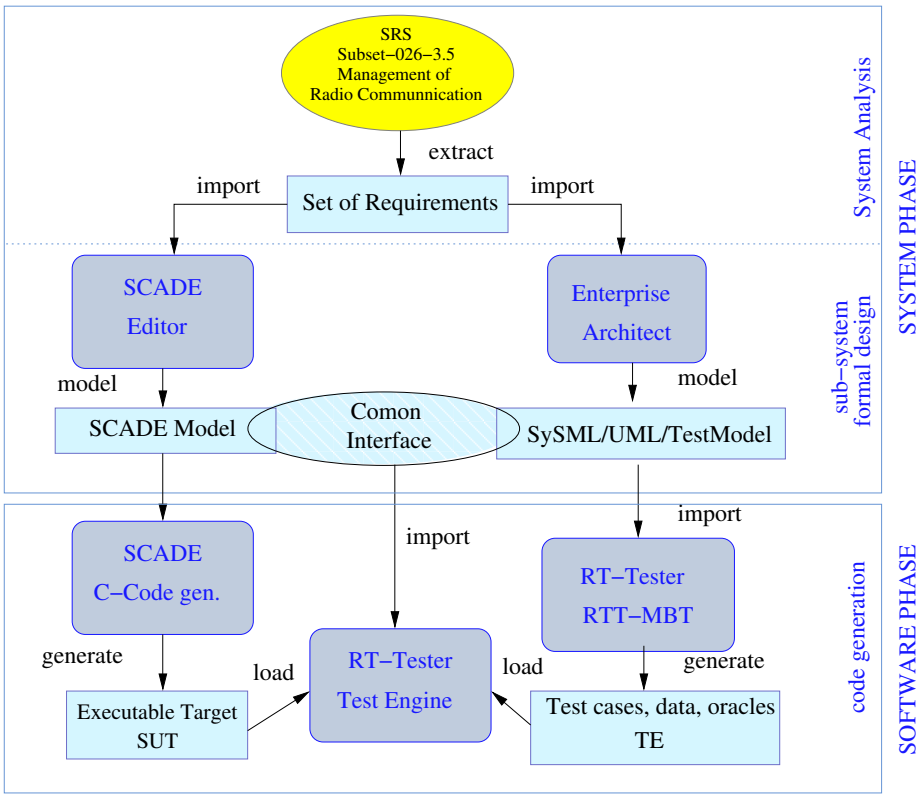


Figure 1: SCADE/RT-Tester methodology

**Available specification** The specification is described in the SUBSET-026[4] chap 3.5 [5]. It describes the communication protocol between the EVC and the RBC or balises. In particular, how the EVC initiates and terminates a communication.

### Detailed verification plan

**Goals** To achieve what has been defined previously a test model in SysML has been developed. The description of this verification artifact may be found here `https://github.com/openETCS/model-evaluation/blob/master/model/EA-SysML/new_version/doc/ea_sysml_report.pdf`

Our main goal is to validate the SCADE model by test simulation. The tests are produced by a model of the subset-026 chap 3.5 described as a state machine.

**Method/Approach** Figure 1 depicts our methodology. From the SRS specification, two models are designed: one SCADE model that will be then

used to produce C code and one SysML model for generating tests.

Our test model contains the behavioral representation of the MoRC, the set of requirements of the chap 3.5, and the link between the behavior and the requirement. Most of the requirements may be represented as single transition or state in the test model state machine. Nevertheless, some of the requirements may be only modeled as a path of the state machine, we choose to represent those as LTL formula, added as constraints of the test model. For example the `REQ-3.5.10` describes the steps needed for the establishment of the radio communication order by the RBC. This requirement explicits a particular path of the test model, thus this path should exists in our test model. To ensure that this particular test will be generated a LTL formula has been added (See [3] for more details).

We need then to link the system under test and the test model. A common inputs/outputs interface of the system under test has been made. The inputs drive the tests and the outputs are the observational points to state if a test pass or fail. Hence, the two models should respect the same interface.

After the modeling phase, starts the test generation phase. Two strategies have been used for the tests generation. First, we generate a set of test following the common behavioral coverage strategies ensuring the following coverage :

- Basic control state coverage (BCS)
- Transition coverage (TC)
- Modified condition/decision coverage (MC/DC)
- Hierarchic transition coverage (HITR)
- Basic Control states Pair coverage(BCPAIRS)

We then apply a requirement coverage strategy that consists of generating tests that covers all the requirements. As each requirement are linked to an artifact of the test model, part of the test cases are generated as subset of the coverage mention above. In addition, test cases from the LTL are also provided.

Finally the test engines run the SUT with the stimuli from the test model. In parallel, it runs the test oracles that states if the test pass or fail.

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

- C code comes from SCADE model.
- Test model is a SysML model designed with Enterprise Architect.
- Test cases, tests data and test oracles are generated with RT-tester.
- Executable code compile with gcc.
- Code run within the RT-Tester engine.

SCADE is EN 50128 certified at SIL 3/4, RT-tester is also certifiable SIL3 as shown in [2].

| Test cases covered | # tests generated |
|---|---|
| BCS | 14 |
| TC | 40 |
| MC/DC | 79 |
| BCPAIRS | 33 |
| HITR | 12 |
| LTL | 2 |
| Total Tests | 180 |

Total Requirements cover 40

Table 1: Test cases generation summary

**Results** Table 1 resumes the set of automatically generated tests. The set of tests cover 40 of the requirements from the chap3.5.

The simulation result of the C code is not yet finished. The analysis of the failed test is a work in progress.

**Summary** What we have done:

1. Created a test model in SysML.
2. Generated test cases.
3. Ran SCADE model against test procedure produces by RT-tester.

The next step:

1. Analyze the result of the test procedure.
2. Coordinate DLR/SIEMENS/Uni Bremen interfaces.
3. Run the test on the DLR lab.

**Conclusions/Lessons learned** From the first result, we could see that one chapter of the specification is not self-contained. This leads to different interpretation in the modeling and thus some non equivalent behaviors.

Moreover some missing information in the specification leads to a under constraints test model. It affects the test generation by providing some non realistic test cases. Some variable behavior that were not mentioned in the spec are considered as free and may have any possible value in their definition range.

**Future Activities** We will also provide the ceiling speed monitoring function to enrich the test model and apply new model based testing approach.

# References


[1] D4.1 openETCS validation & verification plan. Delivrable OETCS/WP4/D4.1V00.09, openETCS, September 2013.

[2] Jörg Brauer, Jan Peleska, and Uwe Schulze. Efficient and trustworthy tool qualification for model-based testing tools. In Brian Nielsen and Carsten Weise, editors, *Testing Software and Systems*, volume 7641 of *Lecture Notes in Computer Science*, pages 8–23. Springer Berlin Heidelberg, 2012.

[3] Cécile braunstein. Radio communication management -srs subset-026-3.5-sysml model. report, openETCS, 2013.

[4] UNISIG. SUBSET-026 – system requirements specification. SRS 3.3.0, ERA, March 2012.

[5] UNISIG. *SUBSET-026 – System Requirements Specification*, SRS 3.5 Management of the Radio Communictation. In [4], March 2012.