ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

openETCS

Workpackage 4
Verification & Validation Strategy
Model-Based Testing in openETCS

supported by:

Federal Ministry of Education and Research

Région de Bruxelles-Capitale

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN

openETCS@ITEA2 Project

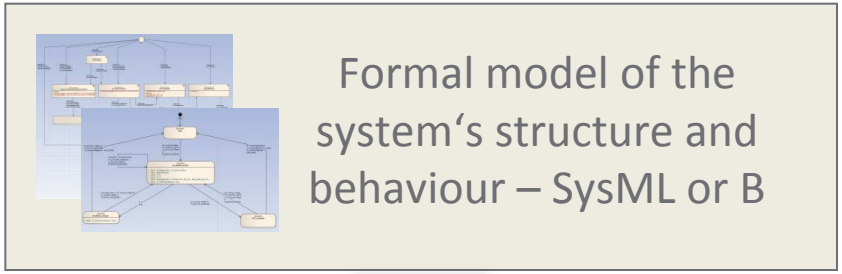
Jan Peleska, University of Bremen

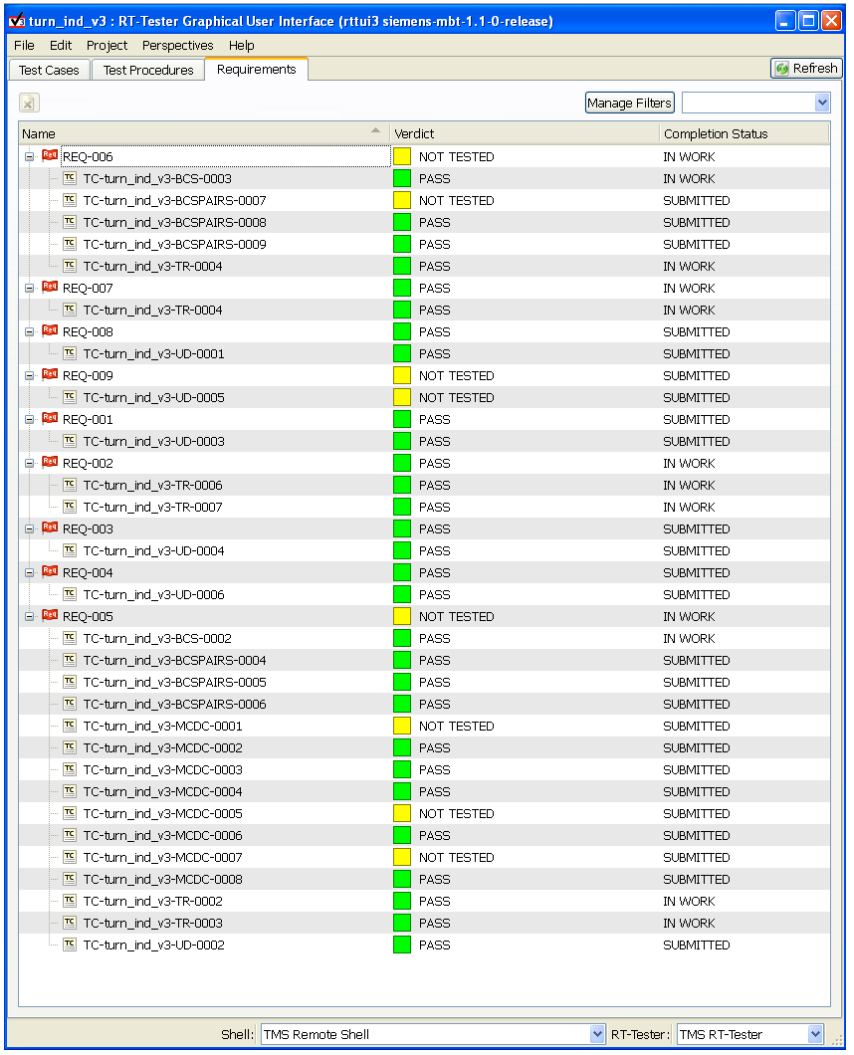
Paris, 03.07.2013

Model-Based Testing (MBT): instead of manual test case identification, test data calculation and test procedure coding,

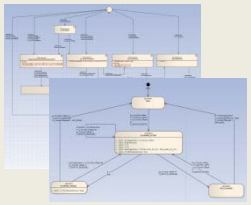
- Create model specifying expected behaviour of system under test (SUT)
- Identify relevant test cases from model in an automated way
- Calculate test data by means of constraint solver
- Generate test procedures in model-based development style

Model-Based Testing Paradigm





Model-Based Testing Paradigm



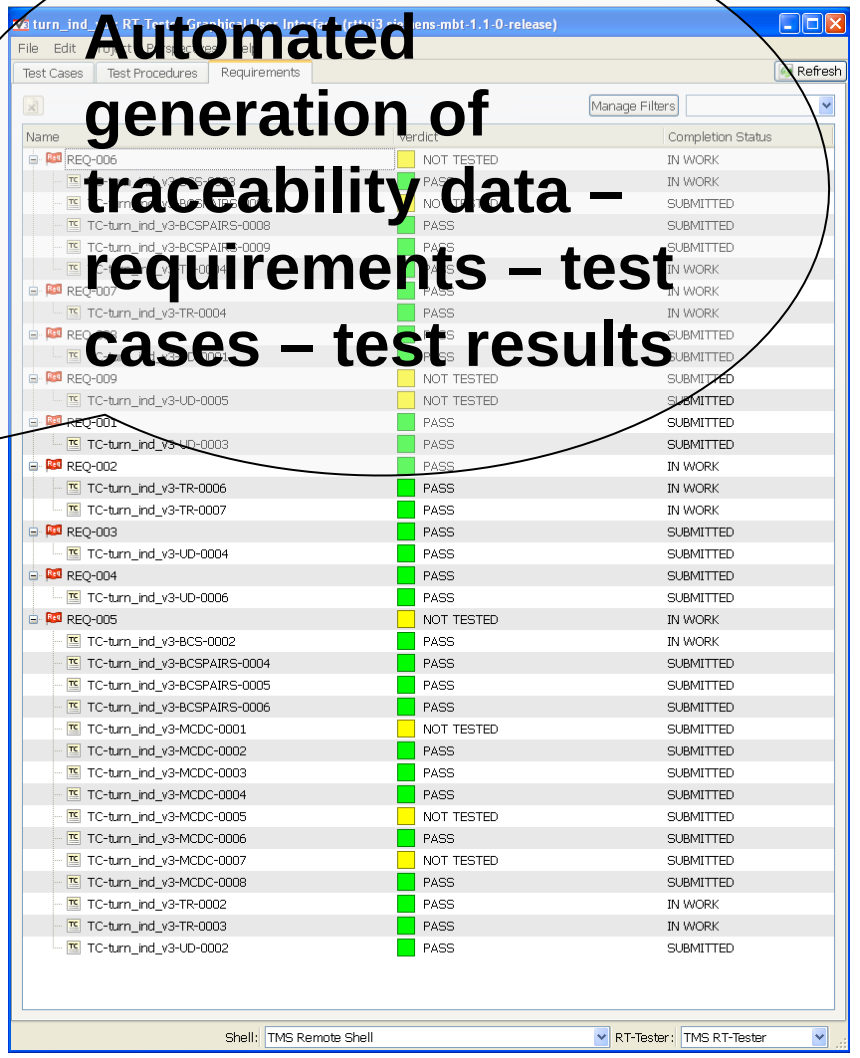
Formal model of the system‘s structure and behaviour – SysML or B

Test case-/Test data generator

Test procedure generator

Automated execution against SUT

Automated generation of traceability data – requirements – test cases – test results



Model-Based Testing Paradigm – System Test

The test engine triggers $\vec{x}(t)$ and checks $\vec{y}(t)$ in hard real-time

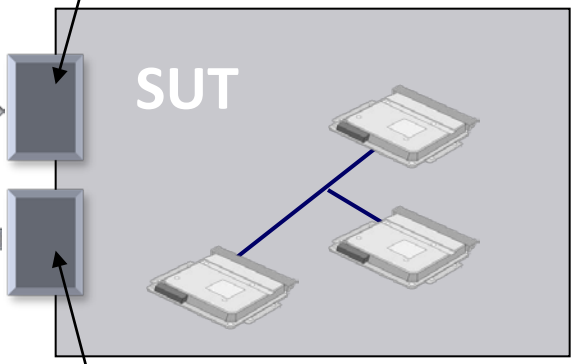
Stimulation of sensors and busses

$\vec{x}(t)$

$\vec{y}(t)$

Observation of actors and busses

Sensors/busses Input interfaces



Actors/busses Output interfaces

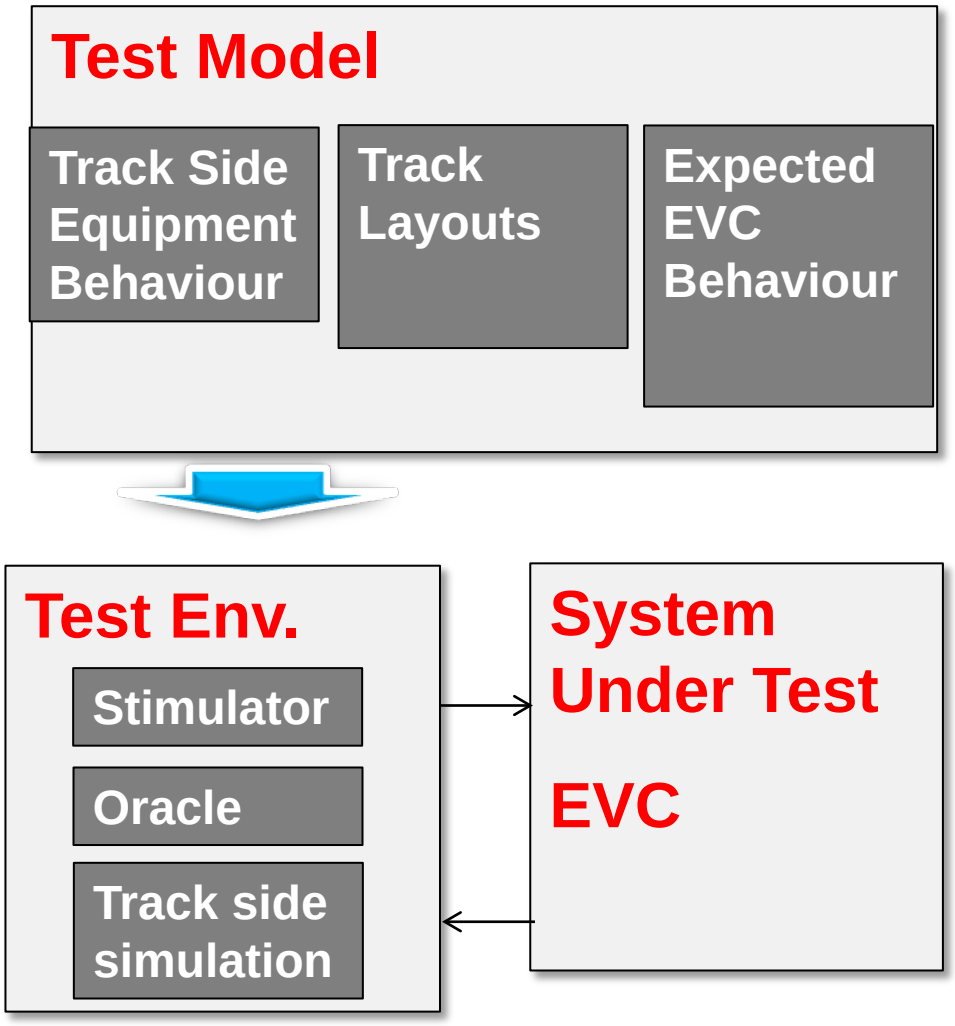
MBT Application Scenarios

Model-based development approach in openETCS suggests model-based testing approch

Three application scenarios for model-based testing in openETCS

- System integration testing
- Software integration testing
- Object code verification

MBT in System Integration Testing



Model components for testing the EVC

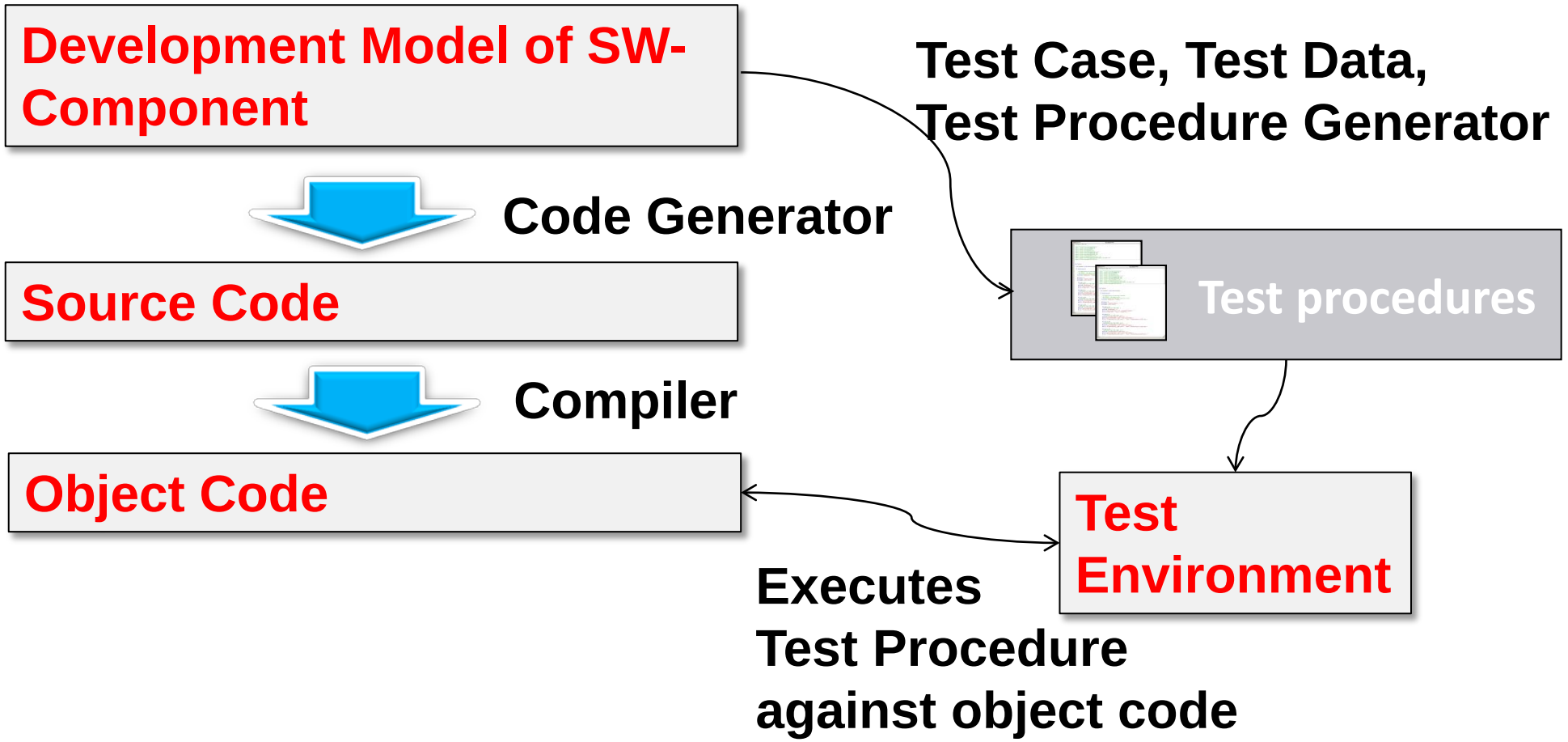
- Expected behaviour of EVC
- Track layouts with different variants allowing for verification of all crucial properties of the EVC
- Simulations of track-side equipment behaviour

MBT for Object Code Verification

Objective: find efficient method for verifying that generated source code complies with model, and compiled object code complies with source code (or model)

- Use automated test suite derived from SW component model to show that source and object code comply with model
- ⇨ No code generator and code generator validation required

MBT for Object Code Verification



Conclusion

Model-based testing has considerable benefits for openETCS

- **Efficiency gain by automated test case, test data, and test procedure generation from the software component's sub-model**
- **Testing can always be performed on 3 levels**
  - Model-in-the-loop – V&V for the model
  - Software-in-the-loop – Verify code
  - Hardware-in-the-loop – V&V for integrated HW/SW system

- **Thank you for your attention!**

- **For further regular information, please send an E-Mail to: wp4+subscribe@openetcs.org**

Jan Peleska
University of Bremen
jp@informatik.uni-bremen.de
Tel: +49 (0) 421 218 63961