Getting Started With Test Modelling for RT-Tester

Jan Peleska
jp@verified.de
2013-10-07

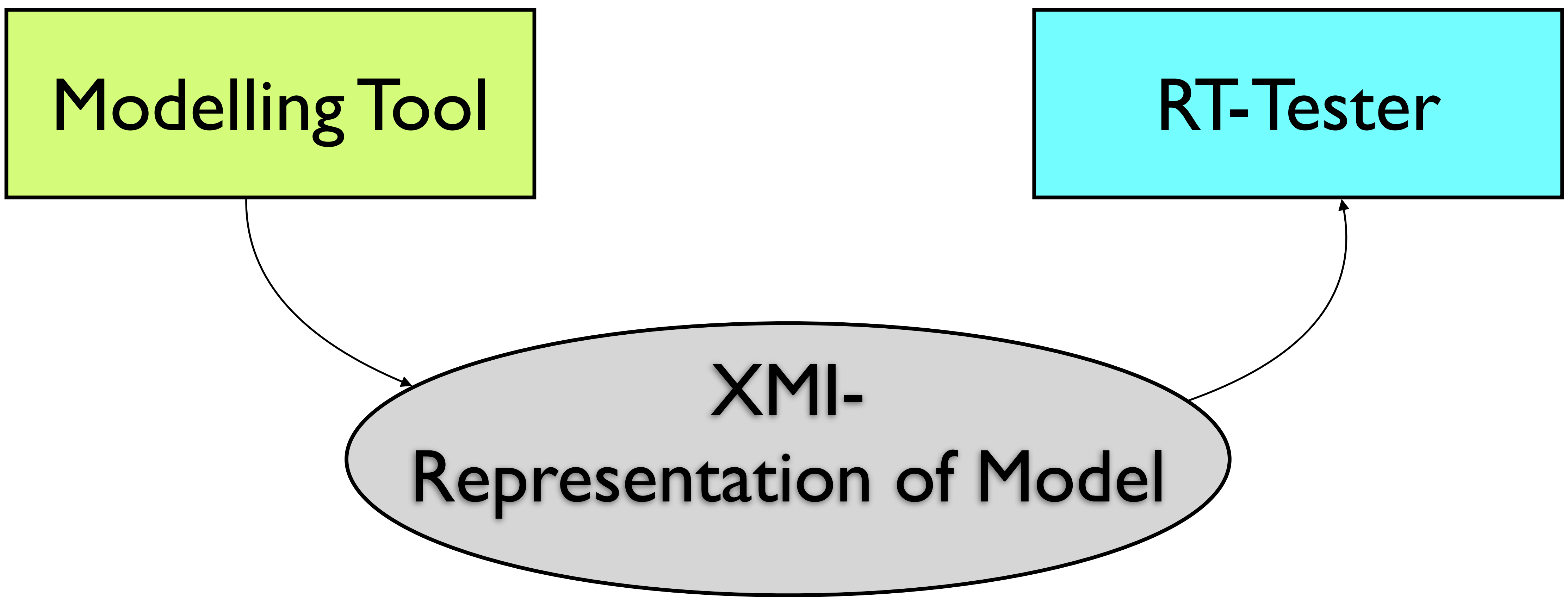
Objectives of Test Models

- Specify expected behaviour of system under test (SUT),
- Specify environment behaviour, to be simulated during the test execution
- Specify interface between SUT and environment, as is visible during the test execution
- Specify traceability information from requirements to model elements

Benefits of MBT

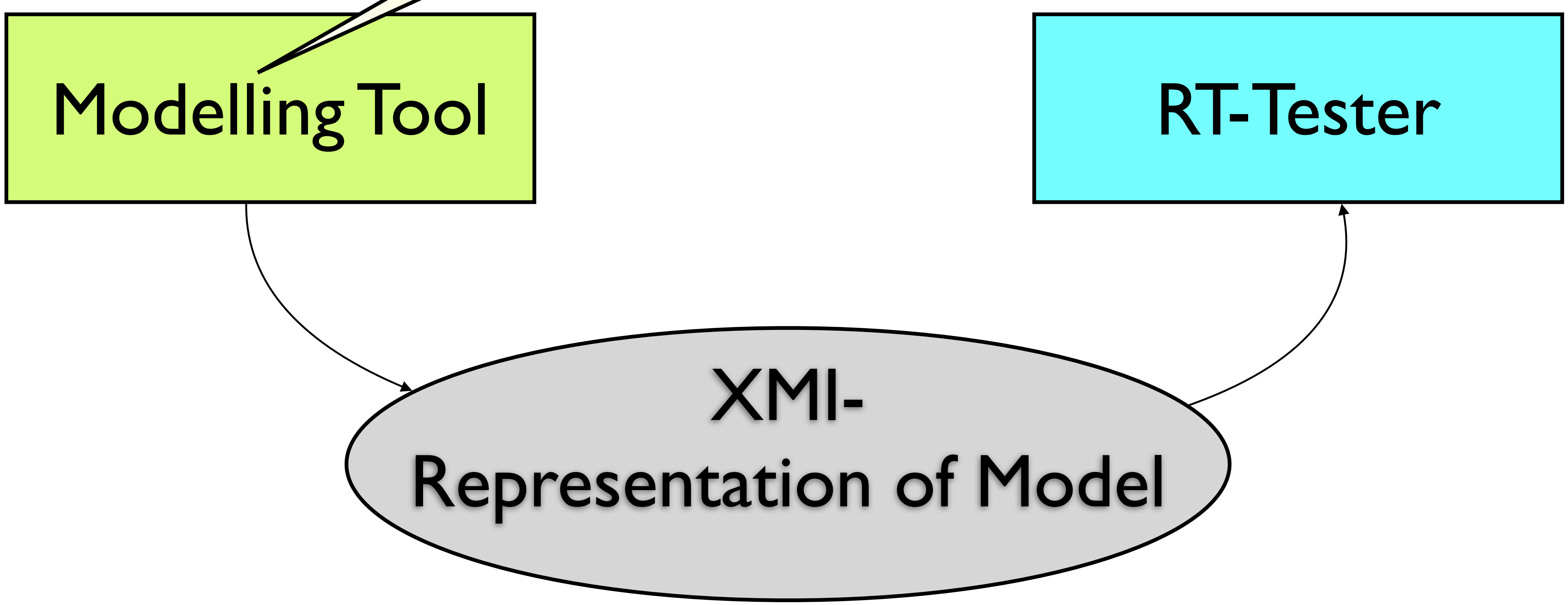
- Relevant test cases are identified automatically in the model
- Concrete test data is calculated for each test case in an automated way
- Simulations are created automatically from the model
- Traceability matrix linking requirements – test cases – test procedures – test results is created in an automated way

How Modelling Tools and RT-Tester Cooperate

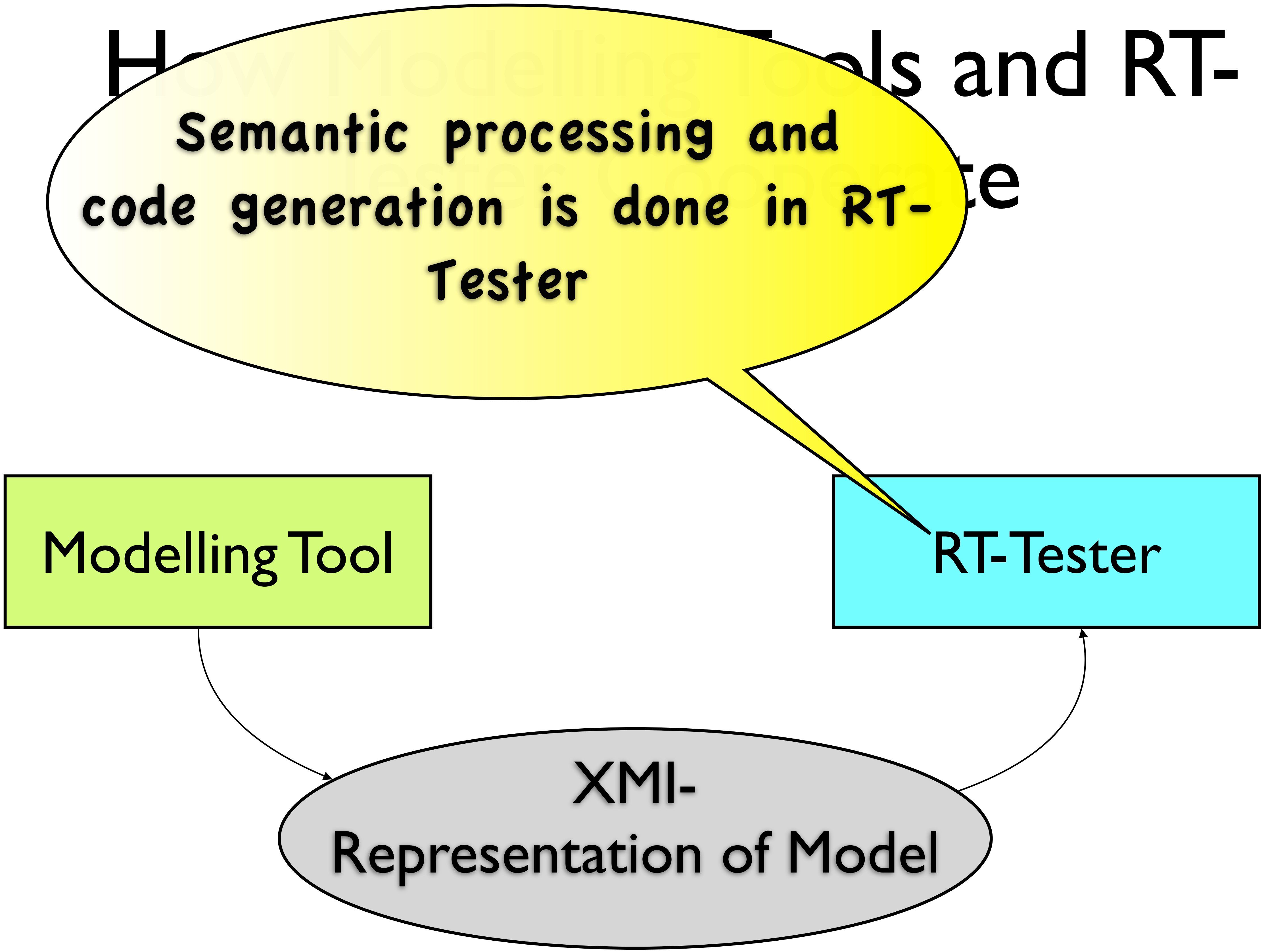


How Modelling Tools and RT-Tester Cooperate

Modelling tool is used as "drawing device"



How Modelling Tools and RT-Tester Cooperate



Modelling Tools

- Test model for model-based testing (MBT) with RT-Tester are written in SysML
- Supported modelling tools
  - Enterprise Architect: supported now
  - Artisan Studio: supported now
  - Rhapsody: supported now
  - Papyrus: supported by January 2014

Creating Test Models

- Ingredients of a test model
  - Interface between test environment (TE) and system under test (SUT)
  - TE sub-model
  - SUT sub-model
  - Requirements

Creating Test Models

- SysML language elements
  - Block definition diagrams
  - Internal block diagrams
  - Interfaces
  - State machines
  - Requirements
  - <<satisfy>> relationship from model elements (incl. constraints) to requirements

Specialisation on EVC Testing

- Test areas (examples)
  - RBC–EVC communication
  - Ceiling speed monitoring
  - Target speed monitoring
  - …

Specialisation on EVC Testing

These test topics are currently in the main focus of University of Bremen

- Test areas (examples)
  - RBC–EVC communication
  - Ceiling speed monitoring
  - Target speed monitoring
  - …

Example – Ceiling Speed Monitoring

- Inputs to SUT
  - Given most restrictive speed profile – V_mrsp
  - Estimated speed V_est
  - Location x, such that EOA or change of speed profile is "far away" – target speed monitoring does not apply
  - SBI_available – service brake implemented AND its use is authorised by national value
  - allowRevokeEB – flag indicating whether EB command may be revoked when speed is <= permitted speed limit
  - Gradient profile

Example – Ceiling Speed Monitoring

- Outputs from SUT
  - Warning, overspeed status indication to DMI
  - SBI – service brake intervention command
  - EBI – emergency brake intervention command

Model Exploration . . .

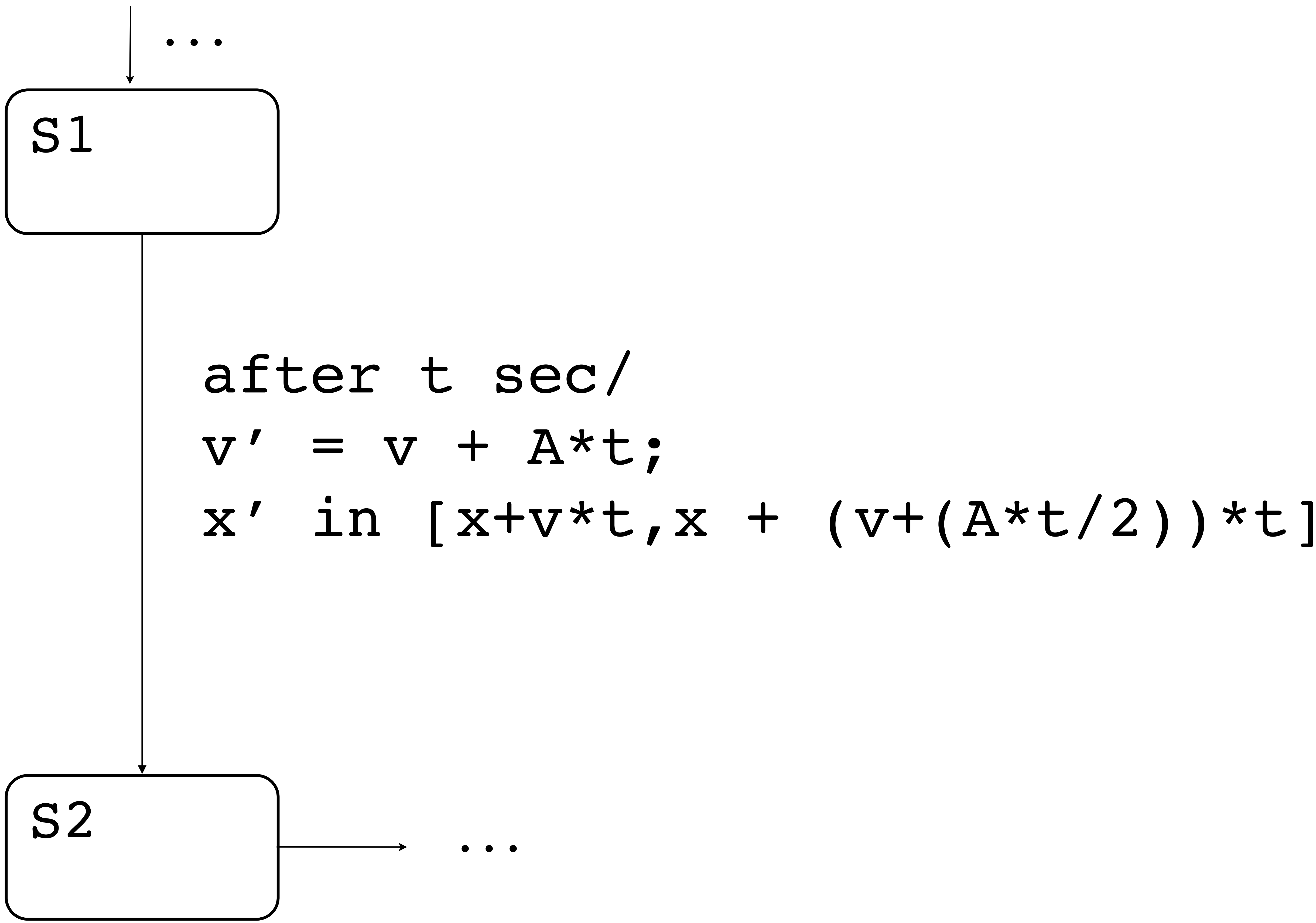
Requirements for Environment Simulation

- Simulate physically reasonable acceleration / deceleration
- Simulate admissible speed (V_est <= V_mrsp)
- Simulate various speed violations, as needed to test all control aspects of the SUT model
- Location always consistent with “integration over speed”
- Calculations according to EEIG : 97E881

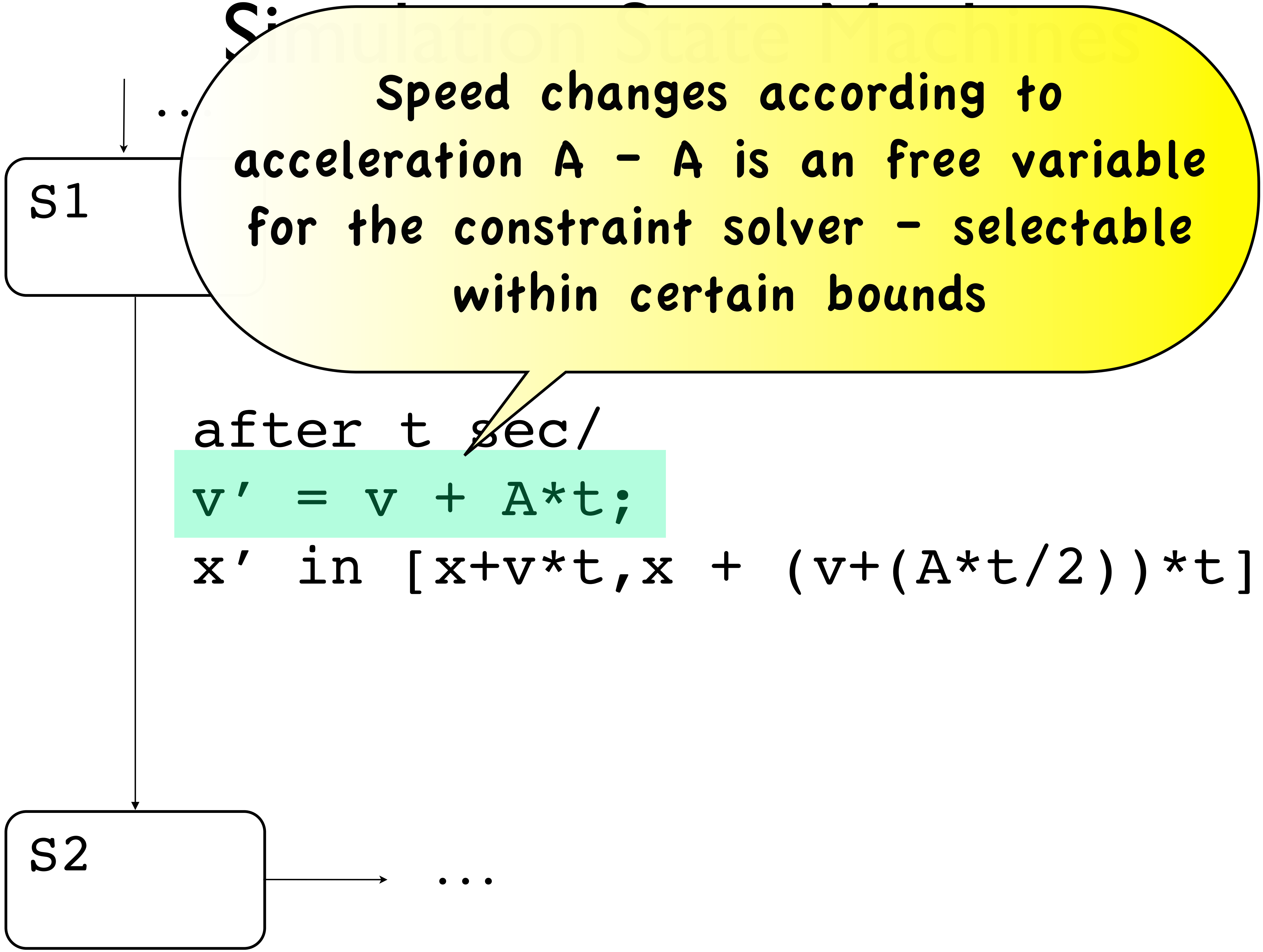
Simulating Deceleration

- Use braking curves influenced by
  - Gradient profile
  - Different behaviour of SBI and EBI
  - Train type

Simulation State Machines



Simulation State Machines



Simulation State Machines

...

S1

Location changes according to speed, with some degree of freedom to fixed by the constraint solver

```
after t sec/
v' = v + A*t;
x' in [x+v*t,x + (v+(A*t/2))*t]
```

S2

...