Advancement in Automated Simulation and Testing Technology for Safety-Critical Avionic Systems

Prof. Dr. Jan Peleska *(Verified Systems International GmbH)*

Dipl. Ing. Klemens Brumm *(Airbus)*

Dipl. Ing. Gunnar Jonas *(RST Rostock System-Technik GmbH)*

Dipl. Inf. Tobias Hartmann *(TZI Center of Information Technology – University of Bremen)*

Overview

1. Model-based simulation and testing:

   Automated simulation, test case and test data generation from powerful specification formalisms

2. Large-scale simulation:

   Integration of large simulation environments, consisting of parallel, possibly interacting, tasks

3. Hard real-time test-bench technology:

   Scalable hard real-time execution platforms for the testing and simulation software

4. Conclusion and Background:

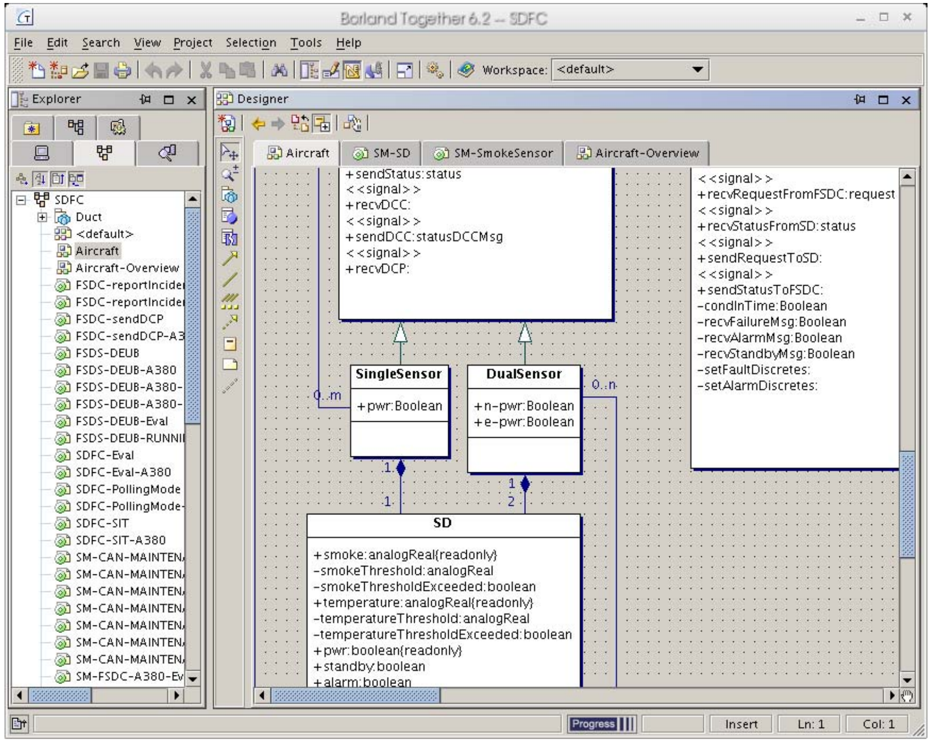
   Research project KATO-TP13

Part1 – Model-based simulation and testing

Objectives:

Instead of manually programming explicit I/O sequences to be performed by simulation and test components ...

- ... generate simulation and test data from specifications in an automatic way
- ... perform on-the-fly checking of system under test behavior against specified expected results



Part1 – Model-based simulation and testing

Objectives (continued):

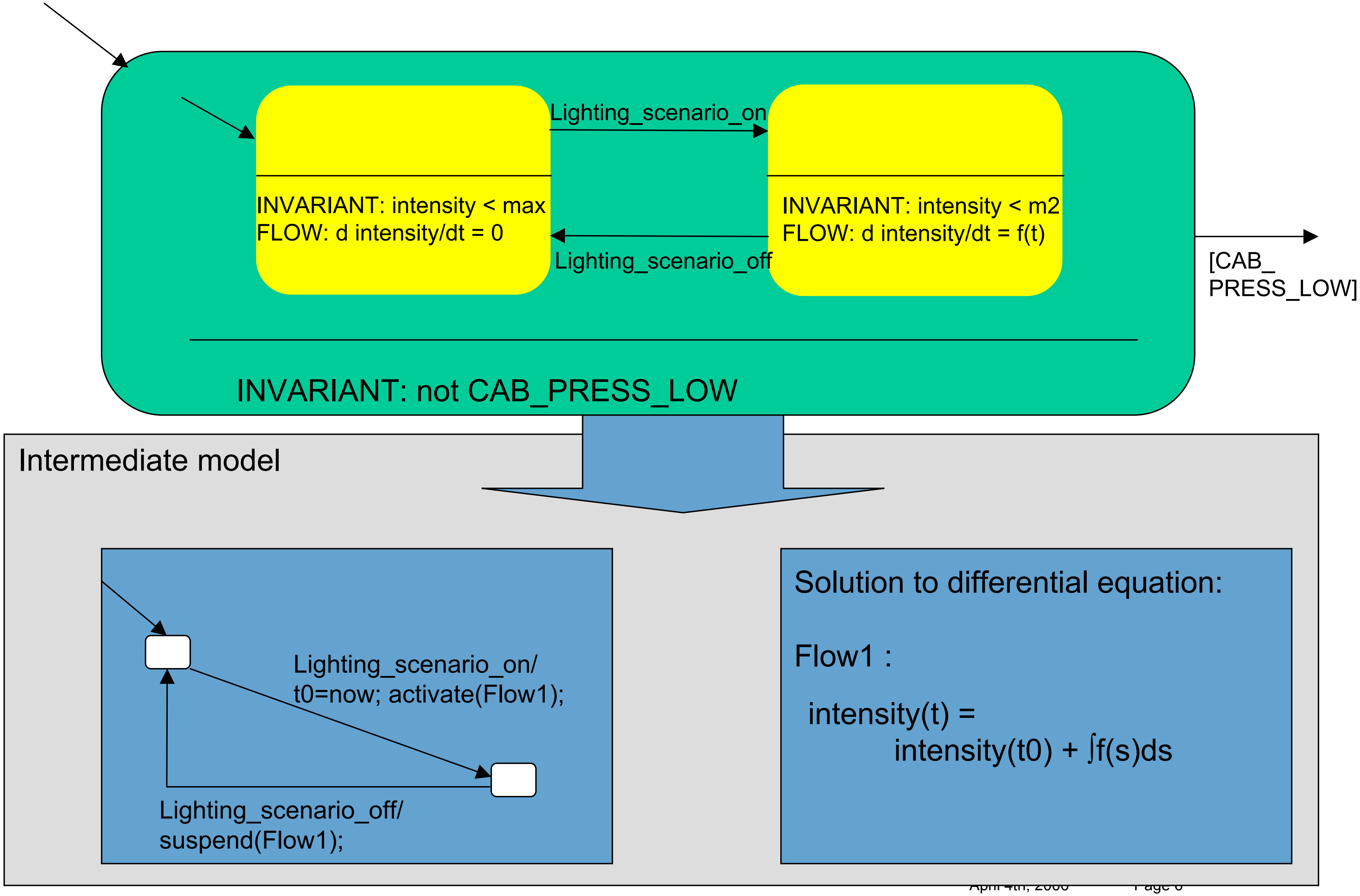
- Provide unified approach for simulation and testing on different levels:
  - Software unit testing
  - Software integration testing
  - HW/SW integration testing
  - System testing
  - Wide area inter-site testing
- Allow for various specification styles made-to-measure for customers’ needs and skills

Part1 – Model-based simulation and testing

Solutions:

- Transform various specification formalisms to intermediate model representation
- Exercise test case generation algorithms on intermediate model
- Compile intermediate model into executable distributed simulation/test program plus test data
- All concepts implemented in RT-Tester Test Automation System

Part1 – Model-based simulation and testing



Lighting_scenario_on

INVARIANT: intensity < max
FLOW: d intensity/dt = 0

Lighting_scenario_off

INVARIANT: intensity < m2
FLOW: d intensity/dt = f(t)

[CAB_
PRESS_LOW]

INVARIANT: not CAB_PRESS_LOW

Intermediate model

Lighting_scenario_on/
t0=now; activate(Flow1);

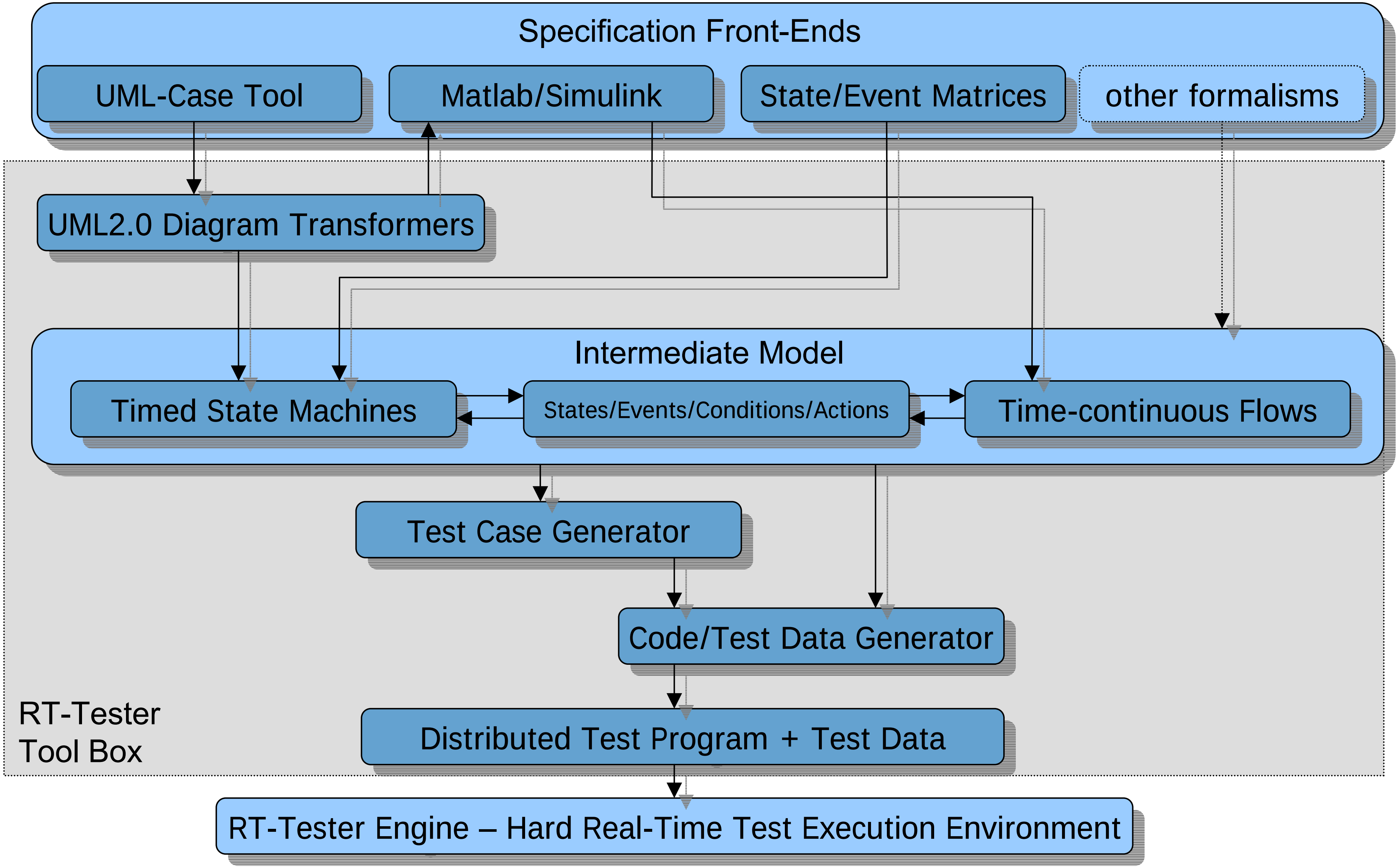
Lighting_scenario_off/
suspend(Flow1);

Solution to differential equation:

Flow1 :

$$\text{intensity}(t) = \text{intensity}(t0) + \int f(s)ds$$

Part1 – Tool chain - overview



Specification Front-Ends

UML-Case Tool

Matlab/Simulink

State/Event Matrices

other formalisms

UML2.0 Diagram Transformers

Intermediate Model

Timed State Machines

States/Events/Conditions/Actions

Time-continuous Flows

Test Case Generator

Code/Test Data Generator

Distributed Test Program + Test Data

RT-Tester Tool Box

RT-Tester Engine – Hard Real-Time Test Execution Environment

Part2 – Large scale simulation

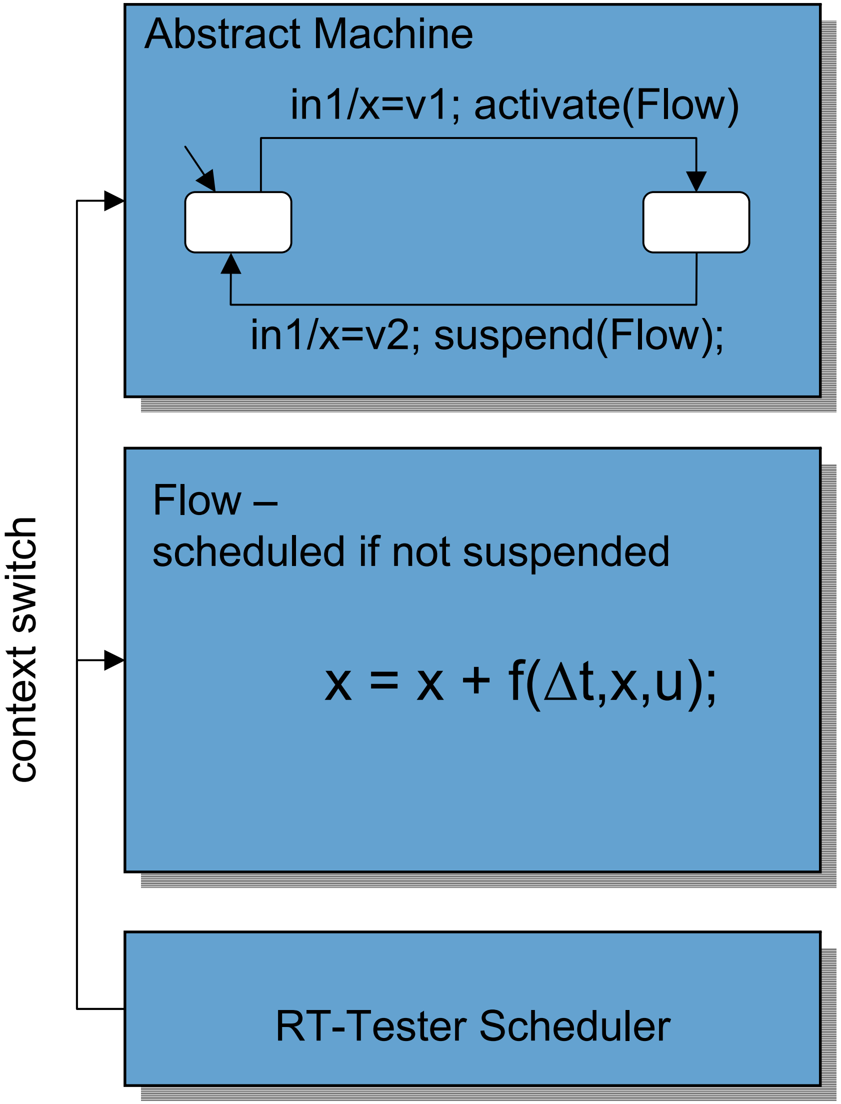
Objectives:

- Integrate large numbers of simulations as parallel tasks in the testing and simulation environment
- Provide global access to simulation and test data to all components
- Distinguish between event- based and state-based – discrete and analog signal data
- Ensure execution in hard real-time

Part2 – Large scale simulation

Solutions:

- Multi-threading architecture with high-speed context switching in user space
- Abstract Machines encapsulate state-based sequential simulations as threads
- Flows encapsulate Δt-integration steps of time-continuous data changes as threads



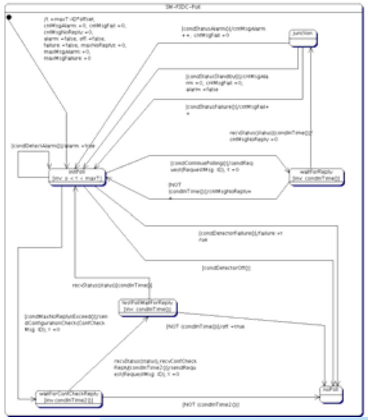
Part2 – Large scale simulation

Solutions:

- UML2.0 Statecharts simulations encapsulated in abstract machines – may activate and suspend flows as special actions
- Matlab/Simulink solutions of differential equations encapsulated in flows
- Customized simulations programmed in
  - Real-Time Test Language RTTL with
  - host language C/C++

  encapsulated in abstract machines or flows
- RT-Tester build tool integrates all simulations in one execution environment

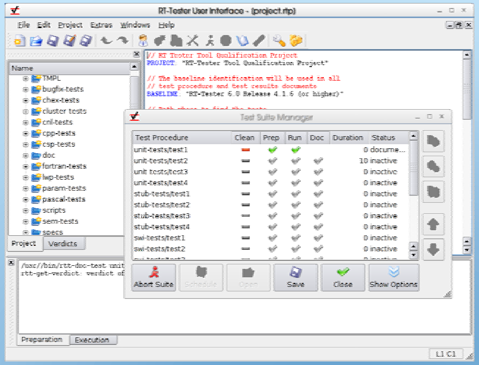
Part2 – Large scale simulation



Statecharts

Real-Time
Test Language

```
@abstract machine temp1() {
 @output port channelTemp1_p on channelTemp1;

 @PROCESS:
   channelTemp1_t ct1;
 @rttBeginTestStep;
   int maxNoFire = 20;
   ct1.actualMeasuredTemp1 = init;
   while (@rttIsRunning) {
       int i = 1;
       while ( i <= maxNoFire ) {
         @rttPut(channelTemp1_p, &ct1);
         @rttWait(1000 _ms);
         i++;
       };
     }
  @rttEndTestStep;
}
```



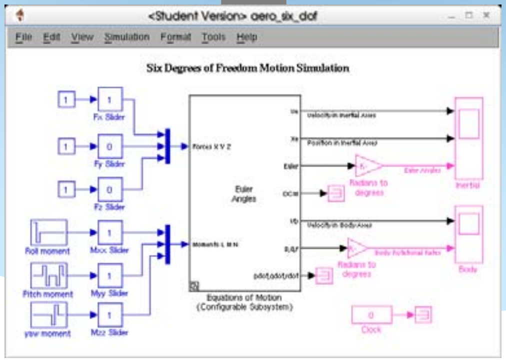
```
// RT Tester Tool Qualification Project
PROJECT: "RT-Tester Tool Qualification Project"

// The baseline identification will be used in all
// test procedure and test results documents
BASELINE: "RT-Tester 6.0 Release 4.1.6 (or higher)"
```

Test Procedure	Clean	Prep	Run	Doc	Duration	Status
unit-tests/test1	−	✓	✓		0	docume...
unit-tests/test2	−	✓	✓	✓	10	inactive
unit tests/test3	−	✓	✓	✓	0	inactive
unit-tests/test4	−	✓	✓	✓	0	inactive
stub-tests/test1	−	✓	✓	✓	0	inactive
stub-tests/test2	−	✓	✓	✓	0	inactive
stub-tests/test3	−	✓	✓	✓	0	inactive
stub-tests/test4	−	✓	✓	✓	0	inactive
swi-tests/test1	−	✓	✓	✓	0	inactive
swi-tests/test2	−	✓	✓	✓	0	inactive

```
/usr//bin/rtt-doc-test unit
rtt-get-verdict: verdict of
```

RT-Tester

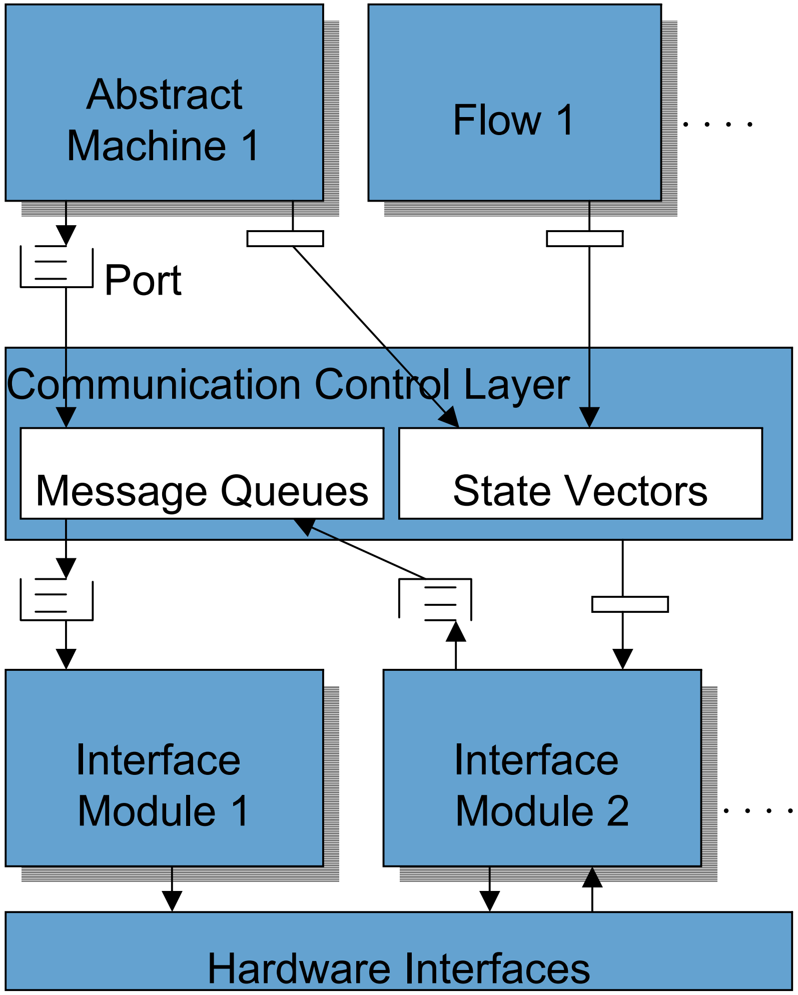
Matlab/Simulink



Part2 – Large scale simulation

Solutions:

- Layered communication architecture
- Message queues implement discrete events in time
- Vectors implement global state components
- Transparent access to events and state vectors within distributed system



Part2 – Large scale simulation

Solutions:

- Universal port abstraction to access all types of interfaces
- Subscription mechanism for states and events provided by
  - Simulations
  - System Under Test (SUT)
  - HW-in-the-loop components
- On-the-fly switching between
  - Simulation S providing state data x
  - HW-in-the-loop original equipment E producing x

  possible: S and E use equivalent ports

Part3 – Hard real-time test bench technology

Objectives:

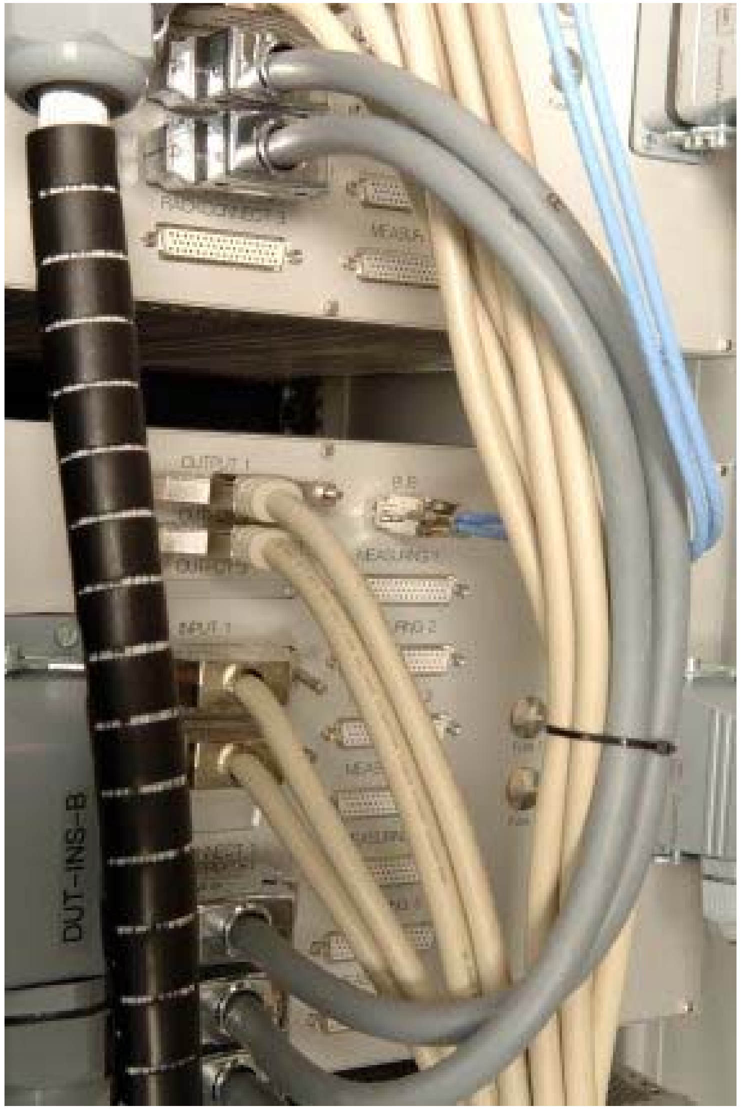
Develop novel test bench technology with

- scalable performance
- scalable number of interfaces
- guaranteed hard real-time properties
- modular architecture

Part3 – Hard real-time test bench technology

Objectives (continued):

- high degree of re-use for different systems under test
- optimized cost/performance ratio by combining off-the-shelf components with customized HW/SW solutions



Part3 – Hard real-time test bench technology

Concepts & solutions – HW:

Scalable test engine power:

- Cluster architecture
- 2 or 4 CPU PC cluster nodes
- High-speed DMA-based cluster communication

Part3 – Hard real-time test bench technology

Concepts & solutions – HW:

Scalable test engine power:

- Distributed interface back-planes connected to different cluster nodes:
  - PCI
  - USB2
  - CAN
  - VME
  - cPCI

Part3 – Hard real-time test bench technology

Concepts & solutions – Software:

Guaranteed hard real-time properties

- Hard real-time kernel extension for Linux
- Simulations and tests run on reserved CPUs – no interference from operating system

Part3 – Hard real-time test bench technology

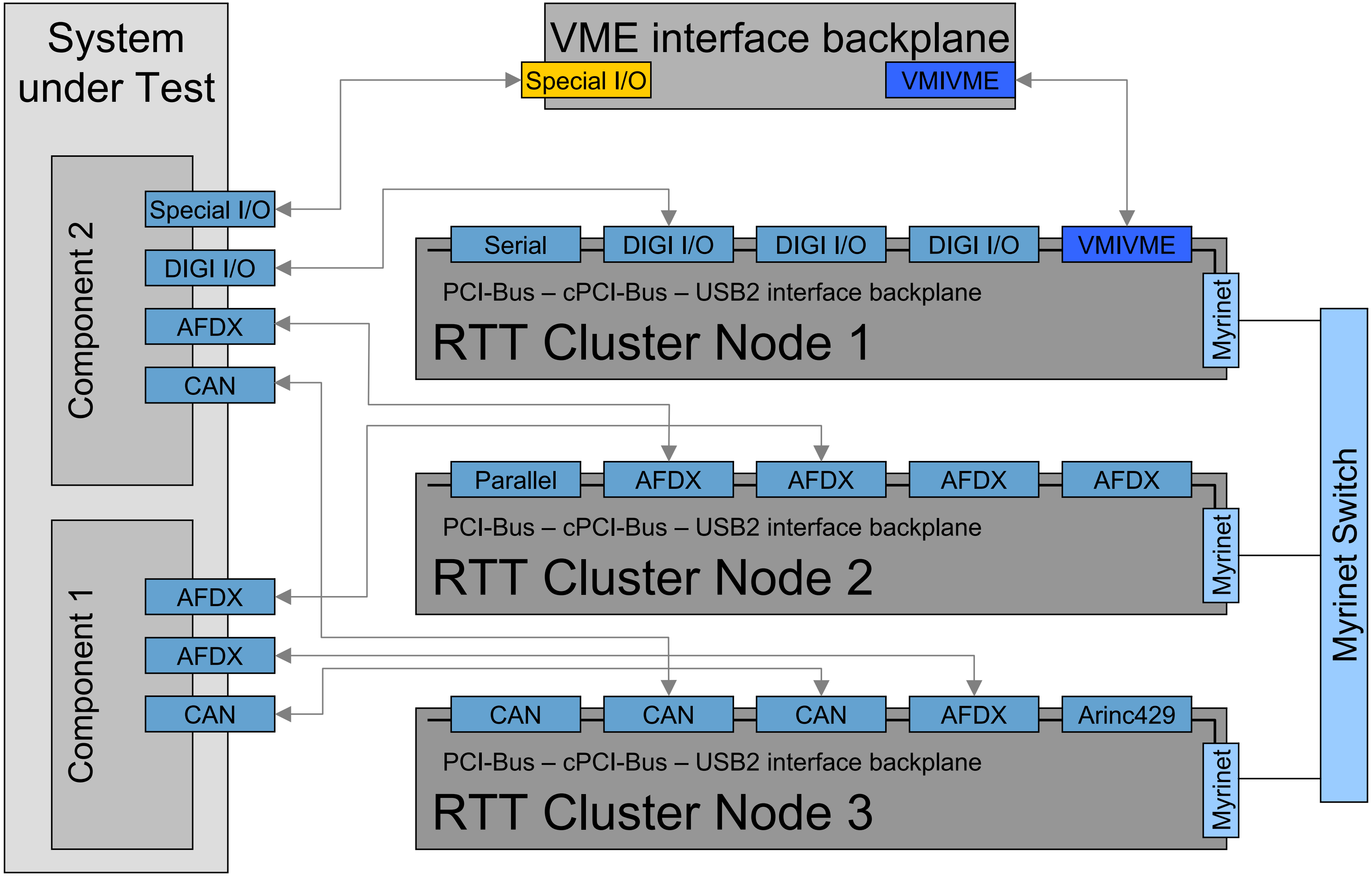
Concepts & solutions – SW:

Guaranteed hard real-time properties

- Scheduling

  precision < 3µs
- Event communication Simulation ↔ Interface < 100µs
- Standard software usable on non-reserved CPUs

Part3 – Hard real-time test bench technology



Conclusion and Background

We have presented novel results on

- Model-based simulation and testing,
- Large-scale simulation,
- Hard real-time test-bench technology,

investigated within research project

KATO-TP13 – a project of the German LUFO III aerospace research program

Conclusion and Background

Research project KATO-TP13:

- Techniques for requirements validation by combined
  - Structured reviews
  - Simulation
  - Model checking
- Exploitation of formally modeled domain knowledge from
  - Aircraft domain (e.g. ATA chapters)
  - Manufacturer’s expertise (re-usable concepts)

  for verification and testing of avionics systems

Conclusion and Background

- Research project KATO-TP13 (continued)
  - Novel testing technology covering both
    - Hardware test bench technology:
      - Scalable performance
      - Flexible test bench adaptation to different systems under test (SUT)
    - Software for automated simulation and testing:
      - Hard real-time platform for executing large networks of simulations
      - Specification-based testing: automated test case generation and checking of SUT responses against specification models

Conclusion and Background

- KATO-TP13 – cooperation partners:

Airbus, Hamburg

University of Bremen
Center of Information
Technology

RST Rostock
System-Technik
GmbH

Verified Systems
International GmbH
Bremen