Continental
The Future in Motion

Senses for Safety

Driver assistance systems help save lives

Reusing SystemC models in heterogeneous environments by Coside coupling and export

Outline

The Basics

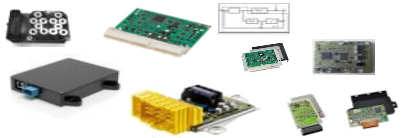
If a car assists you or drives you automatically, it has to …

Sense

…sense its environment and vehicle´s current state...

Plan

…plan its actions...



Act

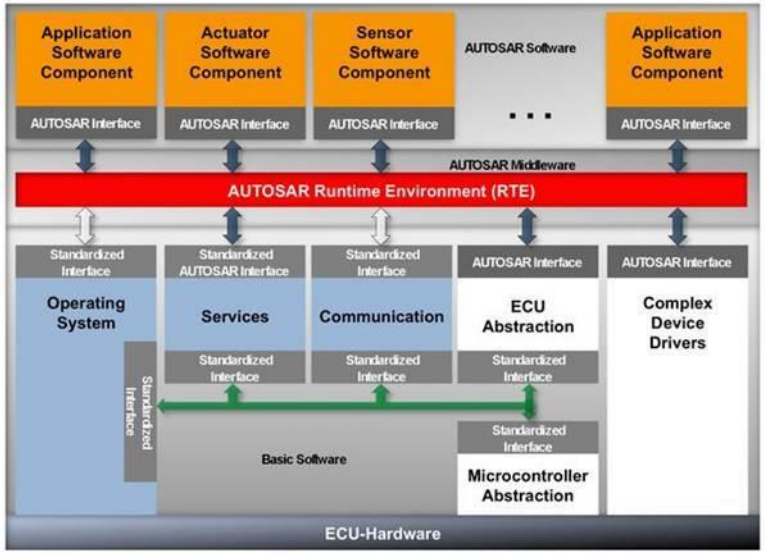
...act using the car´s actuators and control systems.

Example product

- Motor
- Valve block
- Electronic Control Unit (ECU)
  - Microcontroller (MCU)
  - Mixed-Signal IC (PCU)

What do we model for which purpose

- MCU standalone
  - OS and driver development
- MCU + PCU communication interface
  - MCU Abstraction
- MCU + PCU + simple model of selected ECU components
  - ECU Abstraction
- PCU standalone
  - Test case development: SystemC model for VirtualAte
  - System-level concept validation: Verilog-AMS



Source: www.autosar.org

Heterogeneous Model Structure

MCU
- Synopsys SCML
- No source available

PCU
- OSCI SystemC(-AMS)

PCU
- Verilog-AMS

VirtualATE
- C++
- SystemC

PCU Testbench
- UVM-SystemC

PCU Testbench
- SystemVerilog
- Verilog-AMS

Outline

Scenario 1: MCU – PCU coupling

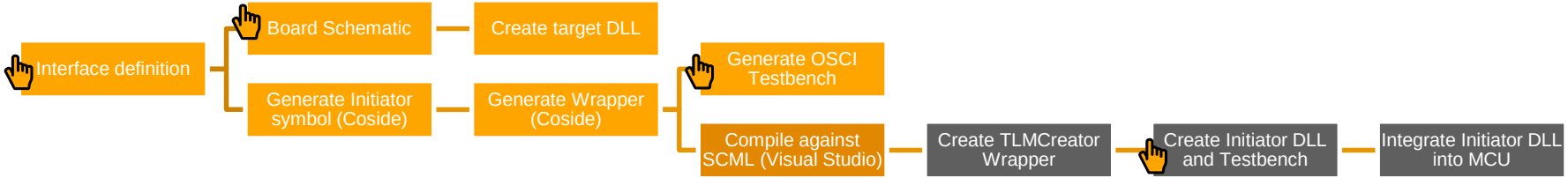
- Software development for combined MCU PCU system
- Synopsys proprietary model needs to talk to our own models

Coside Virtualizer Export	SC2SC coupling
Currently only on linux platform	Only on windows platform
Full visibility from Virtualizer	Not visible from Virtualizer
	Limited tool support

Scenario 1: SC2SC coupling

- Library available in Coside.
- Variants used in Continental since ~10 years.
- Build two Dynamic Linked Libraries (DLLs)
  - First with the external OSCI components build and compiled by Coside (Coupling target).
  - Second with Synopsys TLM Creator against SCML library to be integrated by the supplier into MCU model. Instantiates the coupling initiator.

Scenario1:SC2SC coupling



Scenario 1: Summary

- Initial creation of board and initiator takes some time.
- After an initial setup of initiator and board is available, different boards with different functionality can be created very quickly.
- Board stays OSCI SystemC.
  - Ensures compatibility with other flows.
  - No exposure of board internals in Synopsys environment.
  - Tracing of internals can be configured by in house solution.

Outline

SystemC(-AMS) and Verilog(-AMS)

Verilog(-AMS)	SystemC(-AMS)
Mixed-Signal development	Software and Test development
Accurate on electrical level	Abstract electrical representation
Uses design RTL	TLM based
Compatible with supplier	Allows C debug
Slow (1 ms / minute)	Fast (1 s / minute)

Different use cases require different models. Models are developed independently against specification.

Scenario 2: SC model Verilog-AMS tests

- Align functionality of SystemC(-AMS) and Verilog(-AMS) models.
- Verilog models and tests are developed first.
- Verilog-AMS DUT contains complete electrical interface. Expected to be difficult to stimulate from SystemC test bench.

Sceanrio 2: SC model Verilog-AMS tests

Coside NCsim export → From created *.vams file create Cadence symbol → Replace Verilog-AMS DUT with SystemC model → Add WREAL Electrical converters → Configure *.irun.res → Lauch AMS Designer

Scenario 2: WREAL – Electrical converters

- Cadence can add these converter modules automatically
- Automatically added converters are often not accurate enough. E.g. absdelta function has to high tolerance
- Better create your own converters to be aware of the settings

Scenario 2: Results

Pass	Fail(VAMS)
Fail	Fail(VAMS)
Fail	Pass
Fail	Fail
Fail	Pass
Fail	Pass
Fail	Fail
Fail	Fail
Fail	Fail
Fail	Pass
Fail	Pass
Fail	Pass
Fail	Pass
Pass	Pass
Fail	Fail
Fail	Pass
Fail	Pass
Fail	Fail
	Fail(VAMS)
	Fail(VAMS)
	Fail(VAMS)
Fail	Fail
	Fail
Pass	Pass
Fail	Fail(VAMS)
Fail	Fail(VAMS)
Fail	Fail(VAMS)
Fail	Pass
Fail	Pass

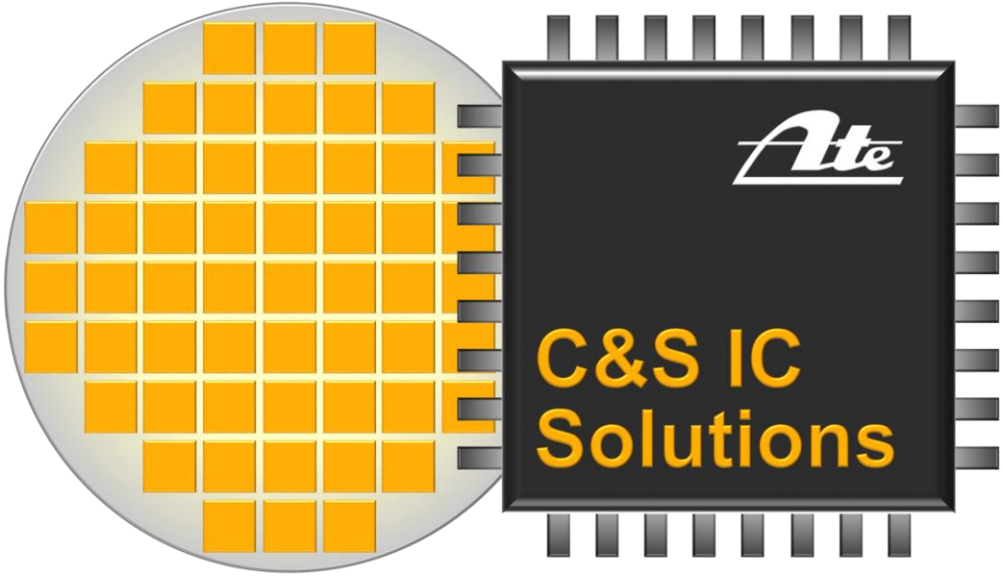
- All SystemC modules were extensively tested at module level before start of comparision
- Initially 23 out of 26 tests behaved differently on the SystemC model as they did on Verilog-AMS.
- Issues were caused by
  - SystemC model implementation
  - Specification issues
  - RTL and test issues
  - Connect Modules
- Significant improvement of model quality.

Outline

Conclusion

- Using models across different domains usually requires some additional initial effort.
- This effort pays off in
  - Better quality of
    - Specification
    - RTL Implementation
    - SystemC model
  - Support of use case that cross the border of an individual IC.

Thank you
for your attention!



ASIC solutions for ADAS

Safe and Dynamic Driving
towards Vision Zero

SensePlanAct