ENABLING THE SIMULATION OF COMPLEX, HETEROGENEOUS SYSTEMS USING THE FUNCTIONAL MOCKUP INTERFACE (FMI)

Hubertus Tummescheit

Modelon Inc & Modelica Association

June 2015

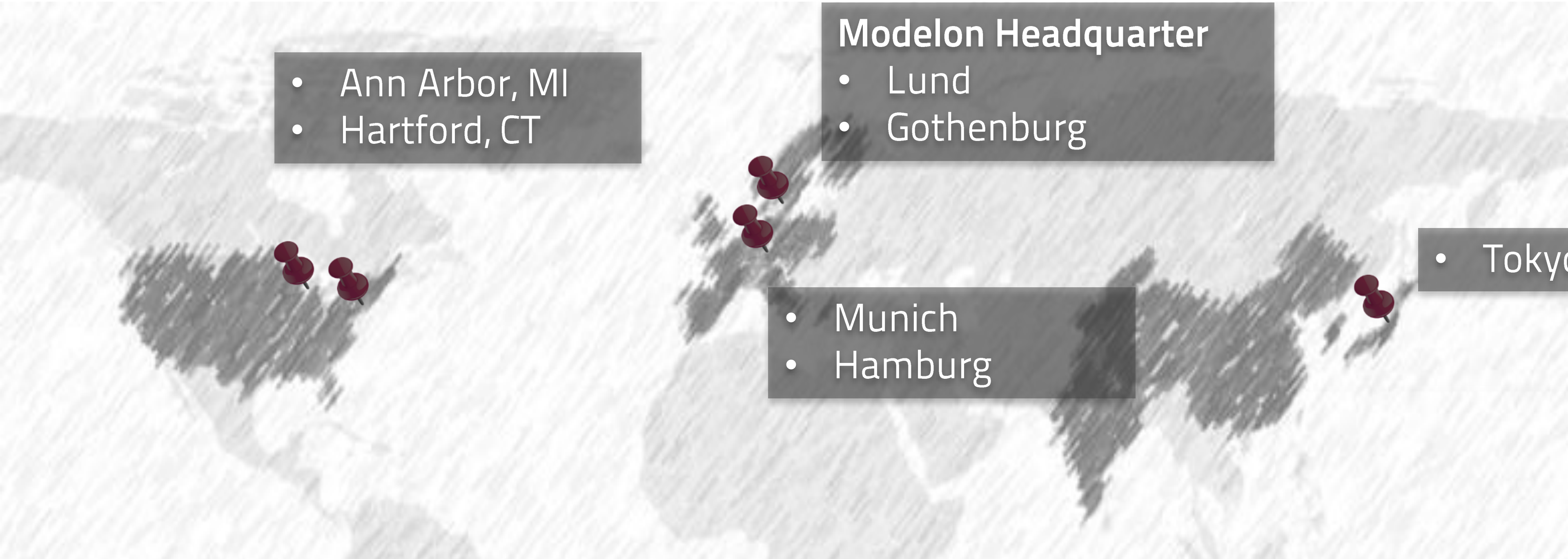
My Background

- MSc Mechanical Engineering Technical University Hamburg-Harburg, 1996
- Modelica Development since 1997
- PhD Automatic Control Lund University 2002 with Karl Åström
- UTRC East Hartford in 2002-2003
- Co-Founded Swedish Company Modelon AB in 2004
- CEO Modelon AB/Inc. Since 2004/2013
- Member of Modelica Association Board and FMI Steering Group

OVERVIEW

- Motivation
- Use Cases
- What is the Functional-Mockup-Interface (FMI)?
- The Business Case for FMI
- Using FMI to improve design flows in the SE 'V'
- Examples for FMI use in advanced design flows
  - Model Deployment
  - Early Validation for automotive controls
  - Real-time simulation on multiple cores
  - High-Fidelity ABS Brake validation

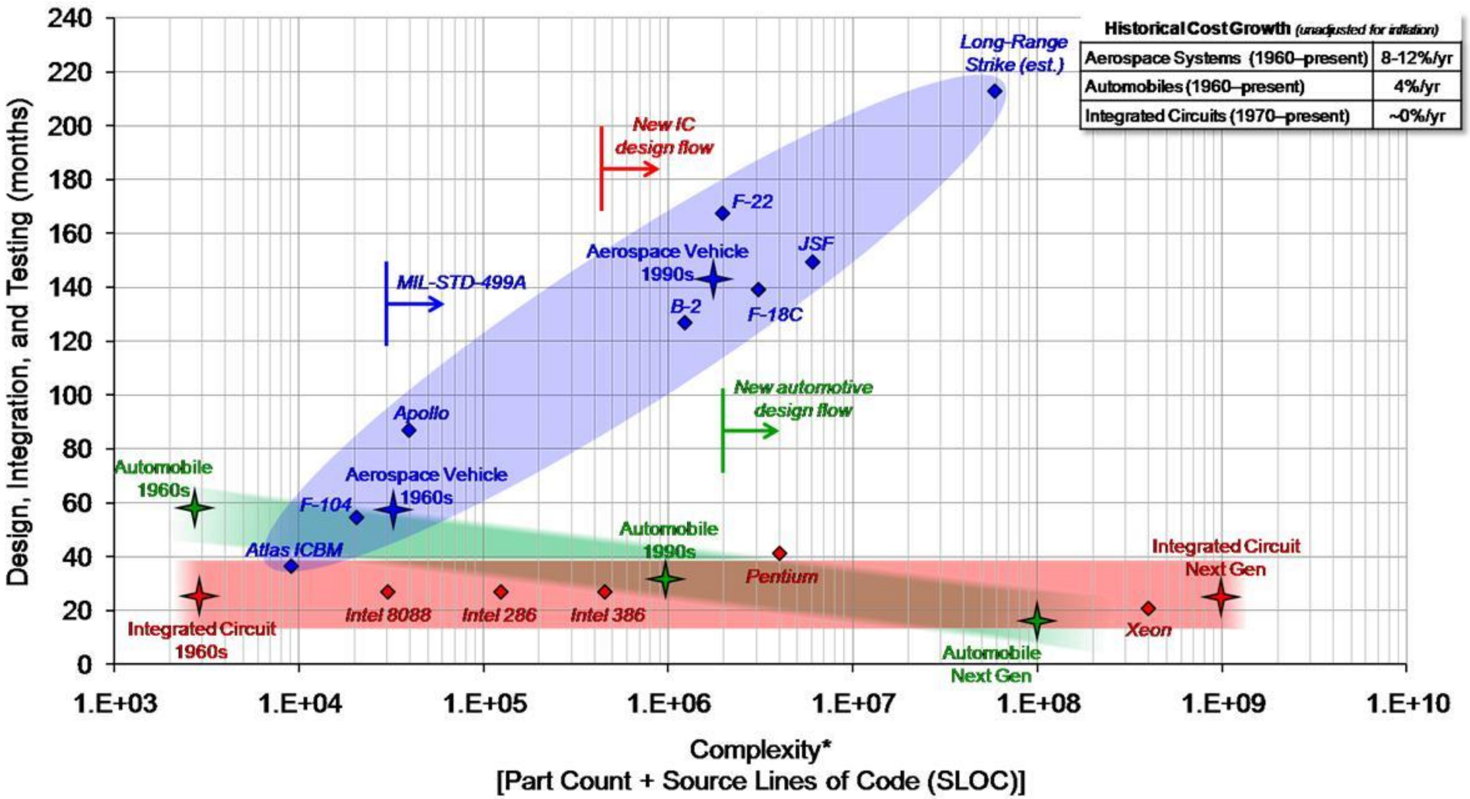
Modelon Overview & Locations



- **Global premier provider of Modelica and FMI based solutions for model based systems development**
- ~50 engineers (MSc / PhD levels) dedicated to Modelica and FMI
- Global customers mostly among Fortune 500 technology companies in Automotive, Aerospace, Energy and Industrial Equipment
- About 30% annual organic growth

THE COMPLEXITY ISSUE

- System complexity increases while
- Required time to market decreases (most industries)
- Without disruptive changes, an impossible equation to solve.

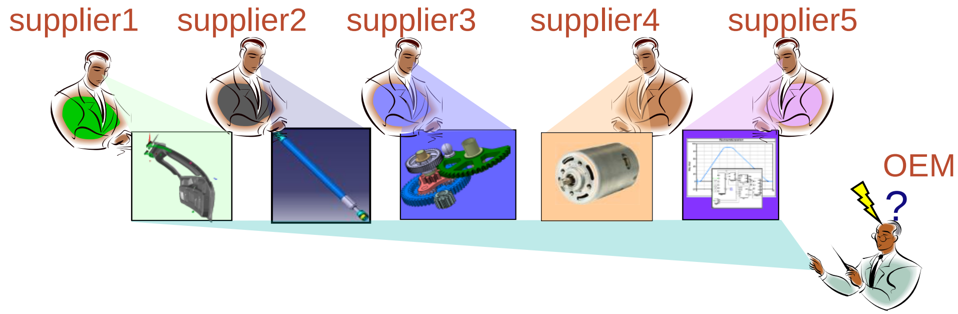


Source: DARPA AVM pres.

1. WHY FMI?

Problem

- Due to **different applications**, **models** of a system often have to be developed using **different programs** (modeling and simulation environments).



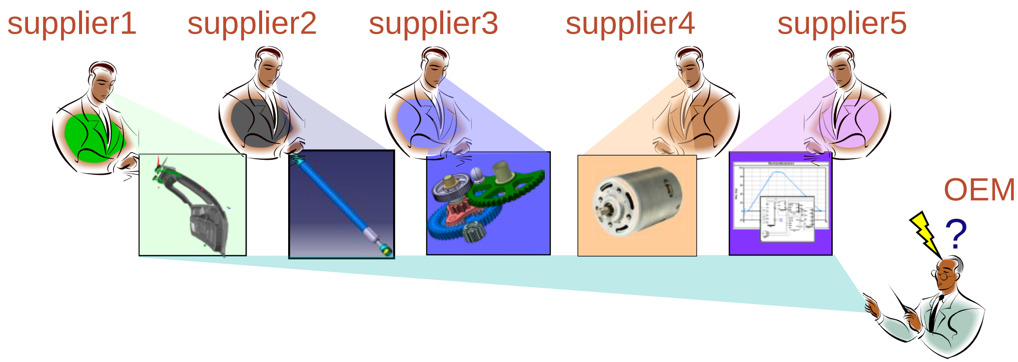
- In order to **simulate** the system, the different programs must somehow interact with each other.

- The system integrator must cope with simulation environments from many suppliers.

- This makes the **model exchange** a necessity. No current standardized interface.

- Even though **Modelica®** is tool independent, it cannot be used as such a standardized interface for model exchange.

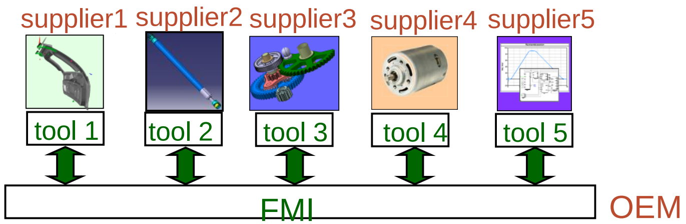
USE CASE I:



Combined simulation for system integration

Solution

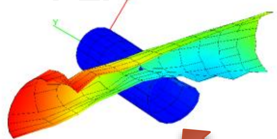
- As a universal solution to this problem the **Functional Mockup Interface (FMI)** was developed by the EU-project MODELISAR, and is now maintained by the Modelica® Association



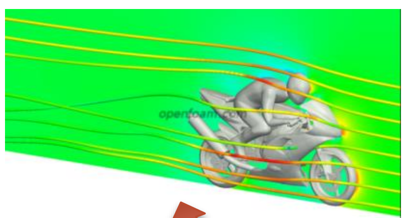
USE CASE II:

- Combine different modeling formats into coherent co-simulation (cyber-physical systems)
  - Physical models, 1D-3D
  - Controls / Software

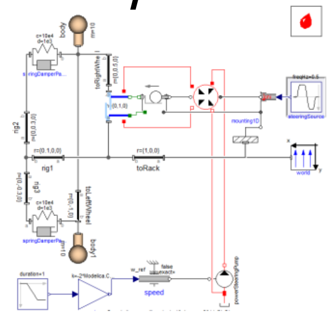
FEA



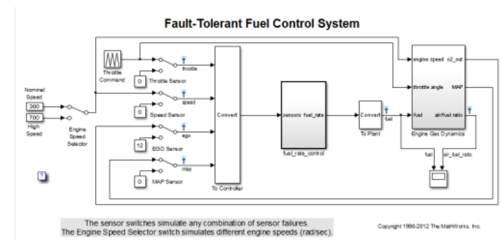
CFD



Modelica / 1-D Systems Simulation



Block Diagrams Controls



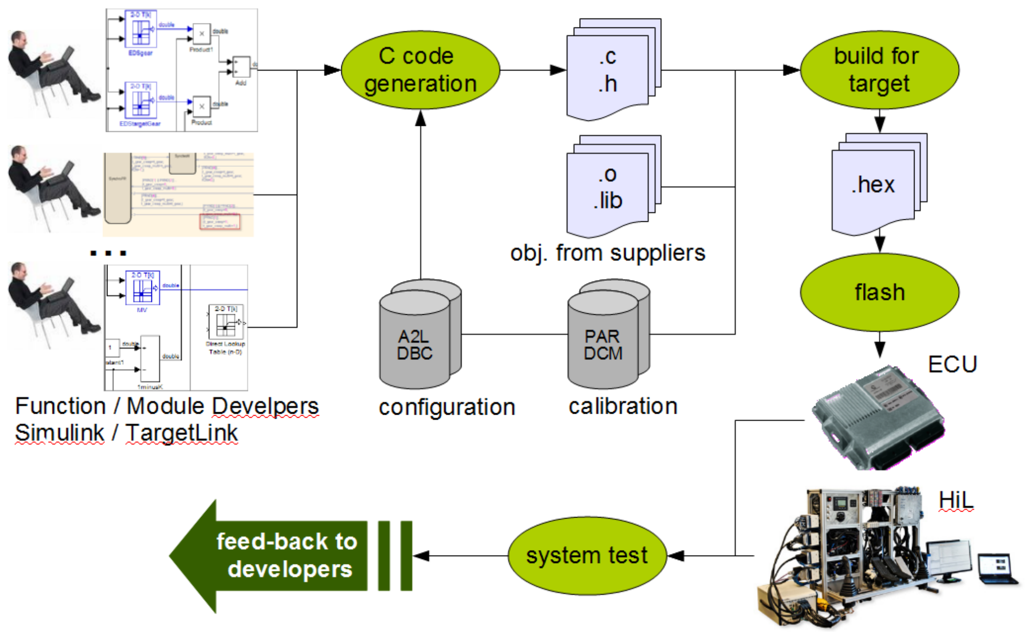
FMI-based System Simulation

USE CASE III: FMI FOR SIL AND HIL

- FMI export support from Controls Tools:
  - Simulink through FMIT Coder (Modelon)
  - Scade Suite (safety critical applications)

- FMI supported by most major HIL Vendors
  - DSPACE
  - National Instruments
  - Concurrent
  - IPG
  - Speedgoat

- FMI for ECU virtualization
  - Silver by Qtronic
  - ETAS tools (Bosch)

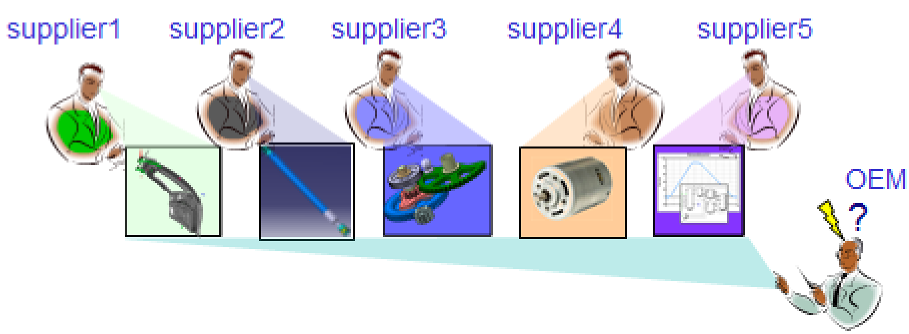


FUNCTIONAL MOCKUP INTERFACE (FMI)

- Tool independent standard to support both model exchange and co-simulation of dynamic models
- Original development of standard part of EU-funded MODELISAR project led and initiated by Daimler
- First version FMI 1.0 published in 2010
- FMI currently supported by more than 60 tools
  (see www.fmi-standard.org for most up to date list)
- Active development as Modelica® Association project
- FMI 2.0 released July 2014 and brings additional functionality to FMI standard

Problems / Needs

- Component development by supplier
- Integration by OEM
- Many different simulation tools



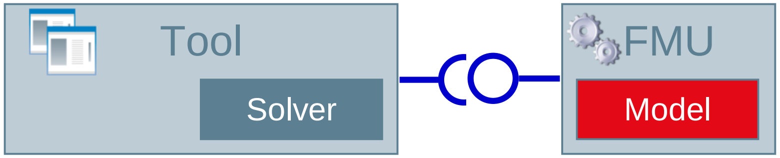
FMU: A MODEL WITH STANDARD INTERFACE

- A component which implements the FMI standard is called *Functional Mockup Unit (FMU)*
- Separation of
  - Description of interface data (XML file)
  - Functionality (C code or binary)
- A FMU is a zipped file (*.fmu) containing the XML description file and the implementation in source or binary form
- Additional data and functionality can be included
- Information & Interface specification: www.fmi-standard.org

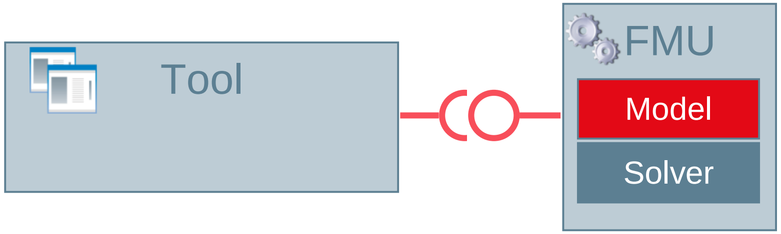
FMI FLAVORS

- The Functional Mock-up Interface (**FMI**) is a **tool independent standard** for

  - **Model Exchange (ME)**



  - **Co-Simulation (CS)**



- The FMI defines an interface to be implemented by an executable called Functional Mock-up Unit (**FMU**)

FMU=Model w/ Standard Interface

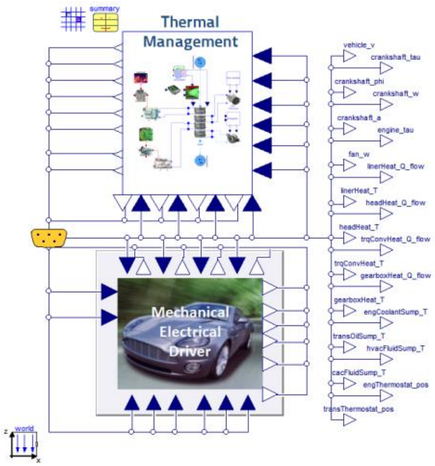
FMI: A BUSINESS MODEL INNOVATION

- FMI-compliant tools often allow liberally licensed export of models for distribution in the organisation and to partners
- Exported FMUs most often don't require a license from the model authoring tool
- Deployment from few simulation specialists to designers, domain specialists, control engineers
  - One FMU used by many engineers (control design)
  - One FMU run on many cores (robust design)

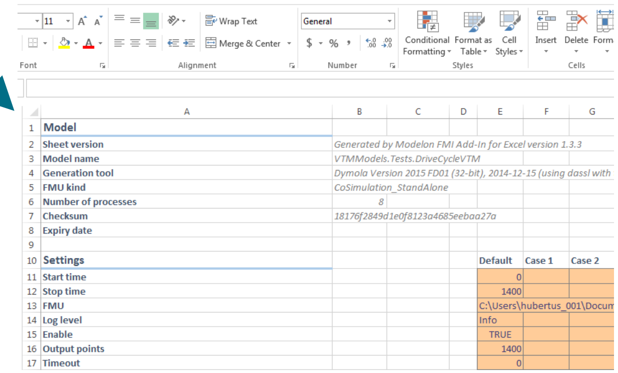
FMI: A BUSINESS MODEL INNOVATION

1. Separate the model authoring tool from the model execution tool !
2. Free the model unit (FMU) from license restrictions
3. Make the standard widely accepted: https://fmi-standard.org/tools

TYPICAL FMI-BASED WORKFLOWS



Export: exported
FMU freely licensed



	A	B	C	D	E	F	G
1	**Model**						
2	Sheet version	*Generated by Modelon FMI Add-In for Excel version 1.3.3*					
3	Model name	*VTMModels.Tests.DriveCycleVTM*					
4	Generation tool	*Dymola Version 2015 FD01 (32-bit), 2014-12-15 (using dassl with*					
5	FMU kind	*CoSimulation_StandAlone*					
6	Number of processes	*8*					
7	Checksum	*18176f2849d1e0f8123a4685eebaa27a*					
8	Expiry date						
9							
10	**Settings**				Default	Case 1	Case 2
11	Start time				0		
12	Stop time				1400		
13	FMU				C:\Users\hubertus_001\Docum		
14	Log level				Info		
15	Enable				TRUE		
16	Output points				1400		
17	Timeout				0		

Model Authoring Tool(s)

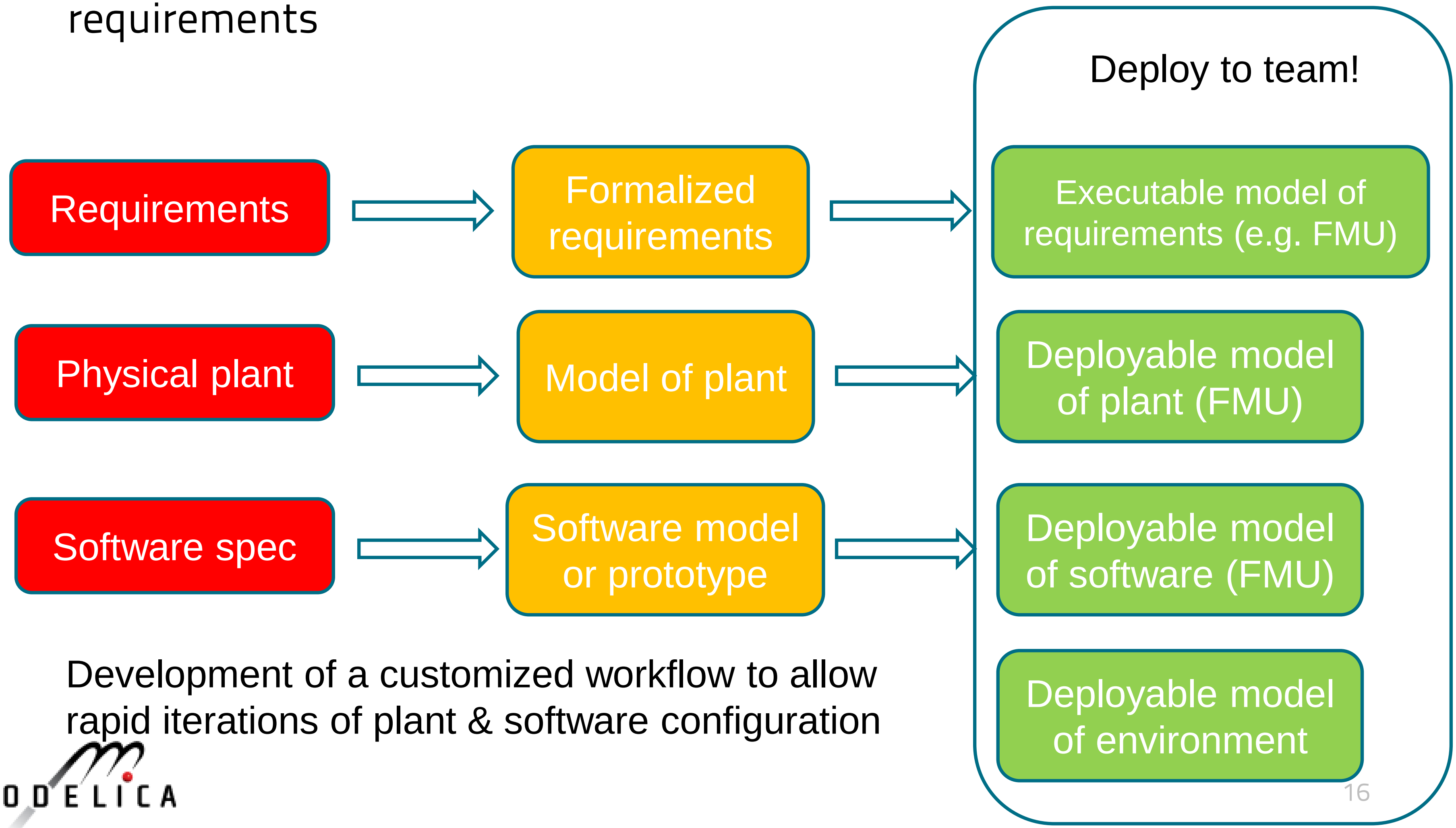
Low-cost Model Execution Platform
May combine FMUs from several tools

- Additional work flow automation for
  - pre-processing,
  - model calibration,
  - post-processing,
  - analysis,
  - automated reporting
  - automated requirements verification

- **True democratization of simulation**
- **Greatly improved utilization of models**

AUTOMATED REQUIREMENTS VERIFICATION

- Systems Engineering centric FMI-based workflow example: automated requirements verification for hardware and software requirements

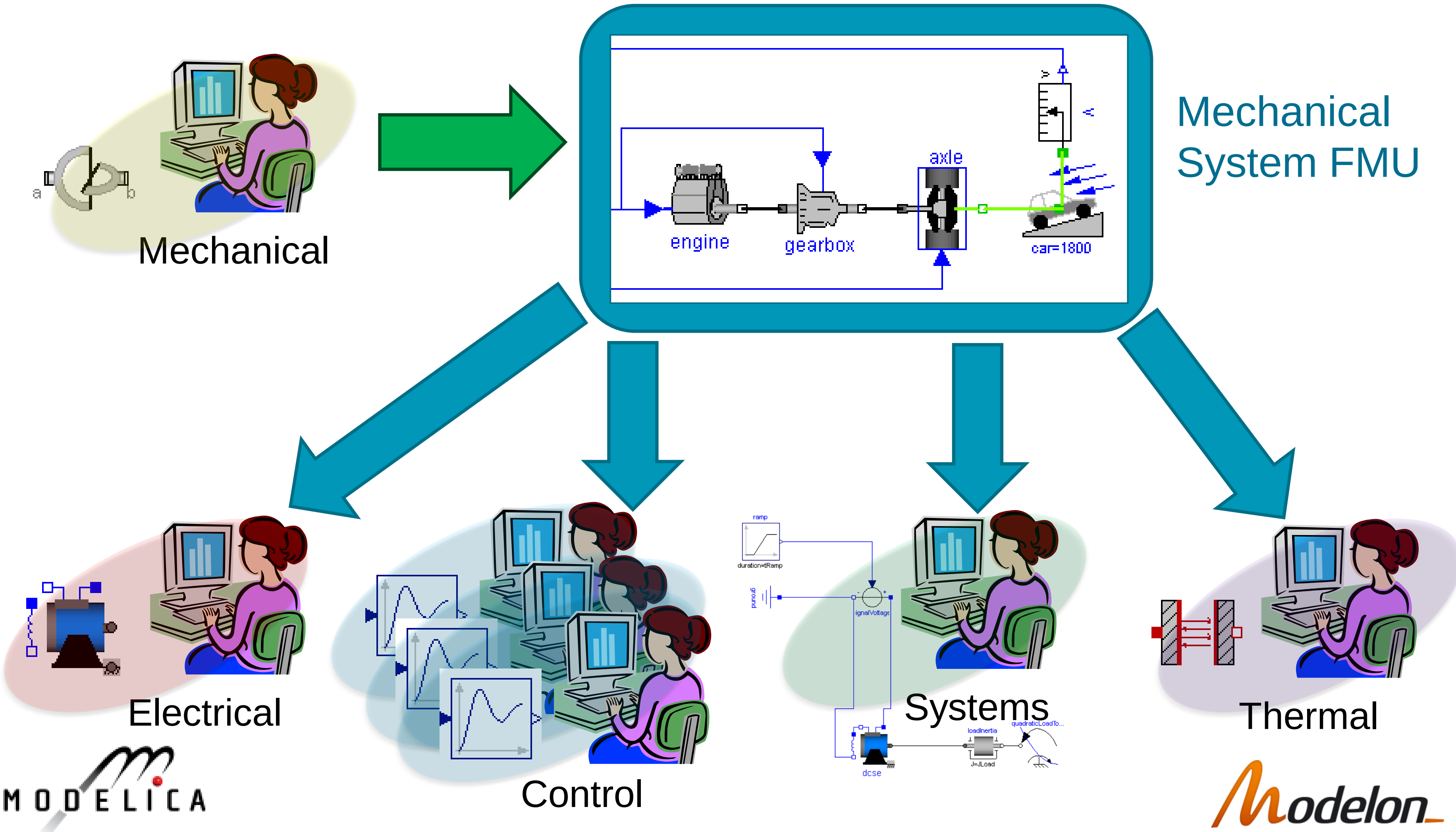


Development of a customized workflow to allow rapid iterations of plant & software configuration

MODELICA

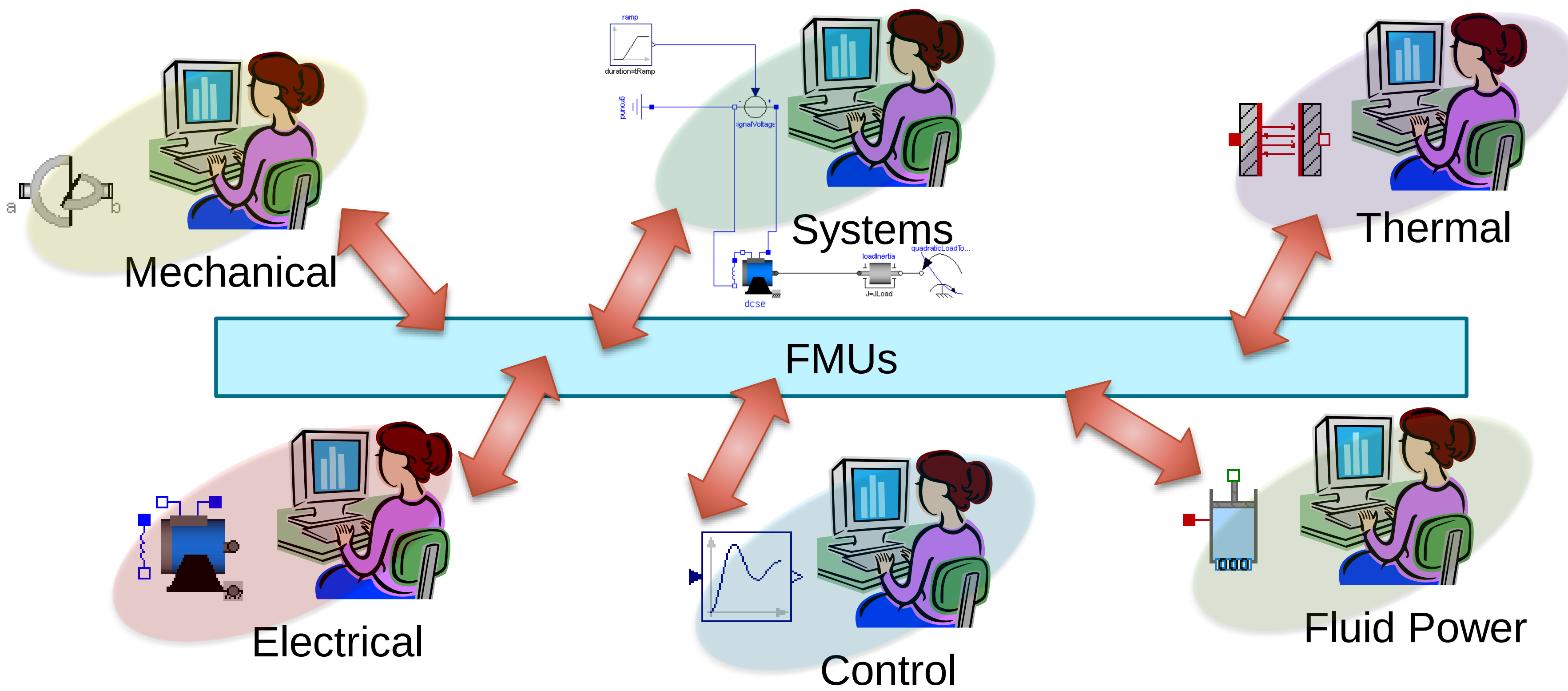
MODEL DEPLOYMENT

- FMU deployed (native tool) to support multiple applications

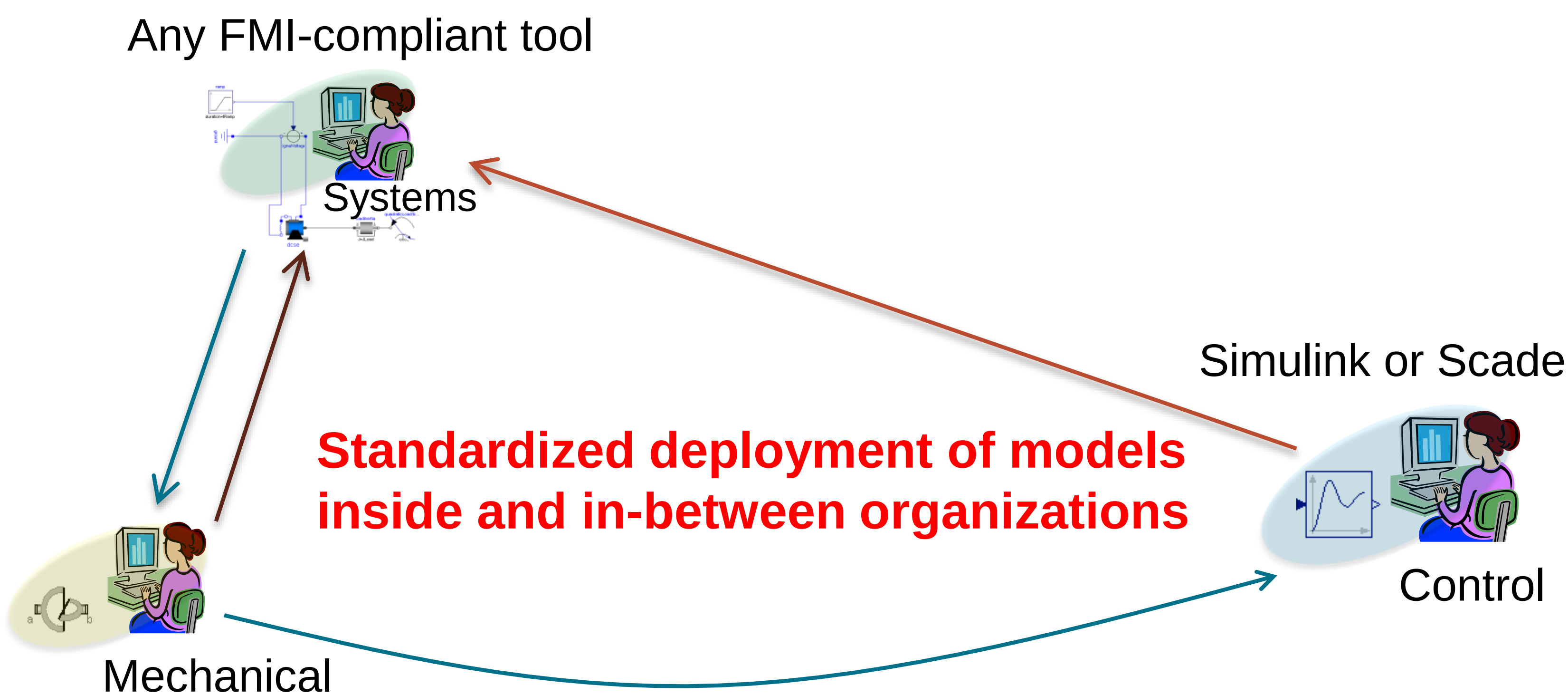


MULTIDOMAN COLLABORATION

- Engineers in different domains work **with FMUs**
  - Share models, distributed collaboration, work in tool of choice, reduced license costs, protect IP, couple carefully ! !



ENTERPRISE MODEL DEPLOYMENT

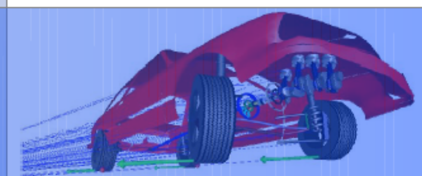


Standardized deployment of models inside and in-between organizations

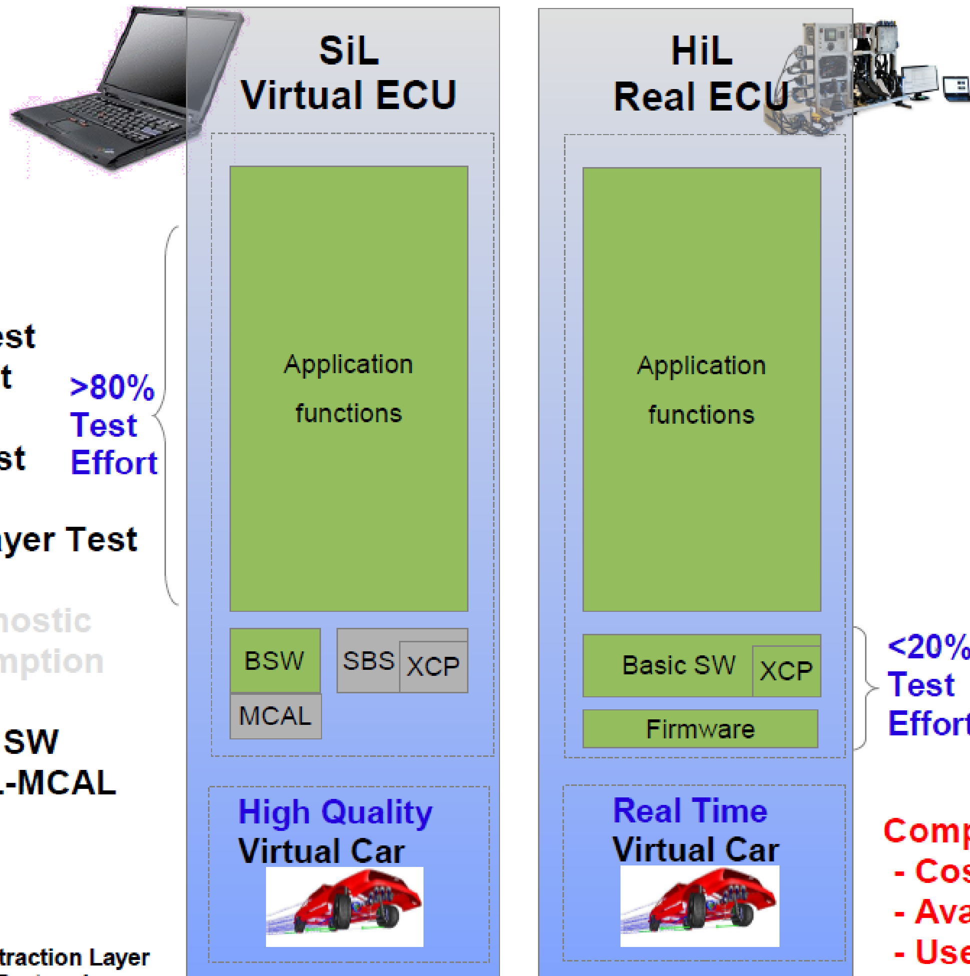
"Daimer, QTronic and Vector describe how Mercedes-Benz currently uses virtual ECUs to validate transmission control software for about 200 variants of the Sprinter series in a highly automated way on Windows PC"

TYPES AND COST OF TESTING PLATFORMS

		Early Cheap Many	Late Expensive Few
		Virtual ECU	**Physical ECU**
Early Cheap Many	**Virtual Plant**	**SiL**	**HiL**
Late Expensive Few	**Physical Plant**	**Rapid Control Prototyping**	**Prototype**



HIL COMPARED TO SIL



Tasks:
- Adaptation
- (pre-)Tuning
- Single-ECU Test
- Multi-ECU Test
- Module Test
- Integration Test
- DEM Test
- Application Layer Test
- CAN Load
- HW Test/Diagnostic
- Power Consumption

>80% Test Effort

- original Basic SW runnable if SiL-MCAL is used

Tasks:
- Adaptation
- (pre-)Tuning
- Single-ECU Test
- Multi-ECU Test
- Module Test
- Integration Test
- DEM Test
- Application Layer Test
- CAN Load
- HW Test/Diagnostic
- Power Consumption

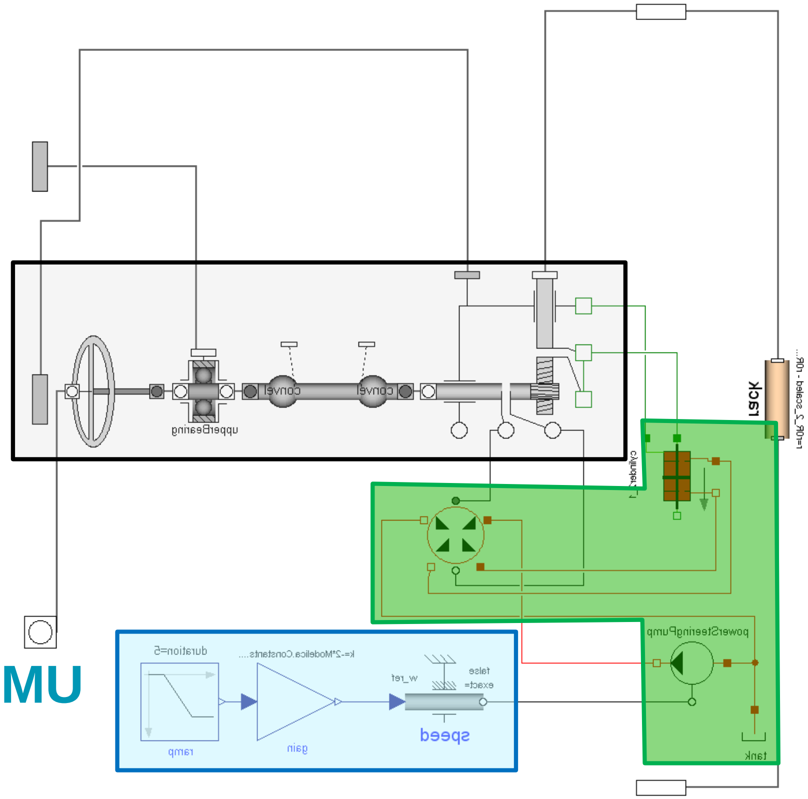
<20% Test Effort

Compare:
- Cost
- Availability (when/how often)
- User Training
- Feedback Time

Legend:
BSW: Basic Software
MCAL: MicroController Abstraction Layer
XCP: eXtended Calibration Protocol
SBS: Silver Basic Software
DEM: Diagnostic and Event Management

REUSABILITY

- Reusable models ~~in standard Modelica language~~ as FMUs
  - Compiled models generated internally, from suppliers, from partners, etc.
  - Protect IP as required
  - Many more tools can participate than just Modelica® tools



Mechanical FMU

Hydraulics FMU

Control/Actuation FMU

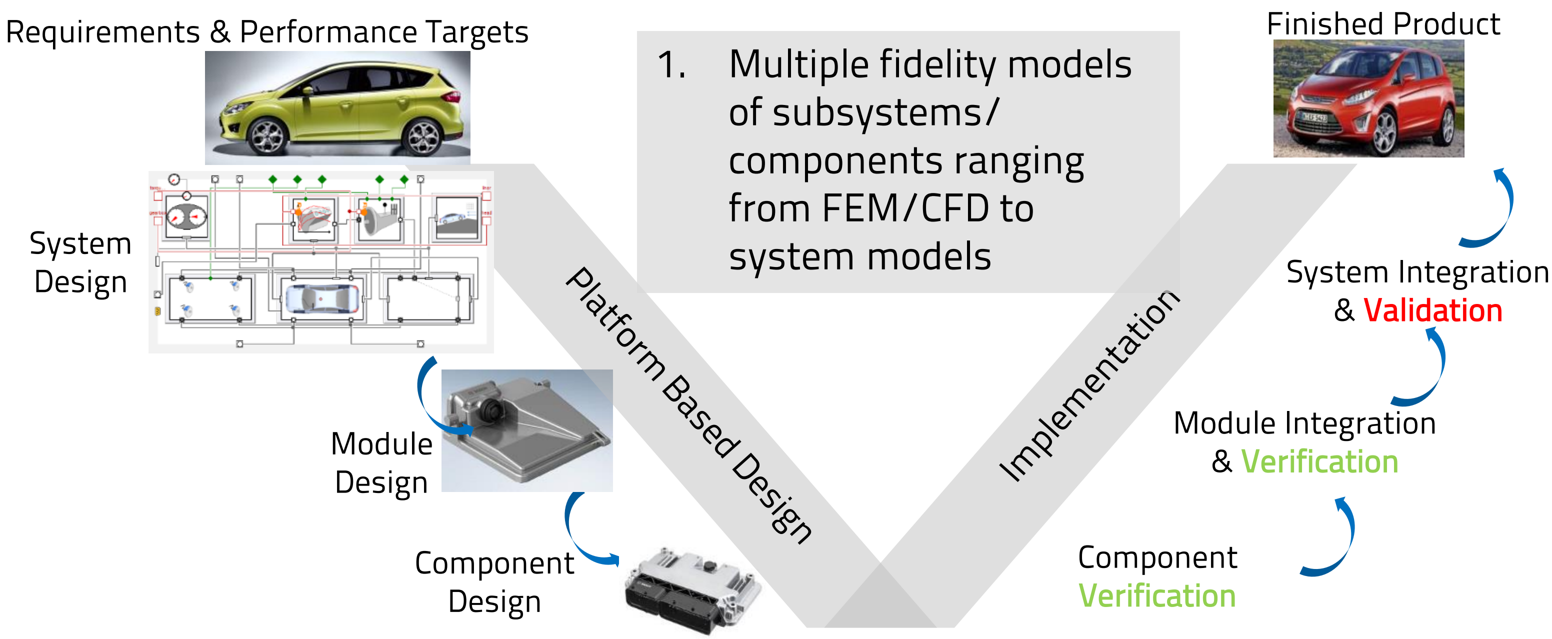
NEW OPTIONS FOR PROCESS INNOVATION

- Availability of cost-effective interfaces between tools creates new opportunities for integrated work flows
- FMI overcomes the N^2-problem: reduced number and cost of interfacing
- How does this fit into the traditional Systems engineering design view?
- Example from the automotive industry

SYSTEM ENGINEERING WITH FMI-BASED WORKFLOWS

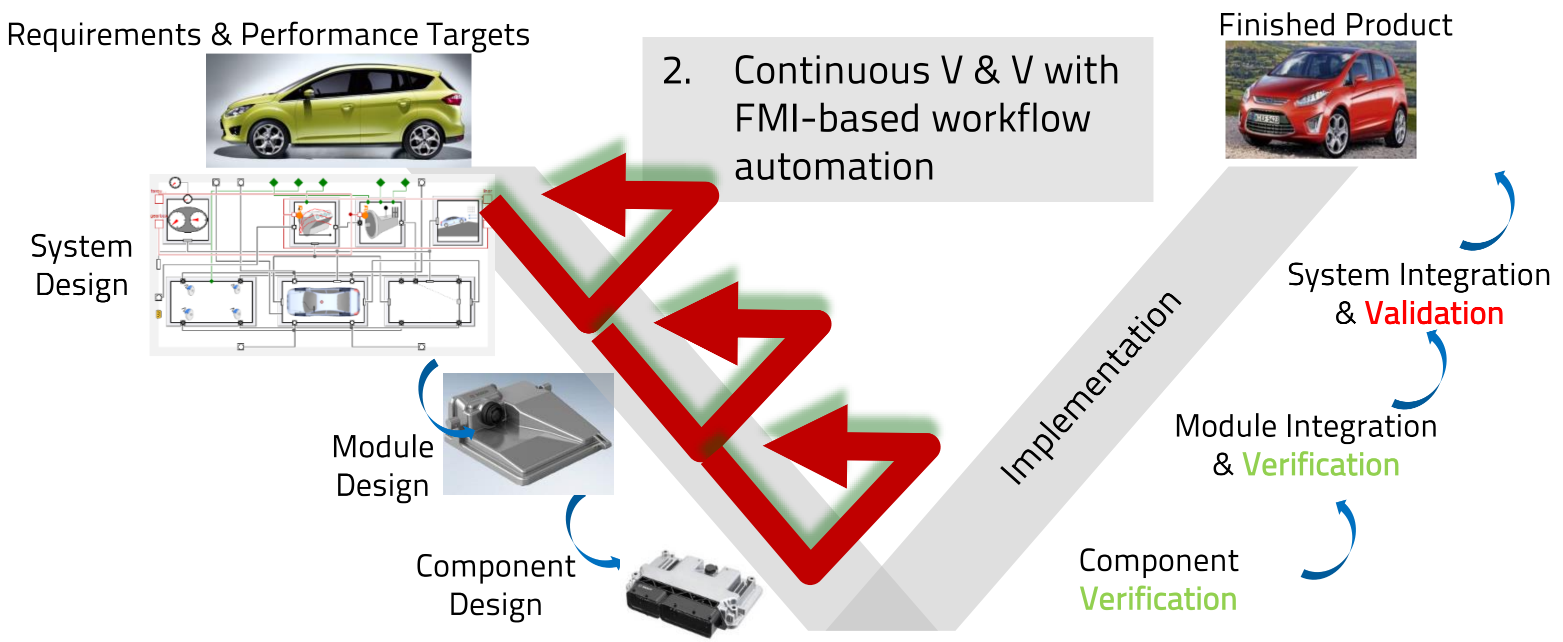
Multi-phase V development with Modelica & FMI

1. Multiple fidelity models of subsystems/ components ranging from FEM/CFD to system models



SYSTEM ENGINEERING WITH FMI-BASED WORKFLOWS

Multi-phase V development with Modelica & FMI

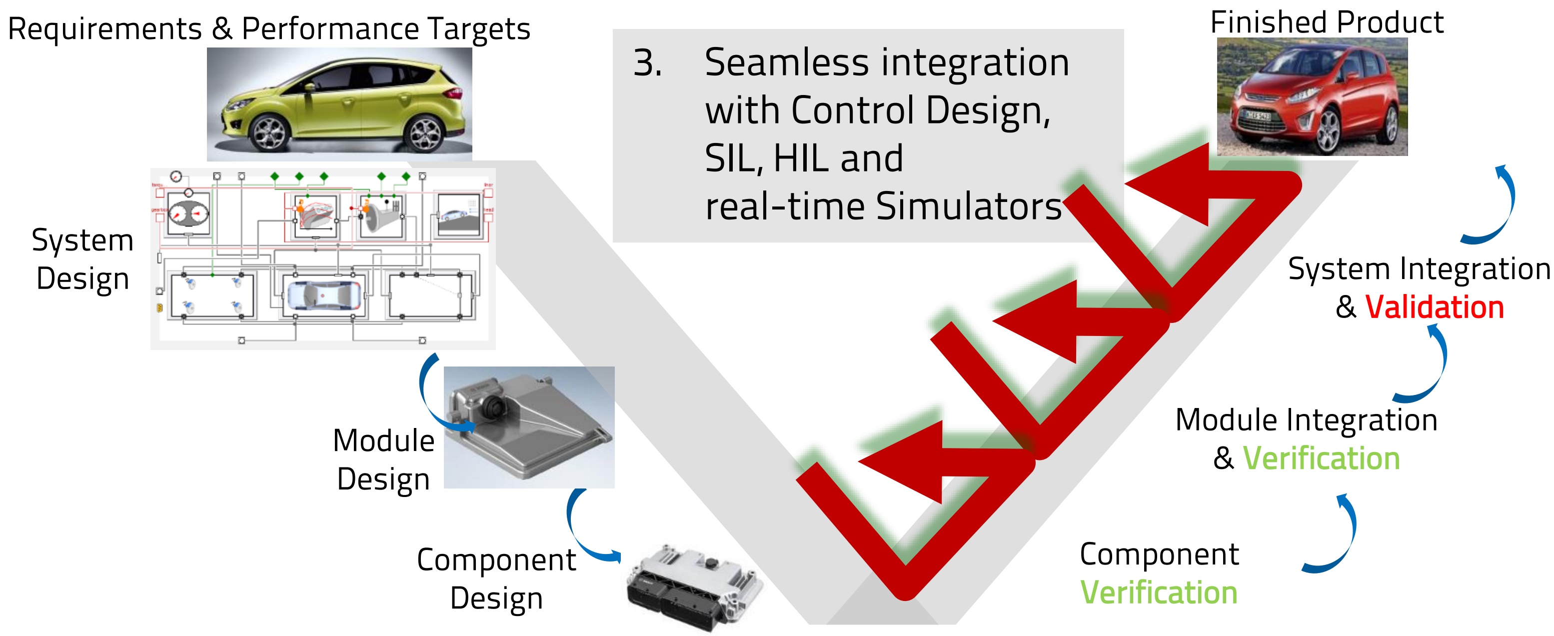


MODELICA

Modelon

SYSTEM ENGINEERING WITH FMI-BASED WORKFLOWS

Multi-phase V development with Modelica & FMI

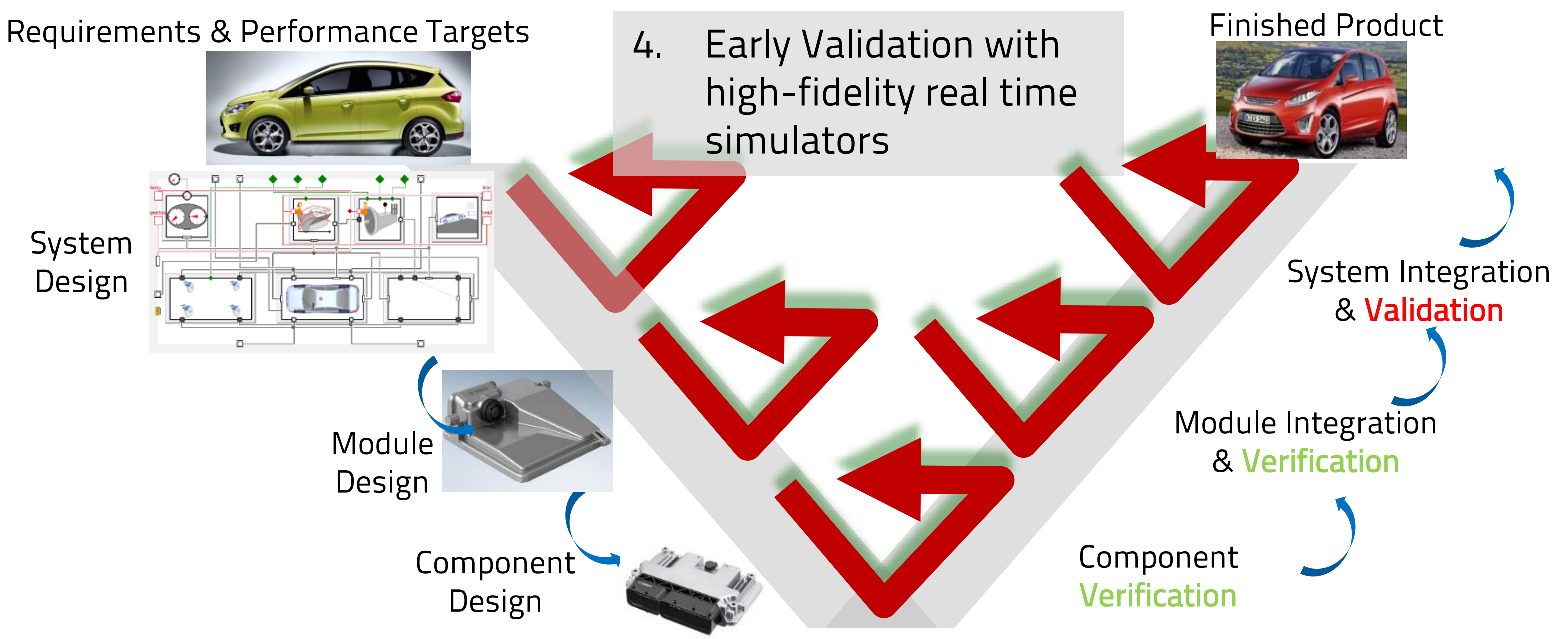


MODELICA

Modelon

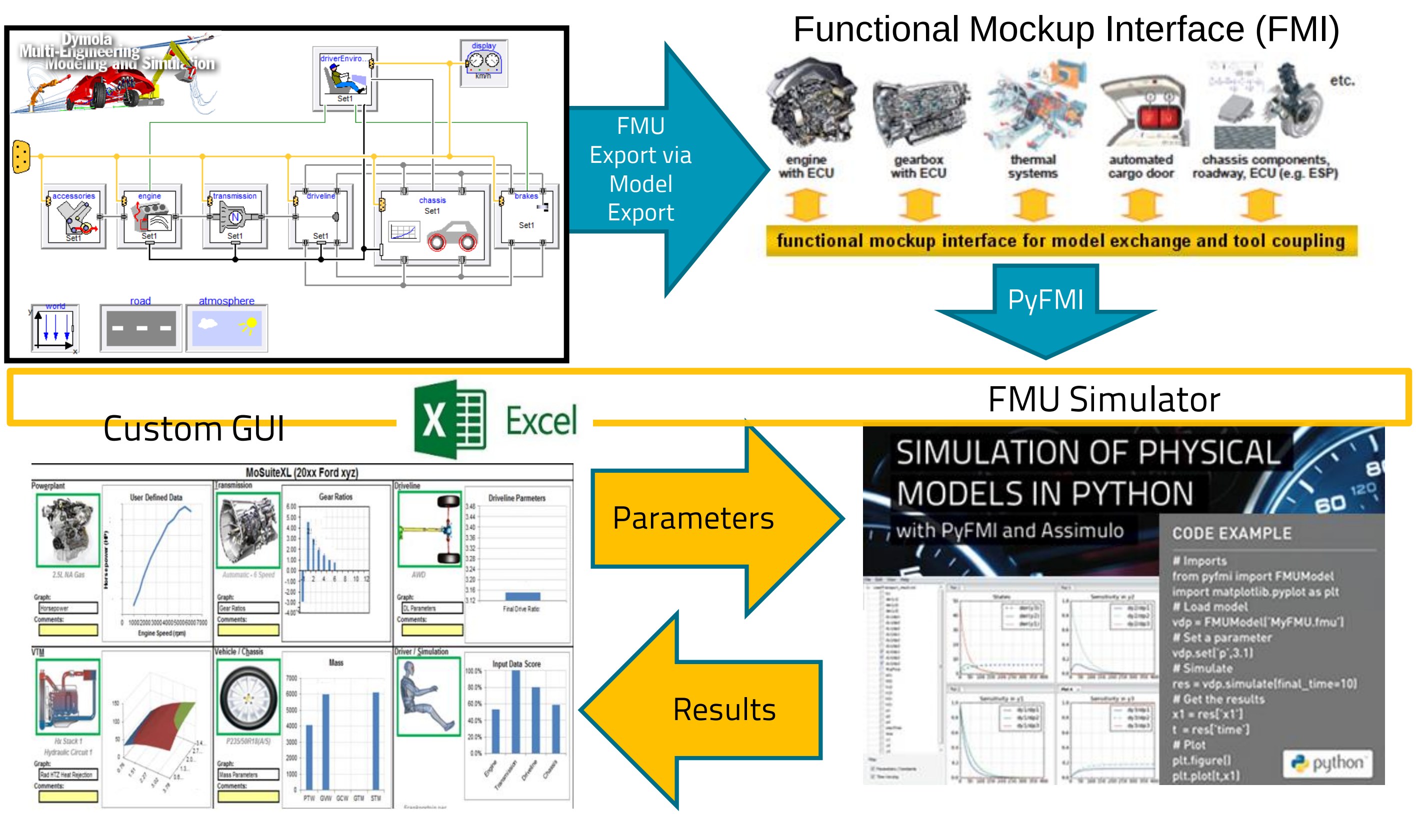
SYSTEM ENGINEERING WITH FMI-BASED WORKFLOWS

Multi-phase V development with Modelica & FMI

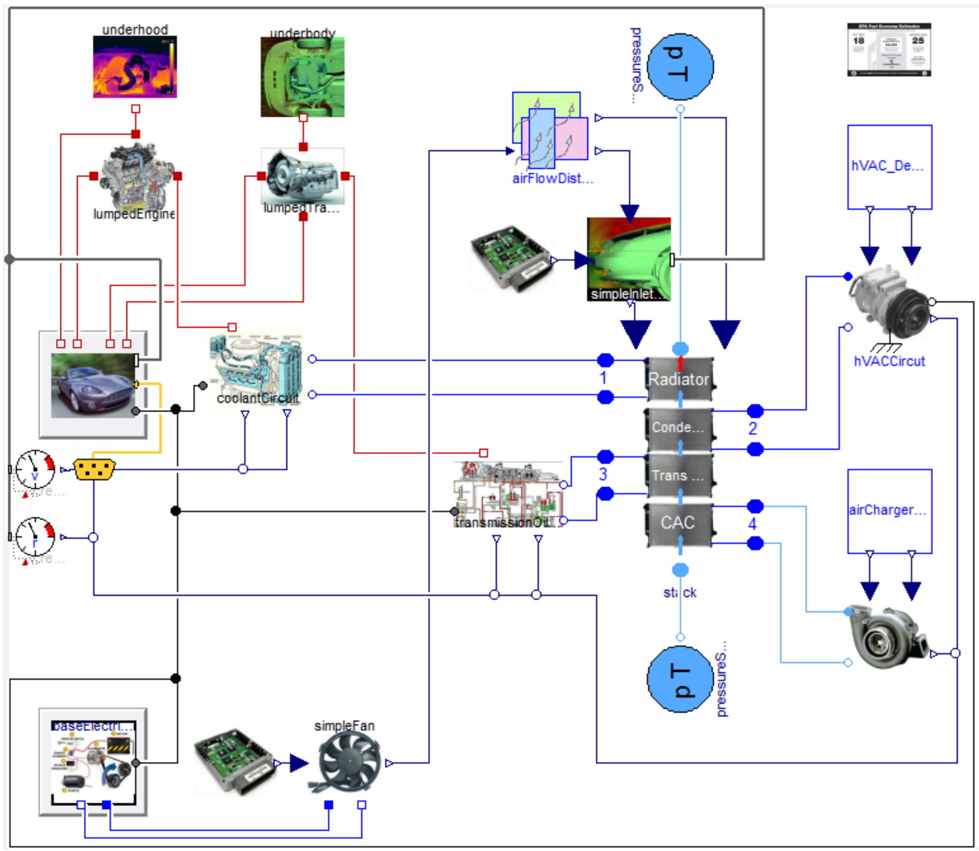


Use Case Examples

DEVELOPMENT TO DEPLOYMENT



VTM SYSTEM MODEL ARCHITECTURE



- Thermal
- Mechanical
- Electrical
- Coolant

Radiator		
HX	Air	Fluid
Tin [degC]	0.0	0.0
Tout [degC]	0.0	0.0
ṁ [kg/s]	0.0	0.0
Q [W]	0.0	

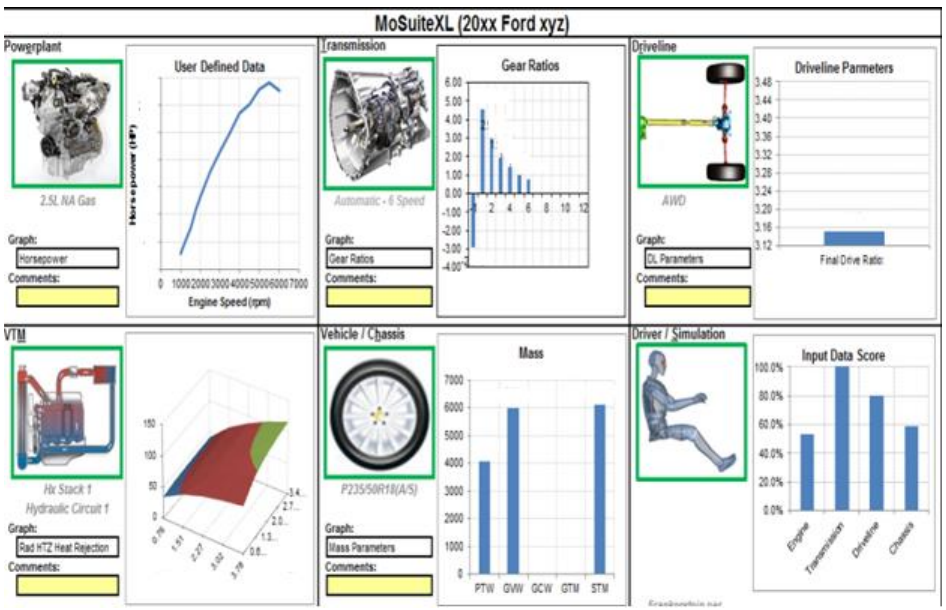
Condenser		
HX	Air	Fluid
Tin [degC]	0.0	0.0
Tout [degC]	0.0	0.0
ṁ [kg/s]	0.0	0.0
Q [W]	0.0	

Trans Oil Cooler		
HX	Air	Fluid
Tin [degC]	0.0	0.0
Tout [degC]	0.0	0.0
ṁ [kg/s]	0.0	0.0
Q [W]	0.0	

CAC		
HX	Air	Fluid
Tin [degC]	0.0	0.0
Tout [degC]	0.0	0.0
ṁ [kg/s]	0.0	0.0
Q [W]	0.0	

CUSTOM USER GUI

Excel



- Intuitive and highly customizable design
- Runs powerful physical models through FMI
- Batch Runs/DOE support
- Access to component database
- Small model foot-print
- Integrated analysis and export of results

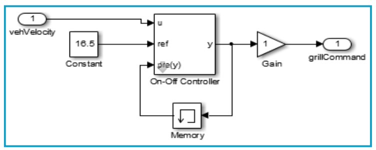
Integrates model configuration, parameterization, simulation, and post-processing into custom front end for end-user deployment

MODELICA

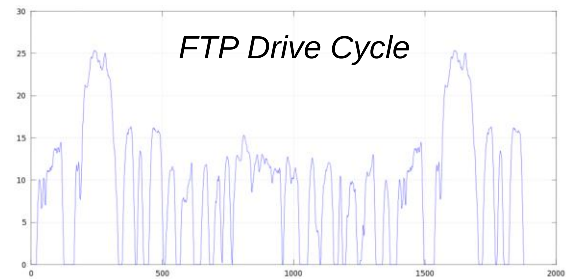
Modelon

CONTROLS DESIGN: ACTIVE SHUTTERS

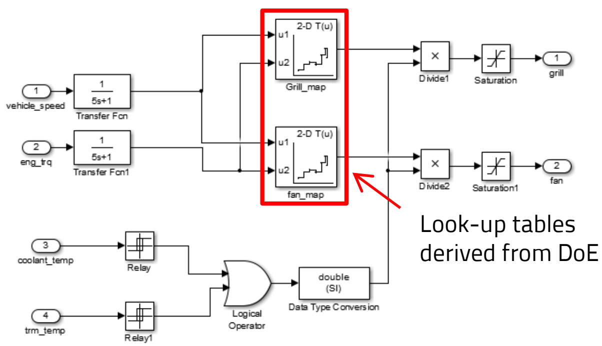
- DOE results for improved controller design: sample application



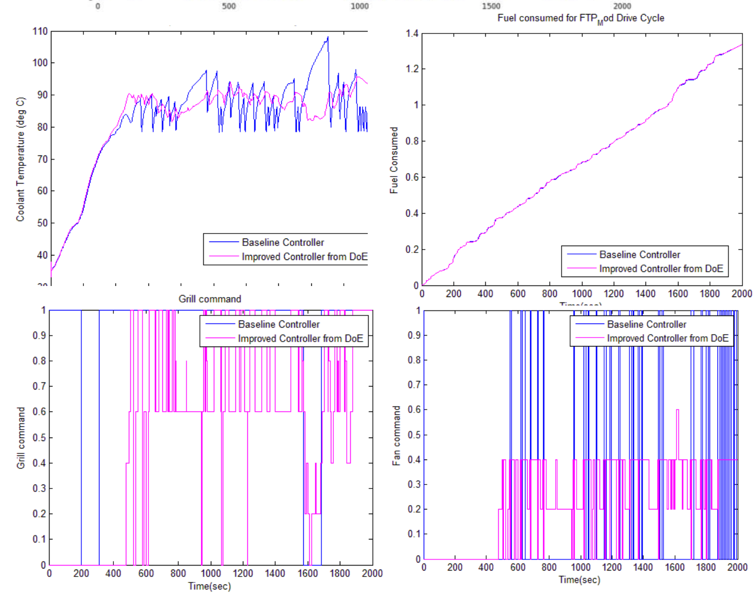
Mechanical shutter shuts the grill when speed is above 18 m/sec.



Grill position controlled as functions of vehicle speed, engine torque, coolant temp, and oil temp.



Look-up tables derived from DoE



Real time & use of Multiple cores

Encapsulated in the FMI-interface

TECHNICAL BACKGROUND

- Real-time applications
- Fixed step solvers: assurance of upper limit of computation time
- Explicit is fast but requires smaller step size, heavily limiting system stiffness
- Implicit is slow but stable and handles stiffness
- Leads to large nonlinear systems of equations
- Expensive (typically $O(n^3)$)
- Inlining of implicit solvers can reduce simulation time somewhat

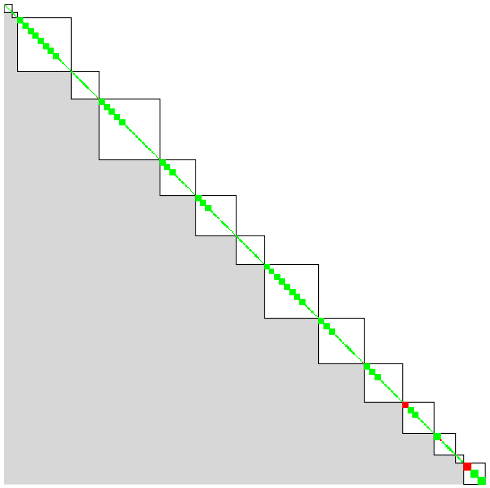
REAL TIME TUNING IN MODELICA TOOLS

- Some Modelica tools use best-in-class and state-of-the-art methods to achieve real time capability for complex models
  - Unique in combining symbolic and numeric methods
- Symbolic Methods
  - Time-discretize equations with most appropriate numeric scheme in as part of symbolic processing “**inlining**”.
  - Partitioning of equation systems
- Numeric Methods
  - Wide selection of fixed-step, low and high order, explicit and implicit integrators

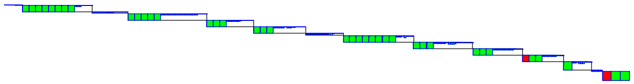
SCHEDULING- IDEA

- Partitioning needs to be made automatically
- Modelica gives good possibility to automatically partition the model equation execution into separate threads
- Equations are traditionally sorted in sequence, but
- ¨Some FMI-generators are now able to execute independent sections in parallel
- See: http://www.ep.liu.se/ecp/096/038/ecp14096038.pdf

The block lower triangle (BLT) is the result after standard Dymola manipulation, and illustrates the dependencies of the model equations.

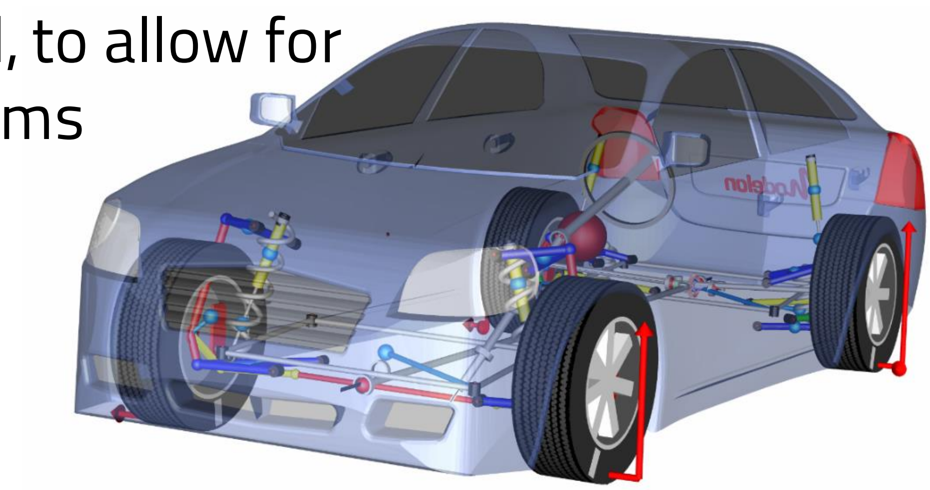


It can be compressed vertically to illustrate the parallel execution of blocks that don't need input from each other



VEHICLE EXAMPLE

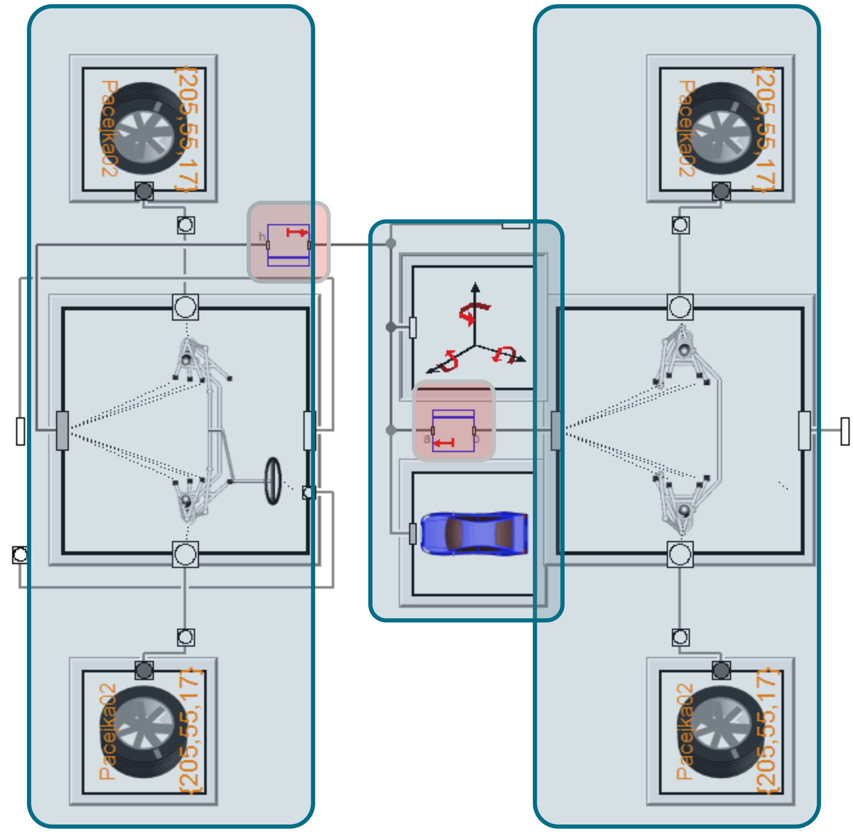
- 300+ states (150+ DOF)
- Defines current state-of-the-art for high accuracy RT vehicle dynamics: **high fidelity real time**
- Inlining with implicit Euler gives one large system {178}
- Highly coupled dynamical system
- Dymola RT automation is not enough on its own
- Modification of vehicle model, to allow for separated different subsystems



PARTITIONING OF MODELICA MODEL

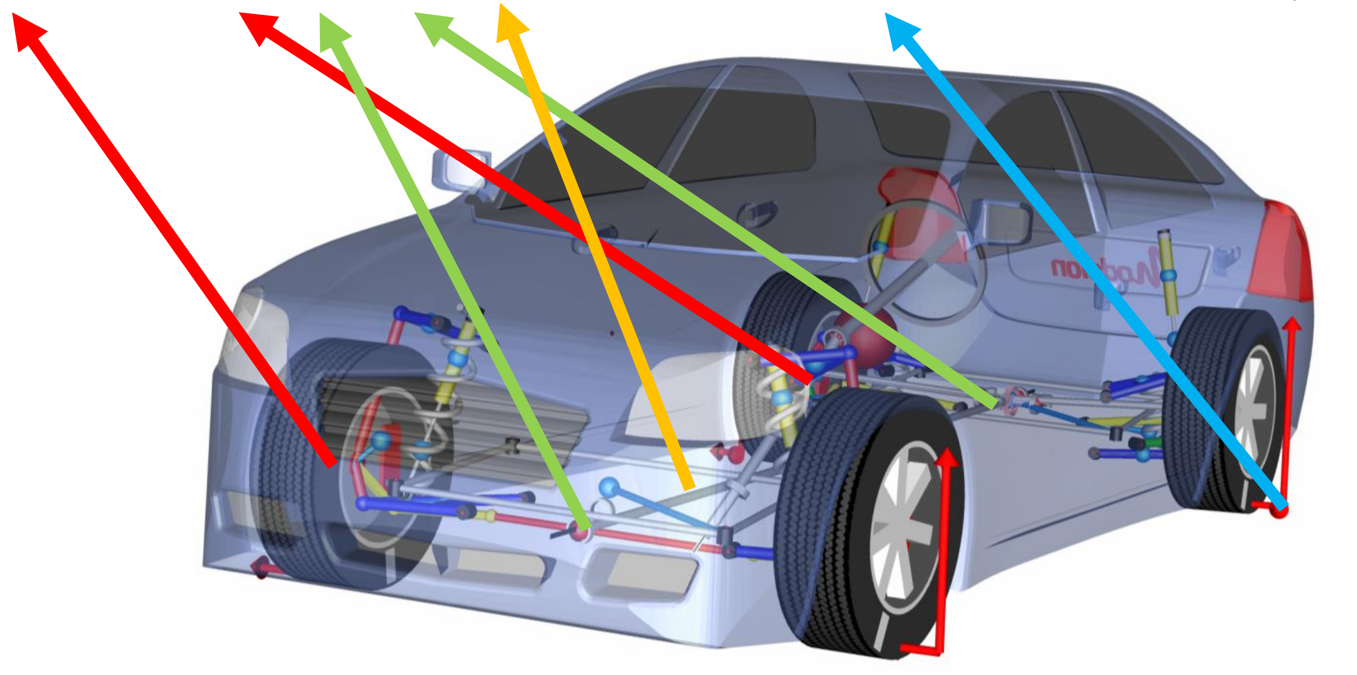
Front/rear separation

- Decouple to separate suspensions from body
- Resulting in 3 major sections
  - Front suspension and wheels
  - Rear suspension and wheels
  - Vehicle body
- Many more partitions further down, not shown here



RESULT

- Sizes of manipulated in-lined implicit integration systems:
- {25, 25, 40, 40, 20, 12, 14, 1, 1, 1, 1, 1, 1, 1, 1}



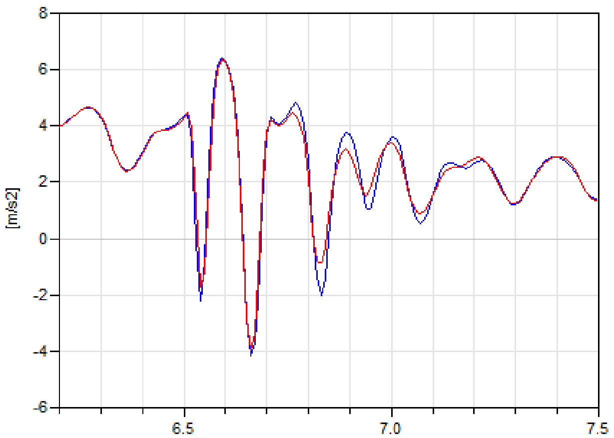
POTENTIAL EFFECT (THEORETICAL NUMBER)

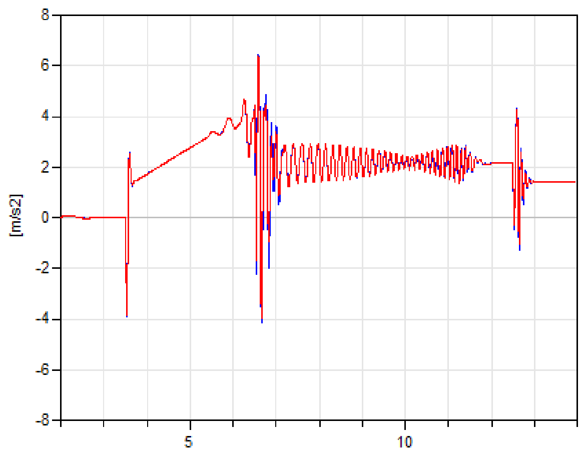
- Solving Systems of equations is "O(n^3)"
  - $$\frac{178^3}{\sum\{25, 25, 40, 40, 20, 12, 14, 1, 1, 1, 1, 1, 1, 1, 1\} .\wedge 3} = 32.84080824550166$$
  - Theoretically, parallelization can speed up by at most that ratio

- Parallelization
  - Depending on number of cores, execution time is just slightly above execution time for the largest /most costly system

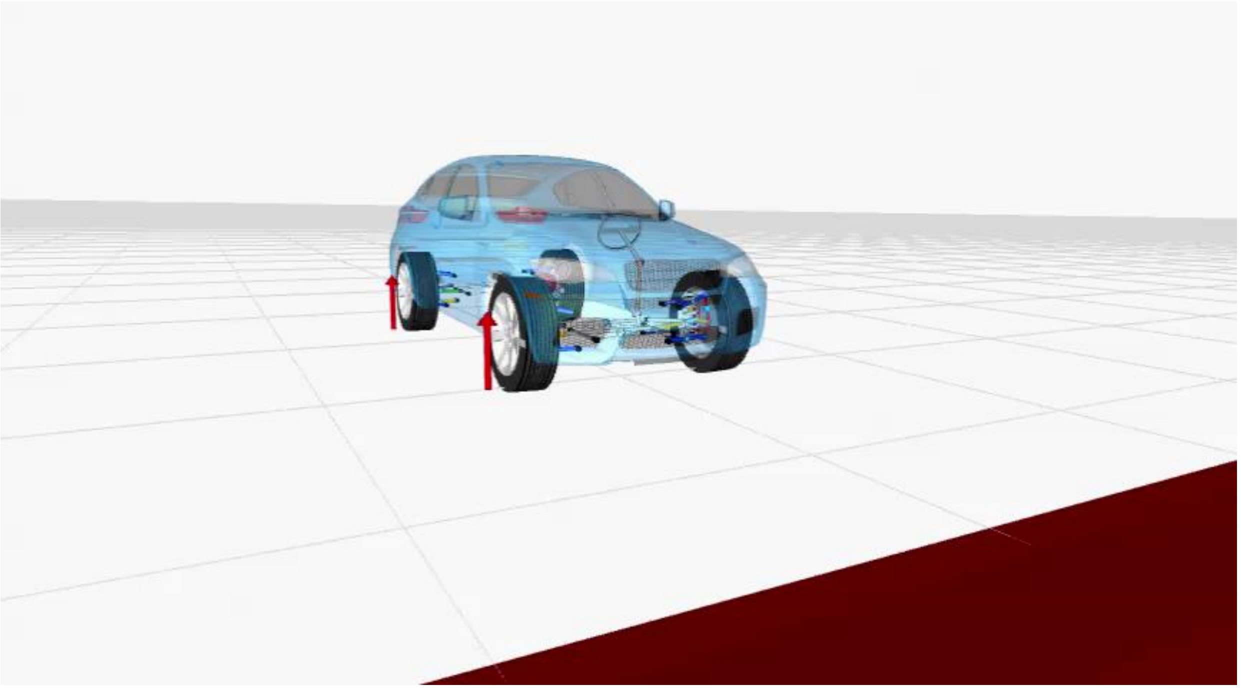
MEASURED RESULTS

- <500μs average evaluation time on Concurrent hardware
- Max 1 time step to recover overrun @ 1ms
- Extreme excitation, jumping 360/police turn
- Good accuracy compared to offline simulation





ANIMATED RESULT OF VALIDATION RUNS



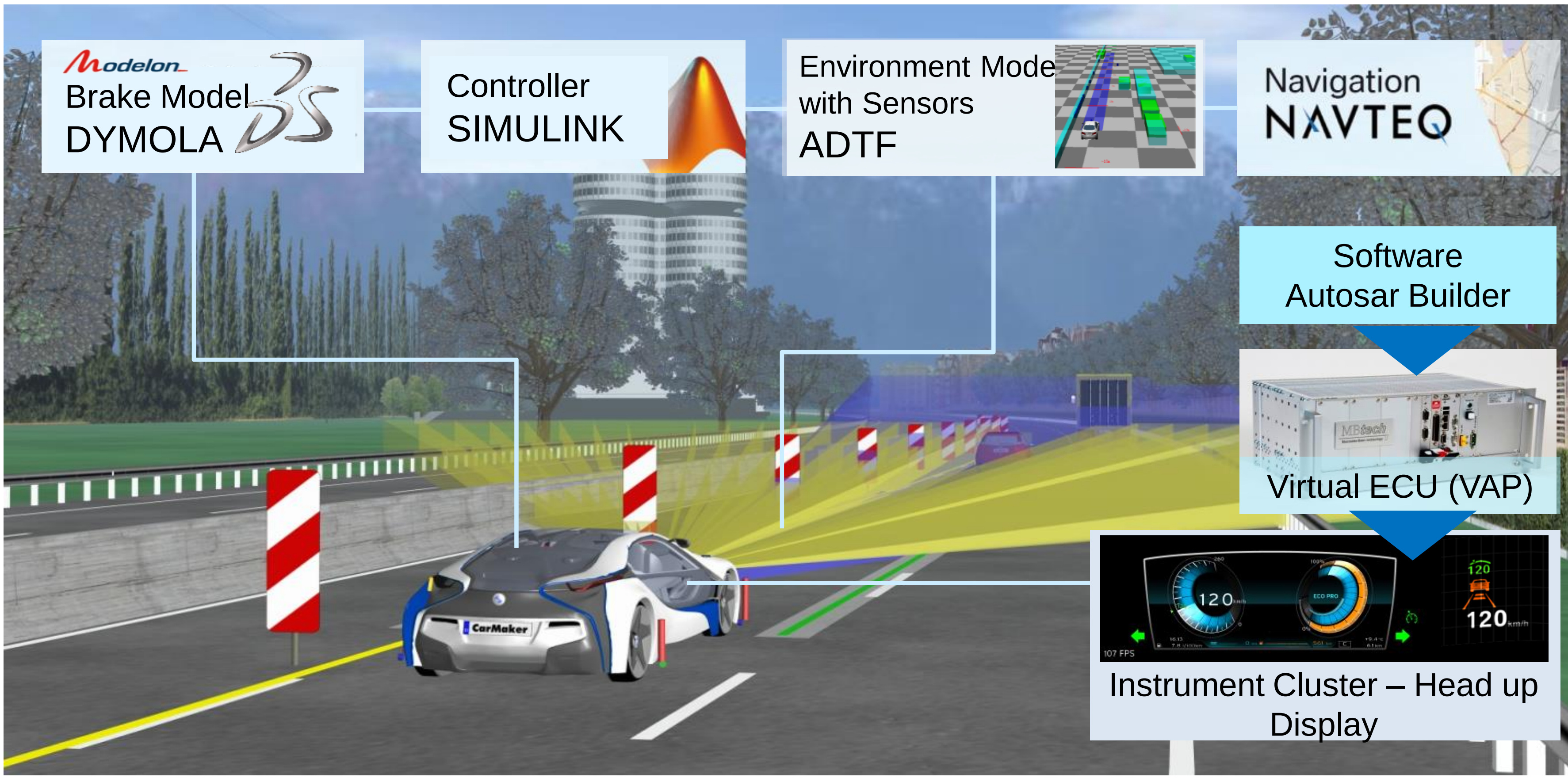
ILLUSTRATE A USE CASE

Functional validation of environment model and
- ACC (Adaptive-Cruise-Control and
- EBA (Emergency Brake Assist)

functions with **Model-in-the-Loop** at BMW

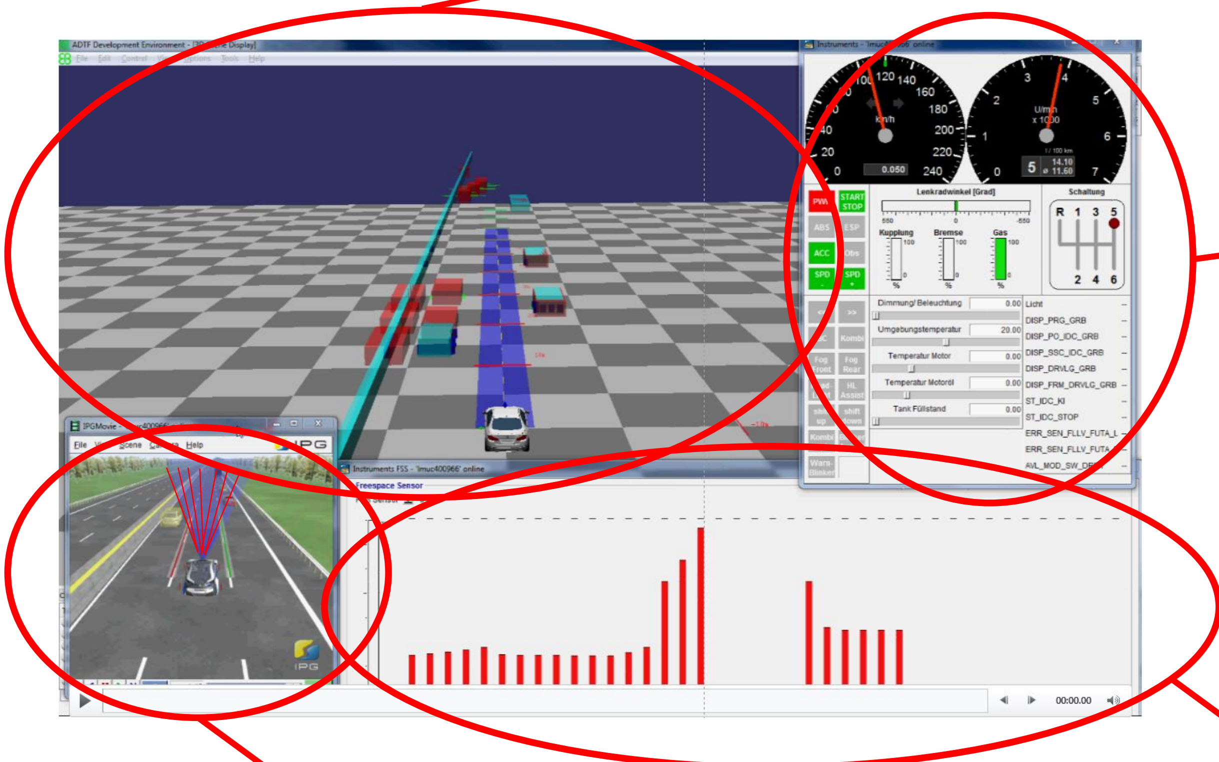
ILLUSTRATE A USE CASE

Function meets multiple function behavior
ACC–Adaptive Cruise Control and EBA–Emergency Brake Assistance



ILLUSTRATE A USE CASE

Development environment for reconstruction the world from raw sensor signal

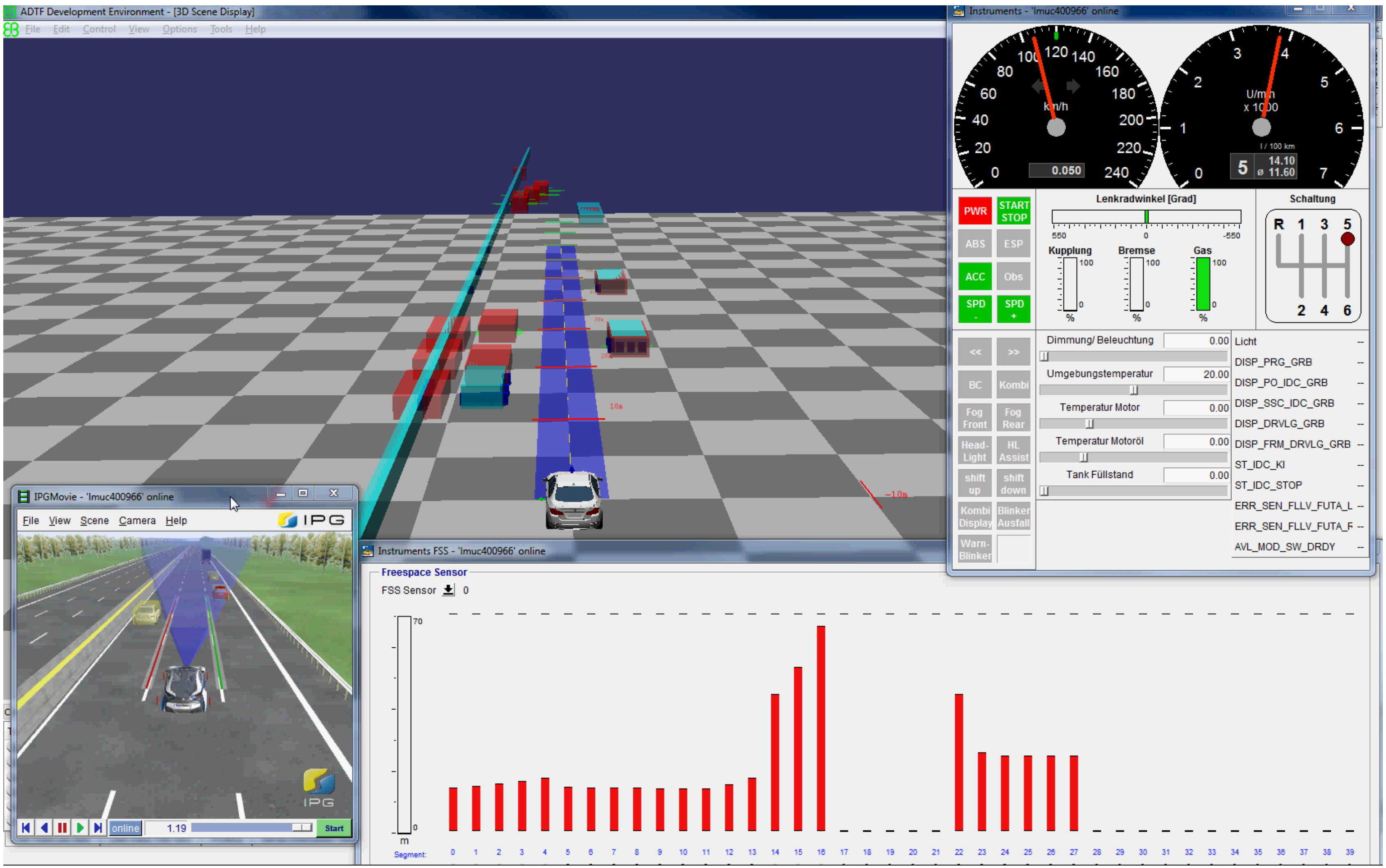


Dashboard: velocity, engine speed, gear, gas level, clutch, status of ACC/EBA controllers

Simulator: birds eye scene view (CarMaker)

Raw signal of free range sensor

ADAPTIVE CRUISE CONTROL IN ACTION



MODELICA

Global Automotive Advisory Group

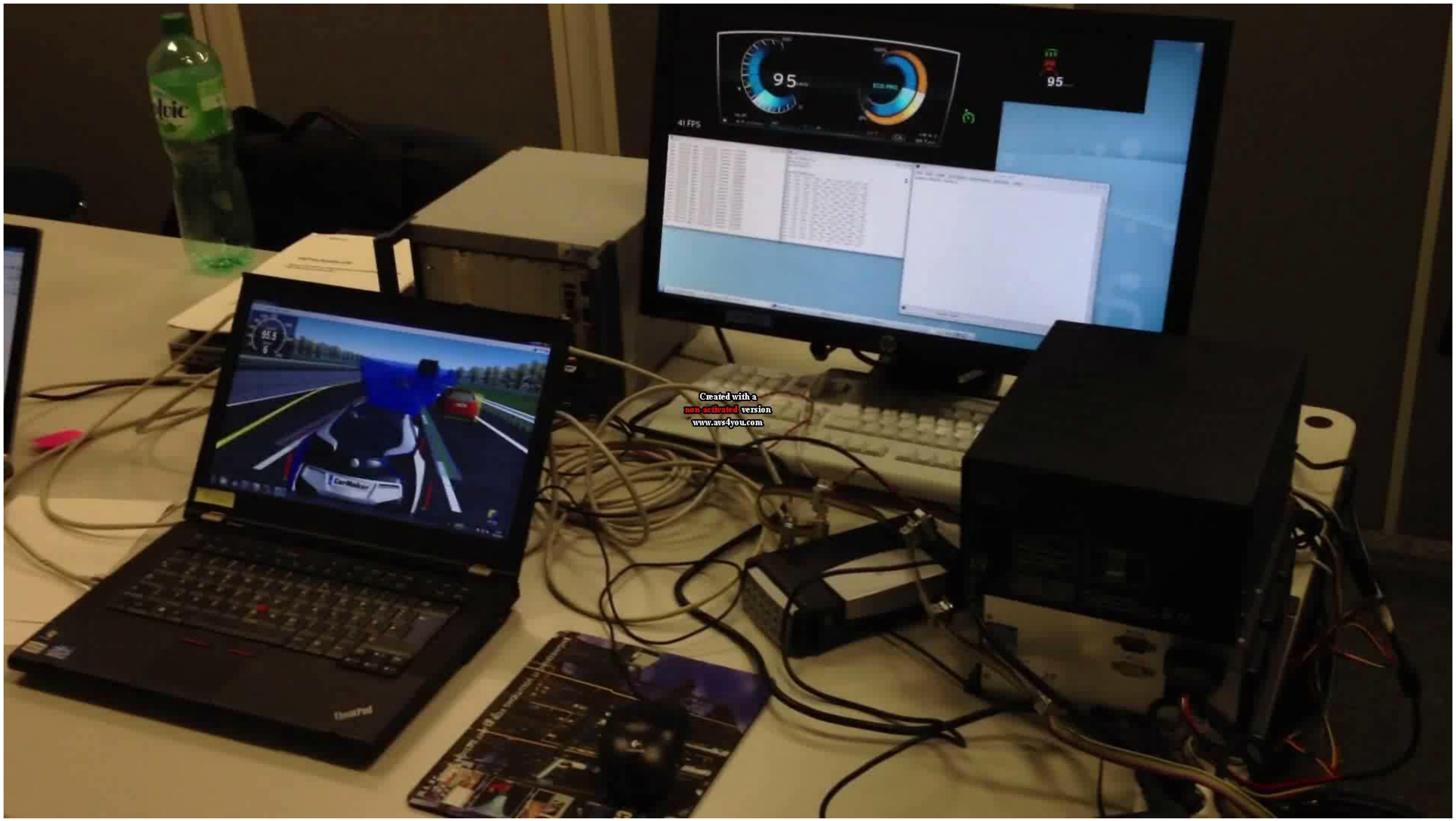
IPG AUTOMOTIVE

Modelon

Modelon

ILLUSTRATE THE USE CASES

Validation of AUTOSAR HMI Software components with virtual ECU by **Software-in-the-Loop** at BMW

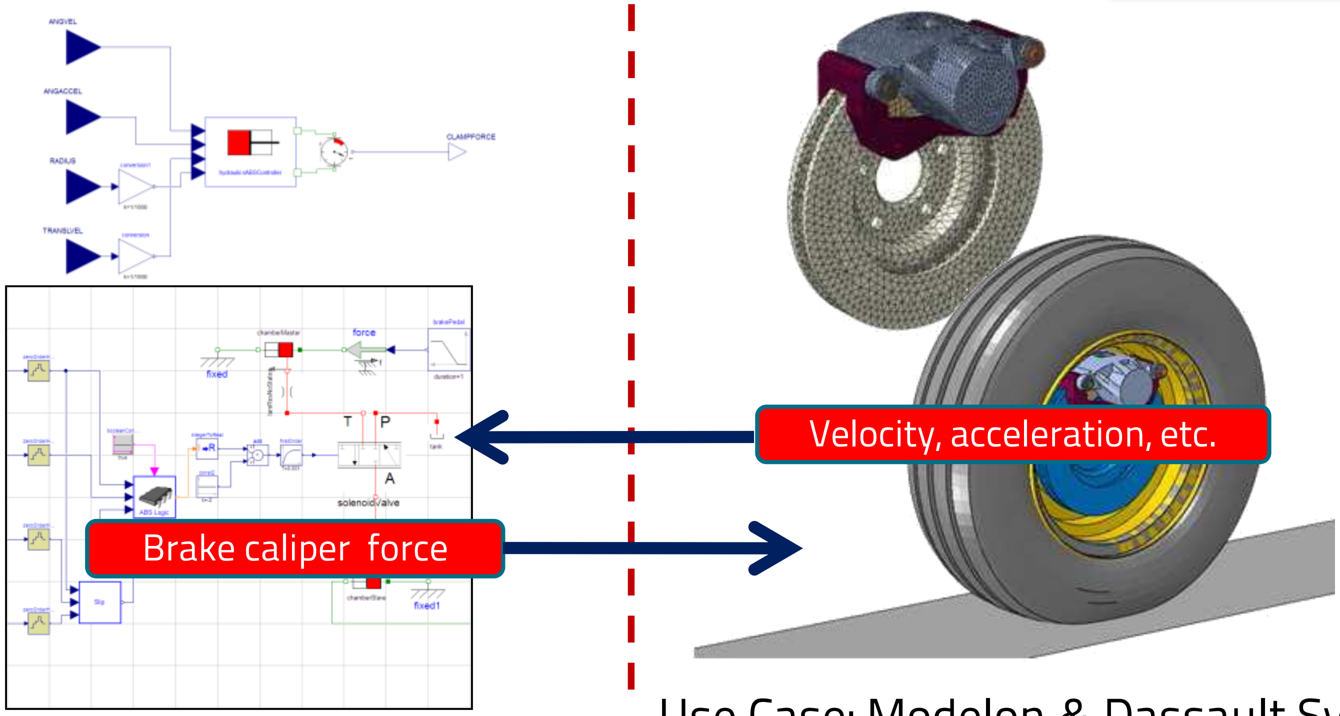


USE CASE II: BRAKING ON WET ROAD

High fidelity antilock brake system simulation

- Hydraulic braking system and ABS logic modeled in Modelica / Dymola
- 3D brake pad and tire/road interaction in Abaqus

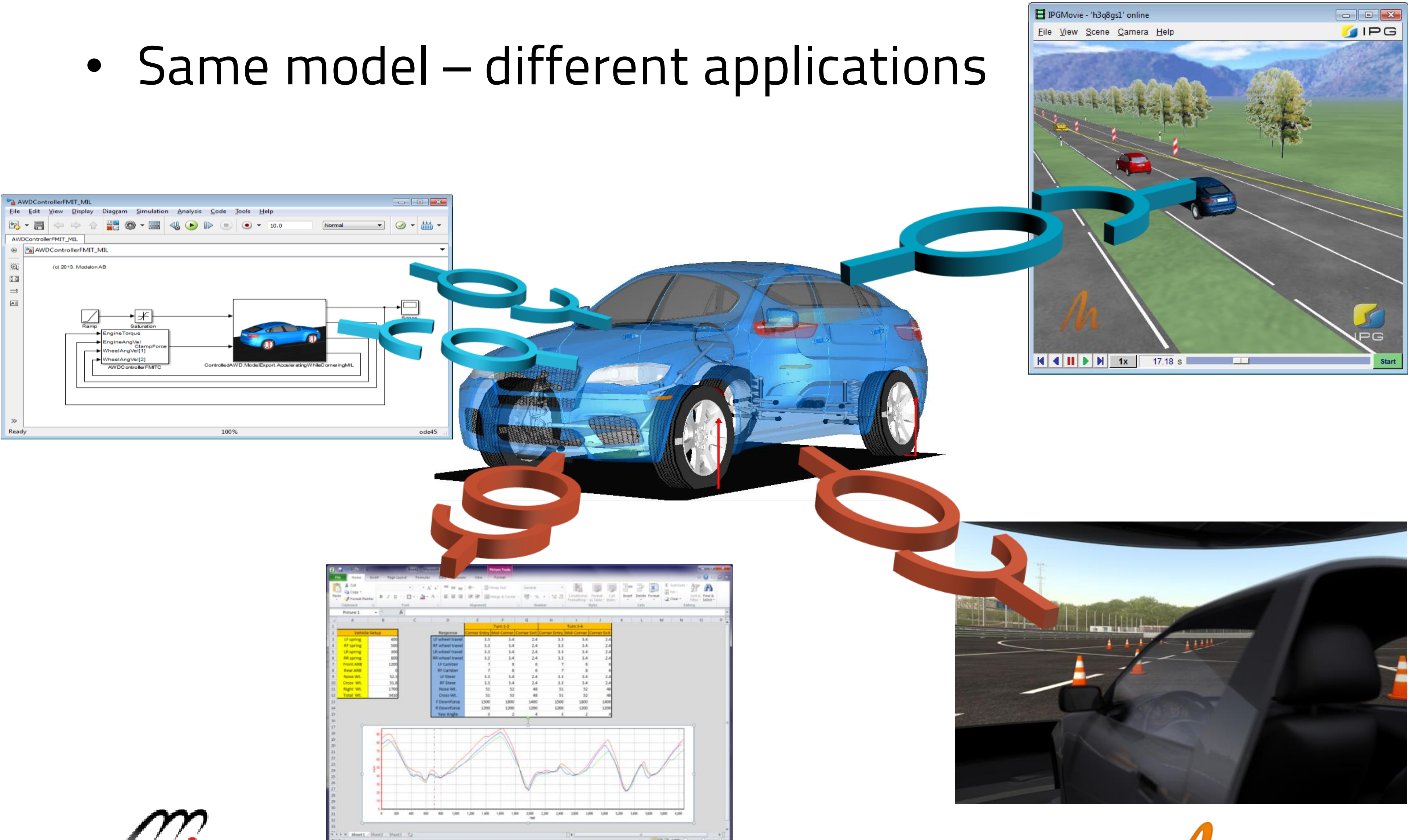
One of many system model configurations

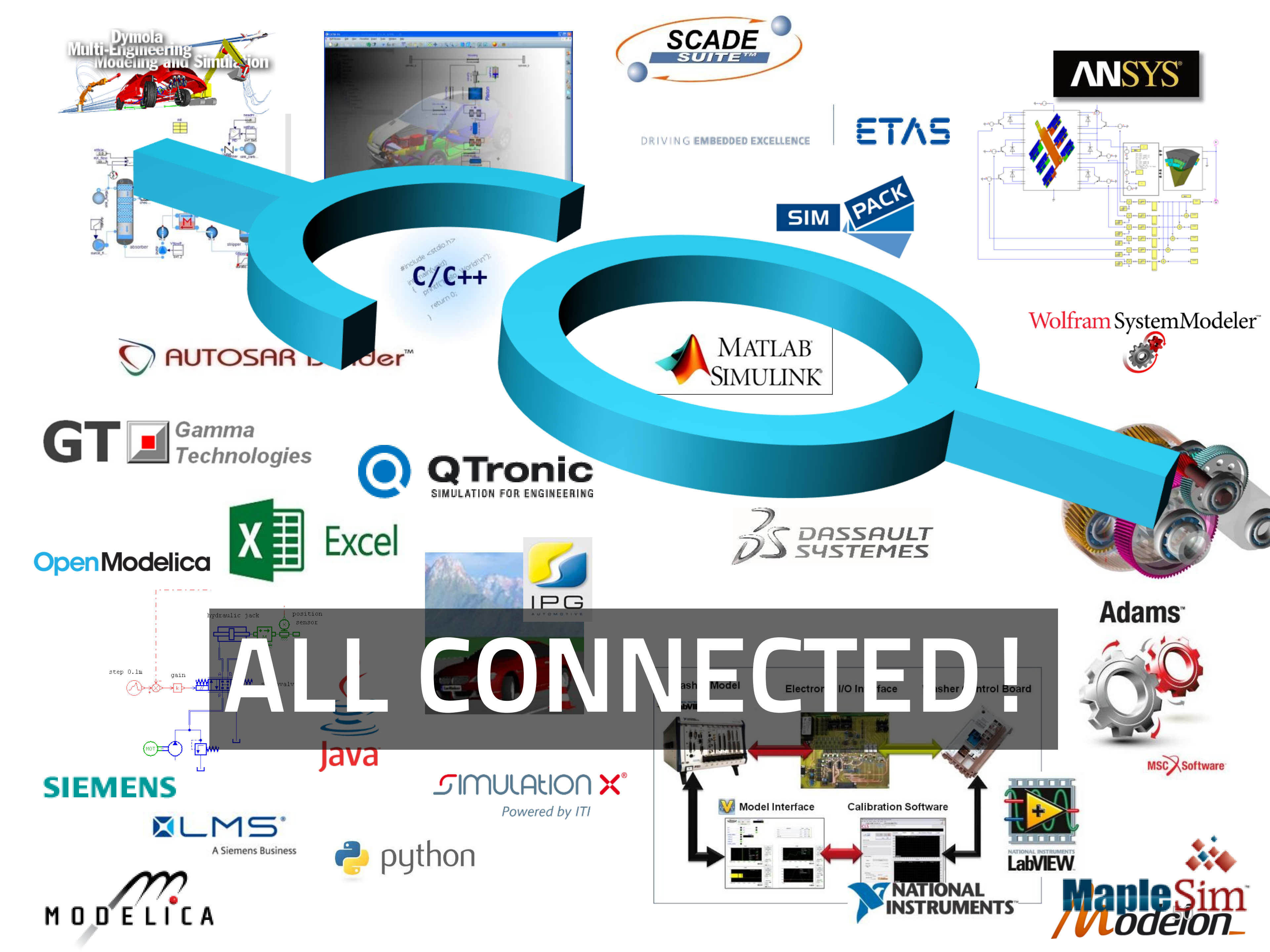


Use Case: Modelon & Dassault Systèmes

FMI ADVANTAGE

- Same model – different applications




Dymola
Multi-Engineering
Modeling and Simulation
SCADE SUITE™
ANSYS
DRIVING EMBEDDED EXCELLENCE
ETAS
SIM PACK
C/C++
Wolfram SystemModeler
AUTOSAR Builder™
MATLAB
SIMULINK
GT Gamma Technologies
QTronic
SIMULATION FOR ENGINEERING
Excel
OpenModelica
DASSAULT SYSTEMES
IPG
AUTOMOTIVE
Adams
ALL CONNECTED !
hydraulic jack
position sensor
step 0.1m
gain
Model
Electron
I/O In
face
ntrol Board
Java
MSC Software
SIEMENS
SIMULATIONX
Powered by ITI
Model Interface
Calibration Software
LMS
A Siemens Business
python
NATIONAL INSTRUMENTS
LabVIEW
MapleSim
Modelon
MODELICA

Controls & Embedded FMI Tools

Tool name	Description
ASim - AUTOSAR Simulation	AUTOSAR product from Dassault Systèmes
ANSYS SCADE Suite	SCADE Suite is a model-based development environment with certified code generation for safety critical embedded applications from ANSYS.
AVL CRUISE	Vehicle system analysis tool for the optimization of fuel efficiency, emission, performance and driveability, from office to HiL to testbed.
CarMaker	CarMaker is an open test- and integration-platform for MiL, SiL and HiL.
ControlBuild	Environment for IEC 61131-3 control applications from Dassault Systèmes
CosiMate	Co-simulation Environment from ChiasTek
dSPACE SCALEXIO	dSPACE SCALEXIO is a Hardware-in-the-Loop (HIL) integration and simulation platform from dSPACE. Please also refer to the dSPACE FMI sites for more information about the FMI 1.0 and FMI 2.0 support.
dSPACE SYNECT	dSPACE SYNECT is a data management tool from dSPACE that enables you to manage FMUs and Simulink models as well as their dependencies, versions and variants throughout the entire software development process. Please also refer to the dSPACE FMI sites for more information about the FMI support.
dSPACE VEOS	dSPACE VEOS is a PC-based virtual integration and simulation platform from dSPACE. Please also refer to the dSPACE FMI sites for more information about the FMI 1.0 and FMI 2.0 support.

Controls & Embedded FMI Tools

Tool name	Description
Dymola	Modelica environment from Dassault Systèmes.
ETAS - ASCMO	Creation and export of statistical (meta) models using Design of Experiments (DoE) from ETAS.
ETAS - FMI-based Integration and Simulation Platform	Integration and simulation platform based on FMI 1.0 from ETAS.
ETAS - FMU Generator for ASCET	FMU Generator for ASCET from ETAS.
ETAS - FMU Generator for Simulink®	FMU Generator for Simulink® from ETAS.
ETAS - INCA-FLOW (MiL/SiL Connector)	Guided and automated calibration of FMUs with connection to ETAS INCA.
ETAS - ISOLAR-EVE (ETAS Virtual ECU)	PC based platform from ETAS for ECU software validation at the component, sub-system or system level; allows for validation of Application Software, Basis Software and complete ECU software in a virtual environment.
ETAS - LABCAR-OPERATOR	Frontend for ETAS HiL systems LABCAR, operating on the creation of experiments and their subsequent execution.
FMI add-on for NI VeriStand	NI VeriStand supports FMI through the use of the FMI add-on for NI VeriStand from Dofware
FMI Library	Open source (BSD) C library for integration of FMI technology in custom applications by Modelon.
FMI Toolbox for MATLAB/Simulink	The FMI Toolbox for MATLAB/Simulink from Modelon enables FMU import and export for MATLAB/Simulink for both model exchange and co-simulation.
FMUSDK	FMU Software Development Kit from QTronic.

Controls & Embedded FMI Tools

Tool name	Description
IBM Rational Rhapsody	IBM® Rational® Rhapsody® family provides a collaborative design, development and test environment for systems engineers and software engineers.
MESSINA	MESSINA is a test platform for model-based ECU function development.
NI LabVIEW	Graphical programming environment for measurement, test, and control systems from National Instruments
PyFMI	For Python via the open source package PyFMI from Modelon. Also available as part of the JModelica.org platform.
Silver	Generation of virtual ECUs and virtual integration platform for Software in the Loop from QTronic.
xMOD	Heterogeneous model integration environment & virtual instrumentation and experimentation laboratory from IFPEN distributed by D2T.