GLOBAL PRODUCT DATA INTEROPERABILITY SUMMIT 2017

The Functional Mockup-Interface: Innovation through Open Standards

Hubertus Tummescheit, Modelon

Biography

Global Product Data Interoperability Summit | **2017**

Dr. Tummescheit is the President of Modelon Inc., and Chief Strategy Officer and one of the founders of Modelon

He has been involved in the Design of the Modelica language and the FMI standard from the beginning. In 2003 he worked as a research scientist at United Technologies Research Center and returned to Sweden in 2004 to start Modelon, the first company fully dedicated to tools and services based on the open standards Modelica and FMI.

Dr. Tummescheit is also a member of the board of the Modelica Association, and of the FMI steering committee, and active in the future development of both standards.

Dr. Tummescheit has served as the CEO of Modelon in Sweden and moved to the United States in 2013 to establish Modelon as a leading player in system simulation here.

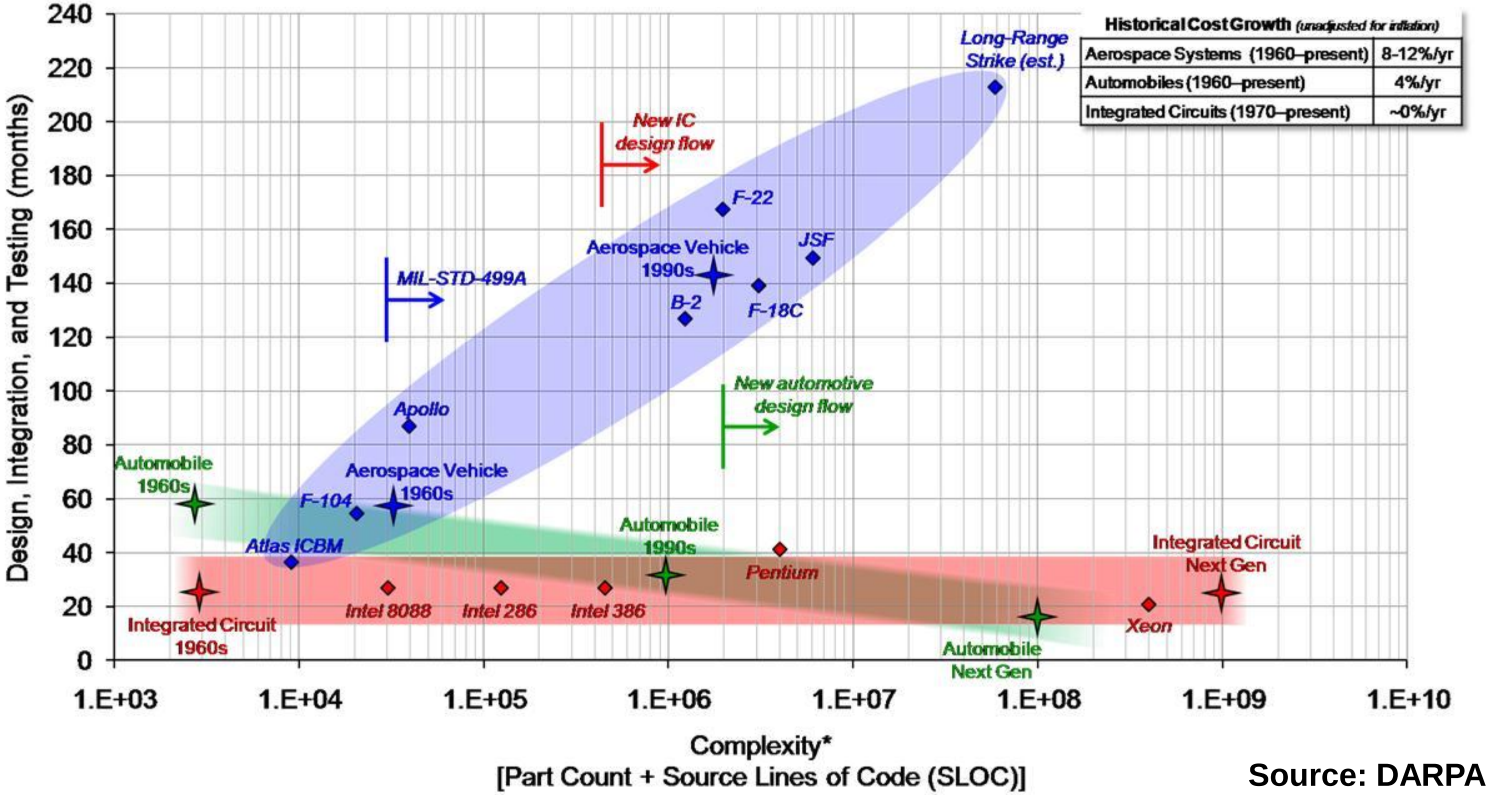
He has an MSc in Mechanical Engineering from Germany, and a PhD in Automatic Control from the University of Lund, Sweden.

Outline of Presentation

- The Functional-Mockup Interface FMI: overview
- Business and Process implications of FMI
- FMI is great – but not magic!
- System Structure and Parameterization SSP: overview
- Conclusions and Outlook

THE COMPLEXITY ISSUE

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

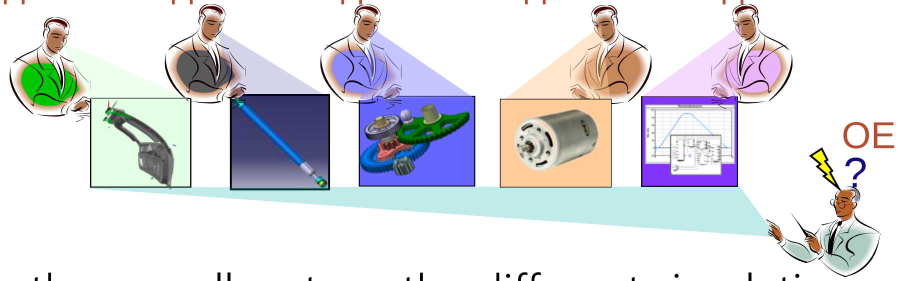


Source: DARPA

1. WHY FMI?

Problem

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



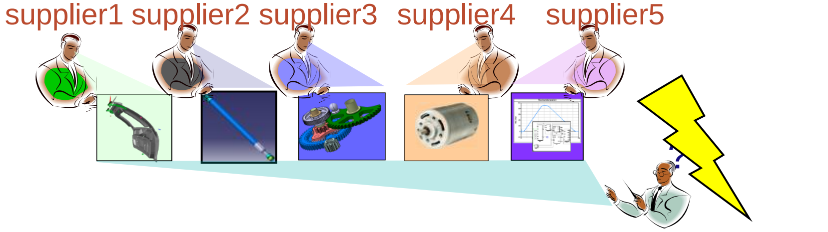
- In order to simulate the overall system, the different simulation programs must interact with each other.
- The system integrator must cope with simulation environments from many suppliers.
- This makes the model exchange a necessity. No current standardized interface.
- Even though Modelica® is tool independent, it cannot be used as such a standardized interface for model exchange.

FUNCTIONAL MOCKUP INTERFACE (FMI)

- Tool independent standard to support both model exchange and co-simulation of dynamic models
- Created to solve model interoperability in automotive industry
- Original development of standard part of EU-funded MODELISAR project led and initiated by Daimler
- First version published in 2010, improved FMI 2.0 in 2014
- Active development as Modelica® Association project

Problems/Needs

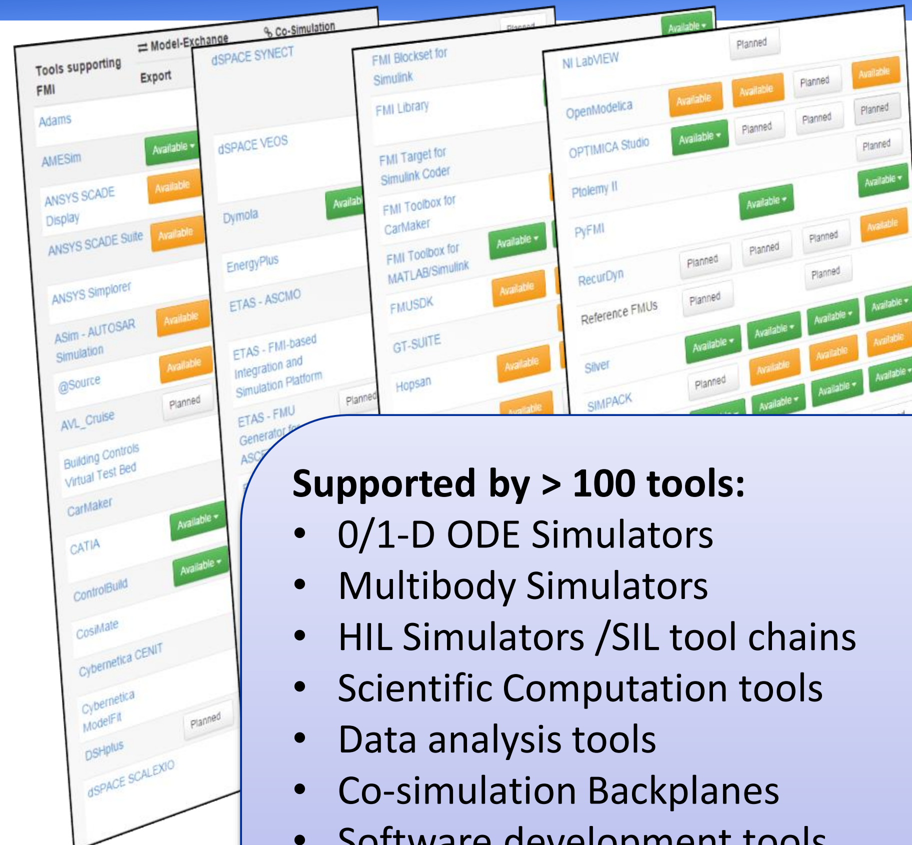
Component developed by Supplier
Integration by systems integrator
Many different simulation tools



FUNCTIONAL MOCKUP INTERFACE (FMI) IN A NUTSHELL

- What is FMI?
  - an application programming interface and its semantics
  - an xml schema that describes the model structure and capabilities
  - the structure of a zip file that is used to package the model, its resources and documentation.
- > 100 tools support FMI in 10 different categories.

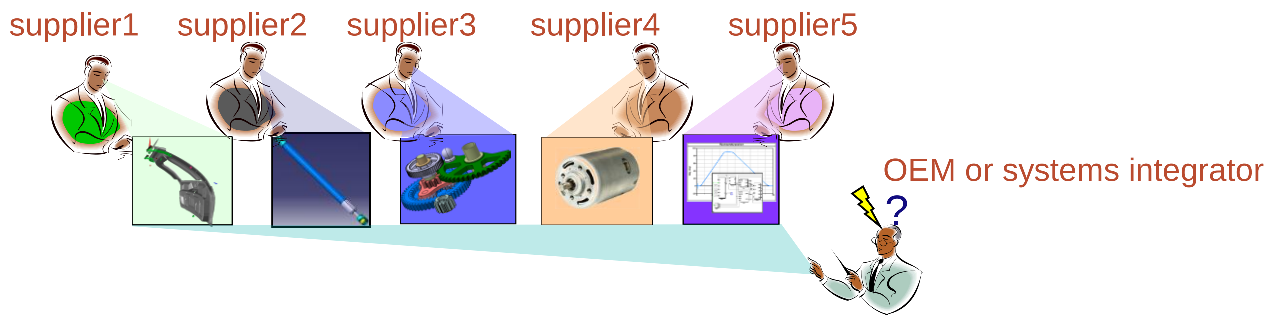
up/-to-date list of tools:
www.fmi-standard.org/tools



Supported by > 100 tools:

- 0/1-D ODE Simulators
- Multibody Simulators
- HIL Simulators /SIL tool chains
- Scientific Computation tools
- Data analysis tools
- Co-simulation Backplanes
- Software development tools
- Systems engineering tools
- SDKs, legacy integration
- CFD tools

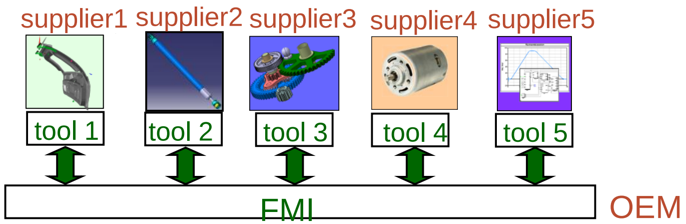
USE CASE I:



Combined simulation for system integration

Solution

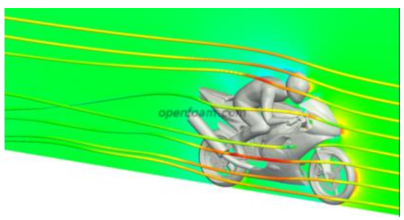
- As a universal solution to this problem the Functional Mockup Interface (FMI) was developed by the EU-project MODELISAR, and is now maintained by the FMI project of the Modelica® Association



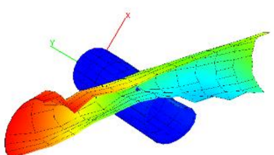
USE CASE II:

- Combine different modeling formats into coherent co-simulation (cyber-physical systems)
  - Physical models, 1D-3D (not 3D to 3D!)
  - Controls / Software

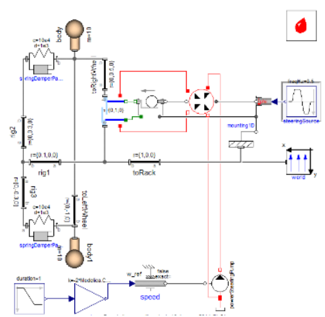
CFD



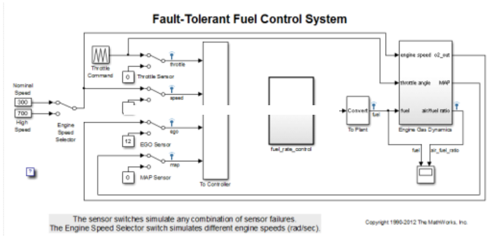
FEA



1-D Systems Simulation



Block Diagrams Controls



Legend

FMI-CS coupling

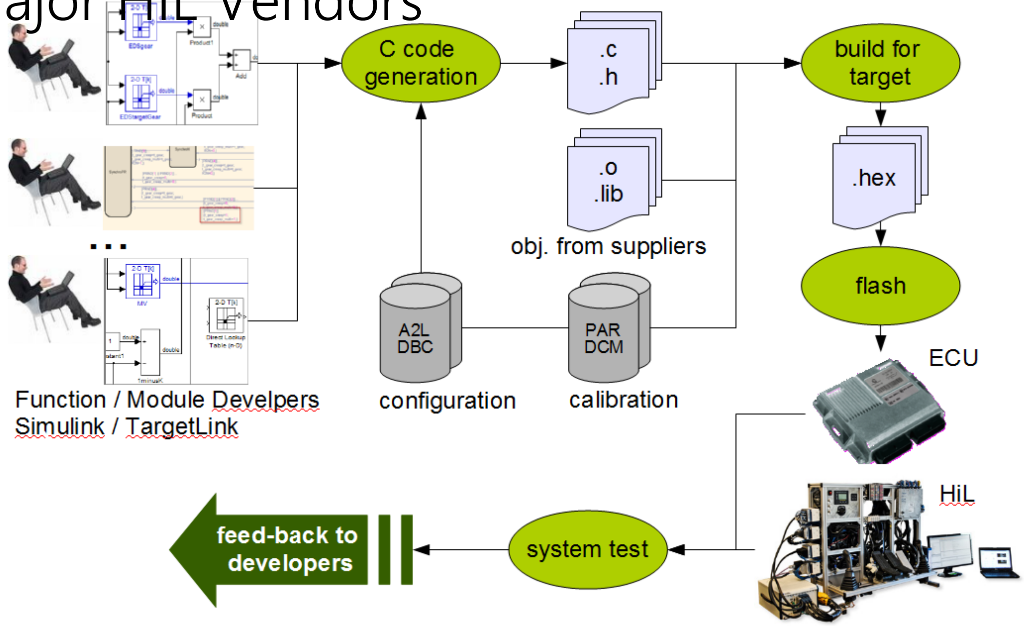
FMI-ME coupling

FMI-based System Simulation

USE CASE III: FMI FOR MIL, SIL AND HIL

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

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

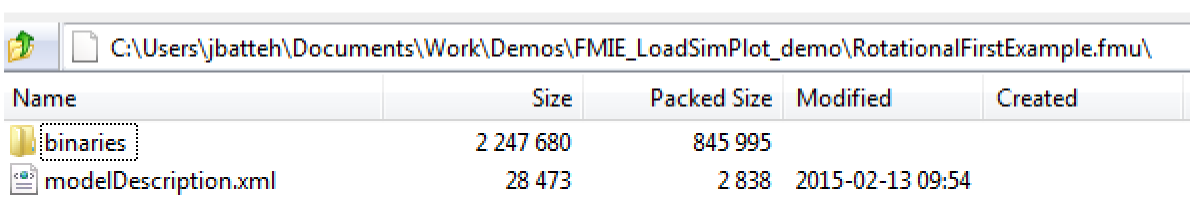


- FMI for ECU virtualization
  - Silver by Qtronic
  - ETAS tools (Bosch)

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

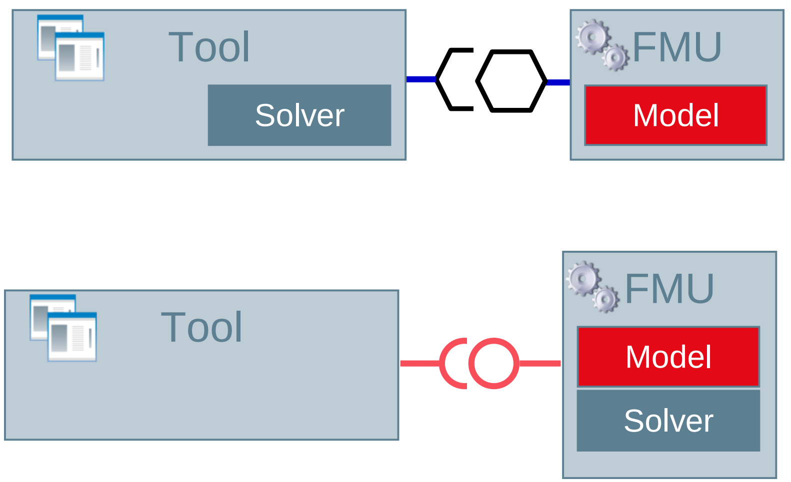

C:\Users\jbatteh\Documents\Work\Demos\FMIE_LoadSimPlot_demo\RotationalFirstExample.fmu\

Name	Size	Packed Size	Modified	Created
binaries	2 247 680	845 995		
modelDescription.xml	28 473	2 838	2015-02-13 09:54	

From the official FMI presentation (adapted)

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 provide an export mode to generate models for license-free distribution in the organisation and to partners
- This is a unique enabler for model exchange inside one company and for OEM/supplier collaboration
- 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)

IP PROTECTION

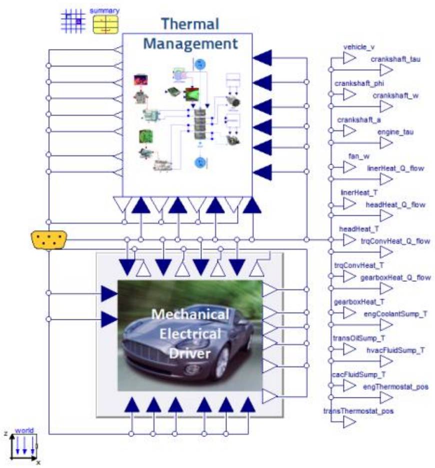
- FMUs contain compiled code
- FMU creator controls level of openness
  - Internal variables
  - Parameters
  - Inputs and outputs only
- Suitable for sharing between OEMs and suppliers

C:\Users\jbatteh\Documents\Work\Demos\FMIE_LoadSimPlot_demo\RotationalFirstExample.fmu\binaries\win32\

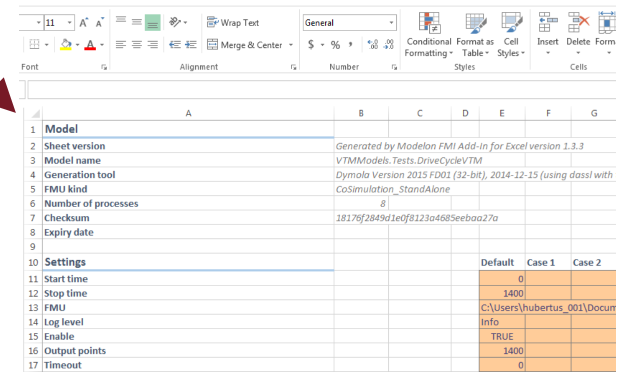
Name	Size	Packed Size	Modified	Created	Accessed
RotationalFirstExample.dll	2 247 680	845 995	2015-02-13 09:54		

10/18/2016

TYPICAL FMI-BASED WORKFLOWS



Export: exported FMU freely licensed



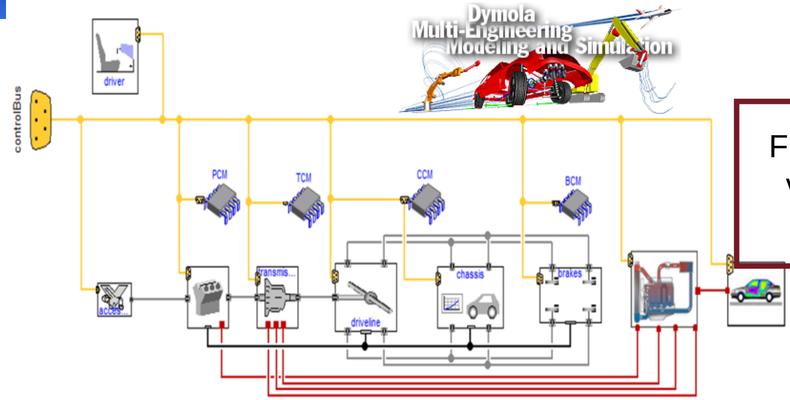
Model Authoring Tool(s)

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

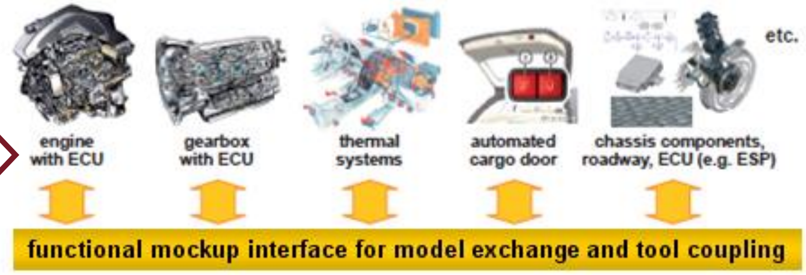
- True democratization of simulation
- Greatly improved utilization of models

EXAMPLE: DEVELOPMENT TO DEPLOYMENT



FMU Export via Model Export

Functional Mockup Interface (FMI)

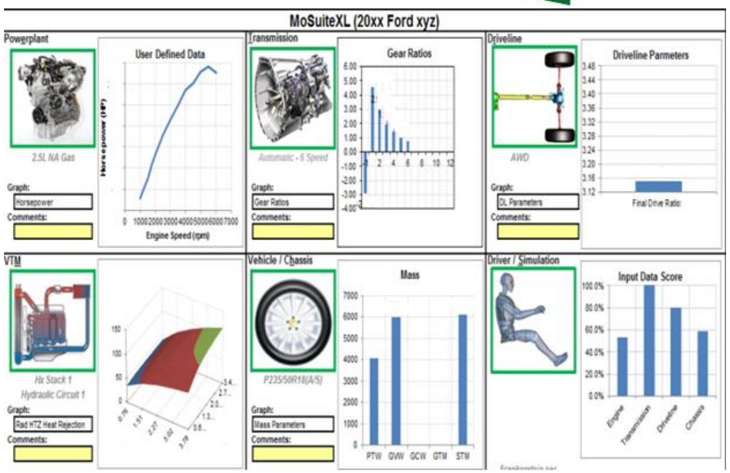


PyFMI

FMU Simulator

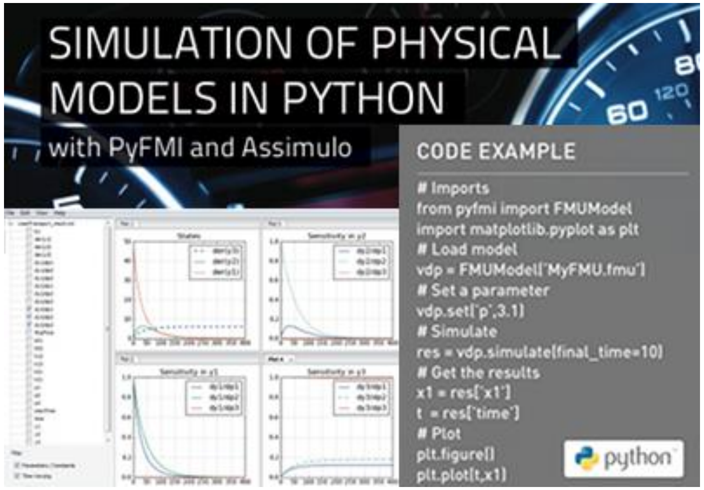
Custom GUI

Excel



Parameters

Results



AUTOMATED REQUIREMENTS VERIFICATION

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

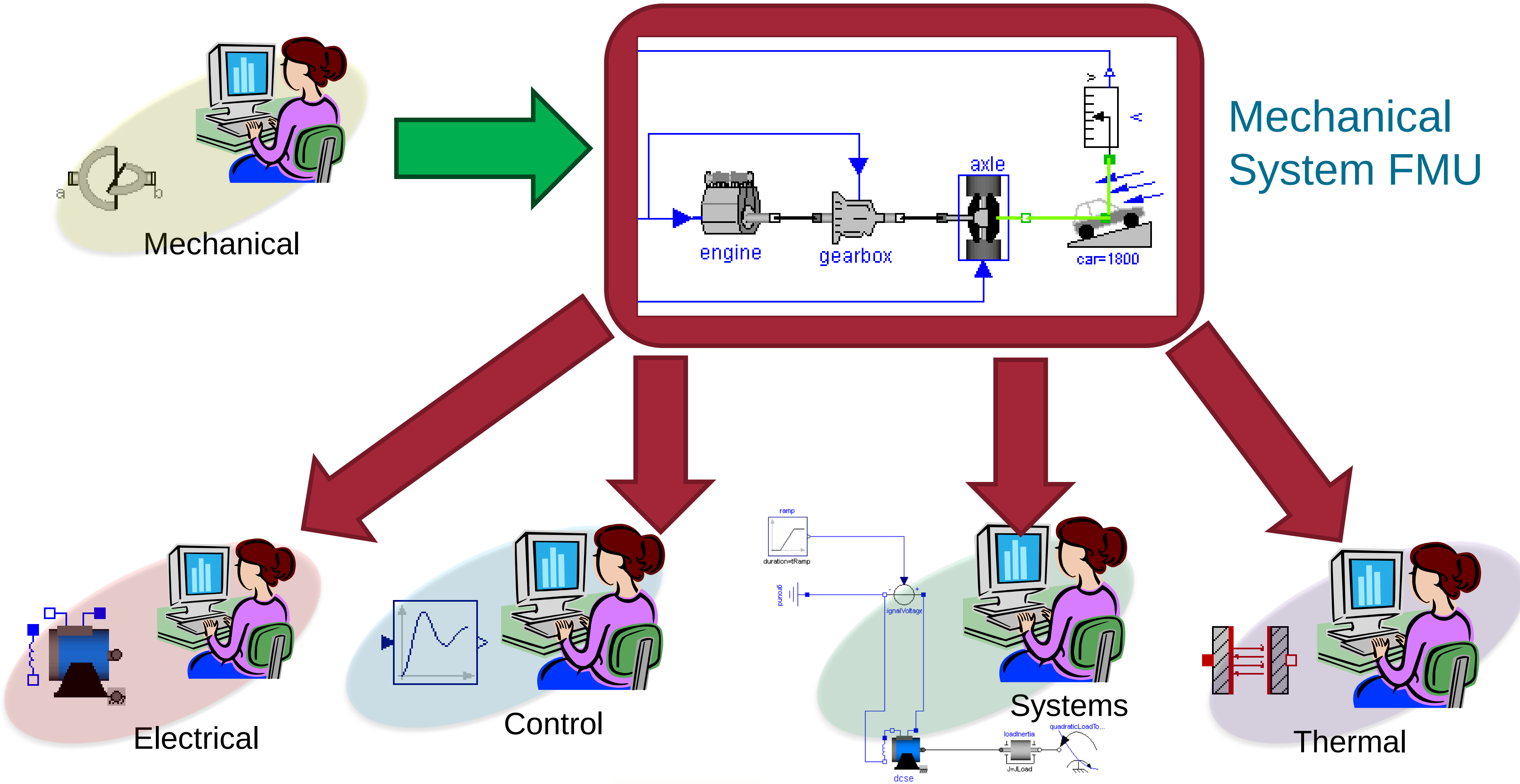
Deploy to team!

- Requirements → Formalized requirements → Executable model of requirements (e.g. FMU)
- Physical plant → Model of plant → Deployable model of plant (FMU)
- Software spec → Software model or prototype → Deployable model of software (FMU)
- Deployable model of environment

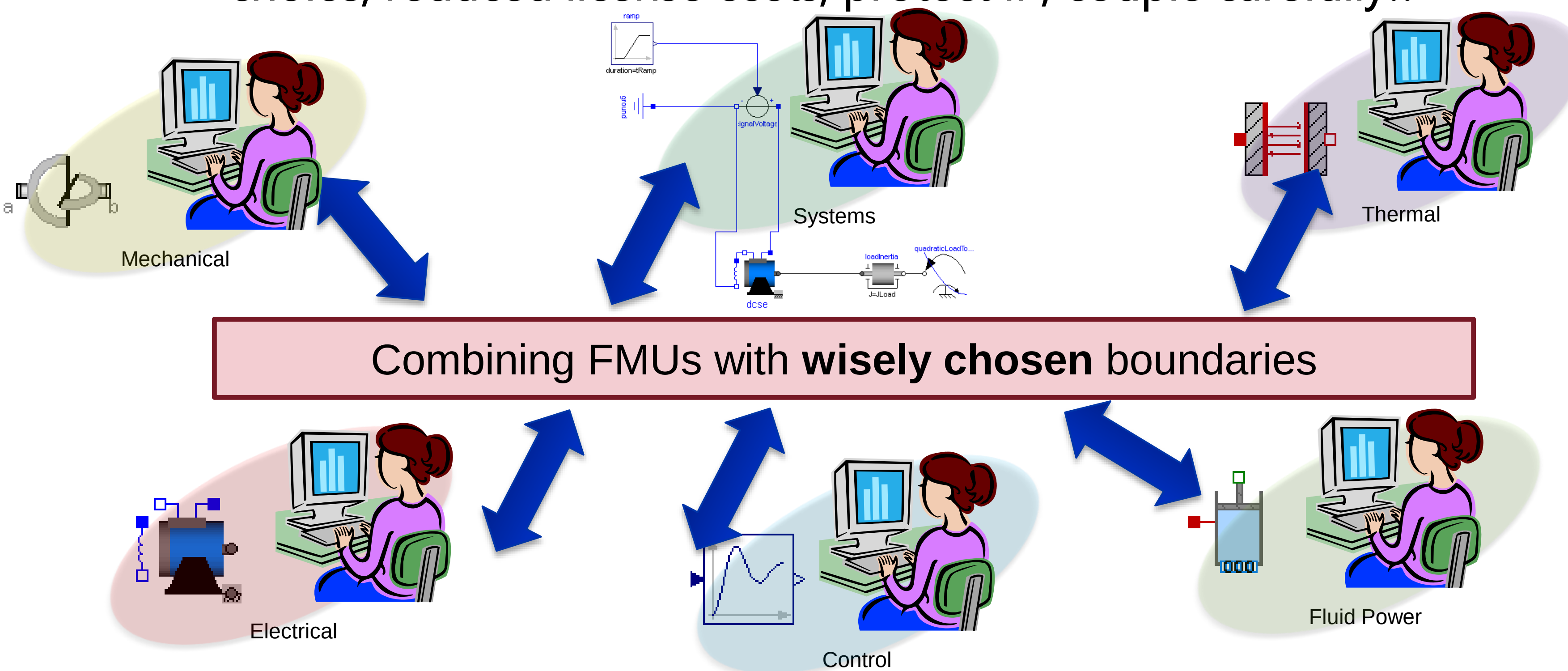
Development of a customized workflow to allow rapid iterations of plant & software configuration

MODEL DEPLOYMENT

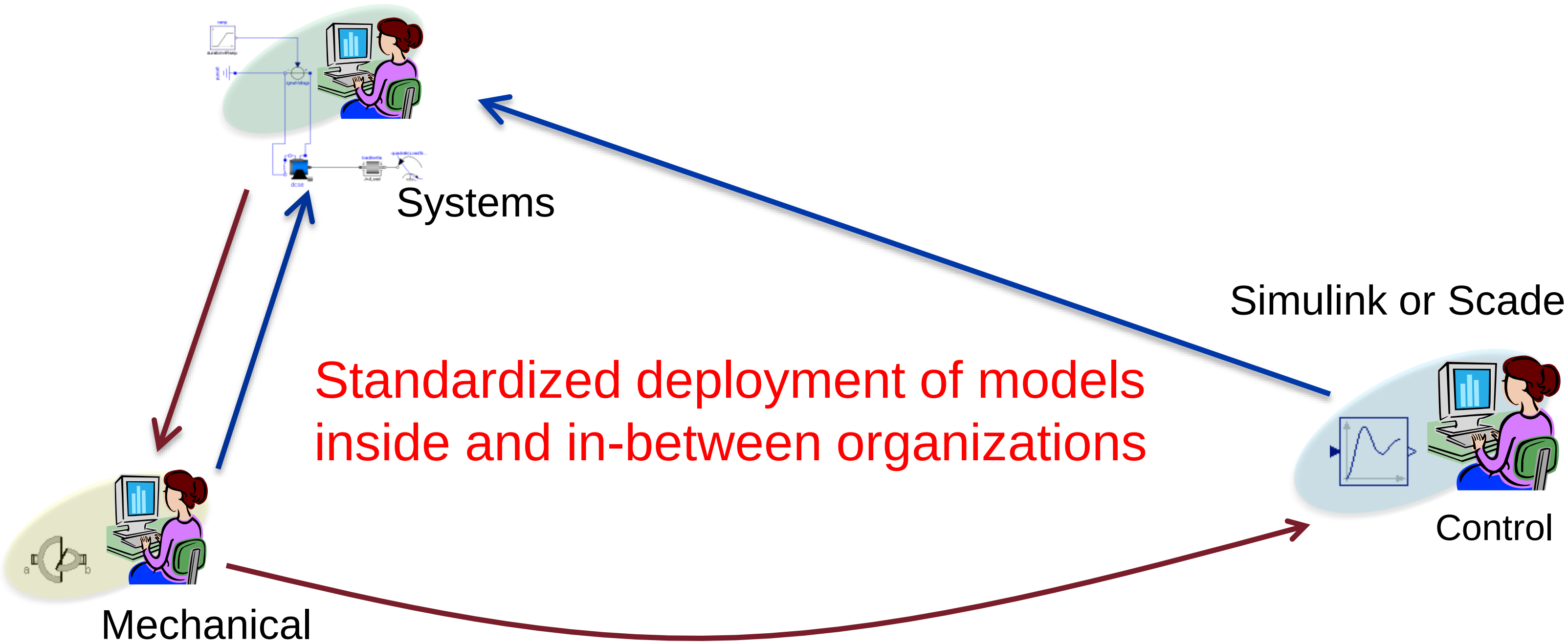
- FMU deployed (native tool) to support multiple applications



- Engineers in different domains work i 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

"Daimler, 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"

ENSURING COMPATIBILITY

Total Numbers	FMI_2.0	ModelExchange		CoSimulation	
Tools supporting FMI	**FMI Version**	**Export**	**Import**	**Slave**	**Master**
FMI Library	FMI_2.0	Planned	Available	Planned	Available
FMI Toolbox for MATLAB/Simulink	FMI_2.0	Available (	Available (	Available (	Available (
FMUSDK	FMI_2.0	Available (		Available (	
General Energy Systems (GES)	FMI_2.0	Planned	Planned	Planned	Planned
GT-SUITE	FMI_2.0		Available (	Available	Available (
IGNITE	FMI_2.0			Planned	Planned
JavaFMI	FMI_2.0				Available
JModelica.org	FMI_2.0	Available (		Available (	Available (

https://www.fmi-standard.org/tools

Open testing framework between FMU exporters and importers

ILLUSTRATE A USE CASE

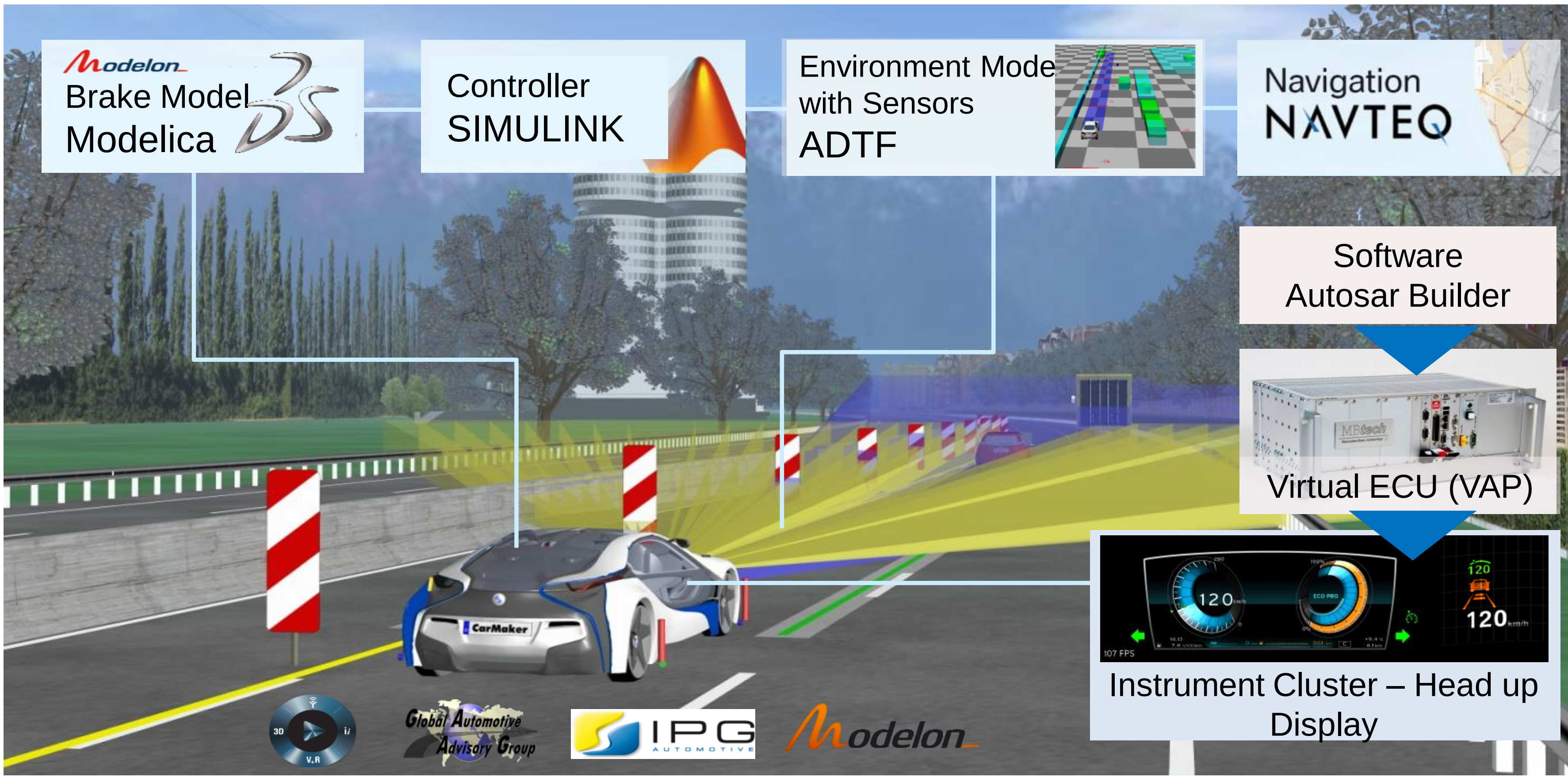
Functional validation of environment model and

- ACC (Adaptive-Cruise-Control and
- EBA (Emergency Brake Assist)

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

ILLUSTRATE A USE CASE

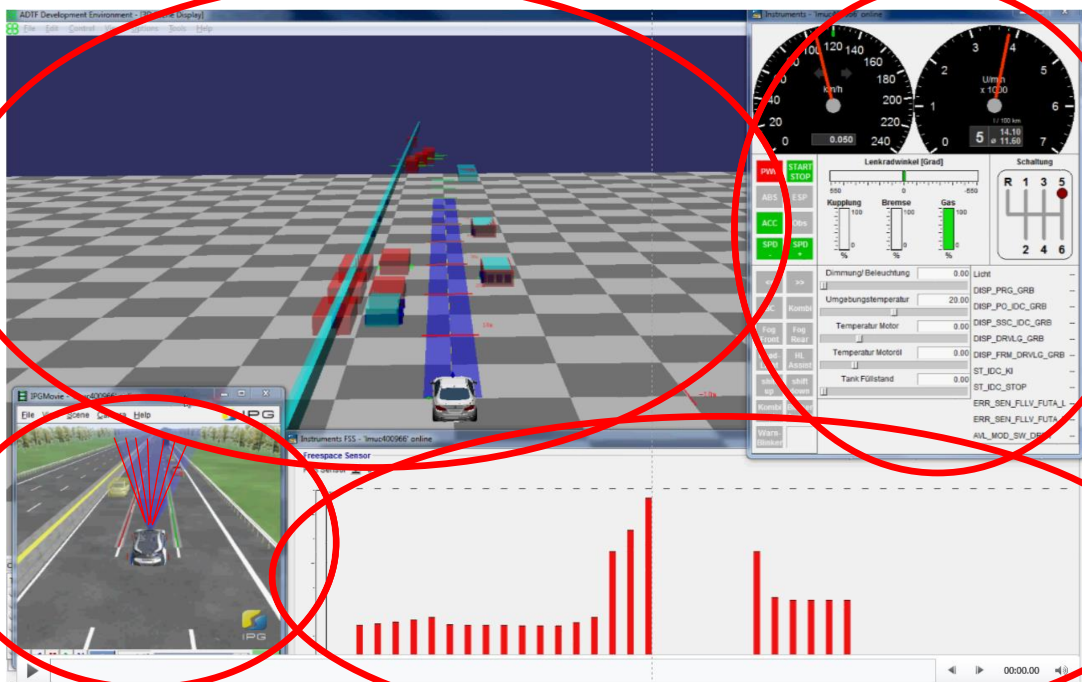
Function meets multiple function behavior
ACC–Adaptive Cruise Control and EBA–Emergency Brake Assistance



ILLUSTRATE A USE CASE

Development environment for reconstruction the world from raw sensor signal

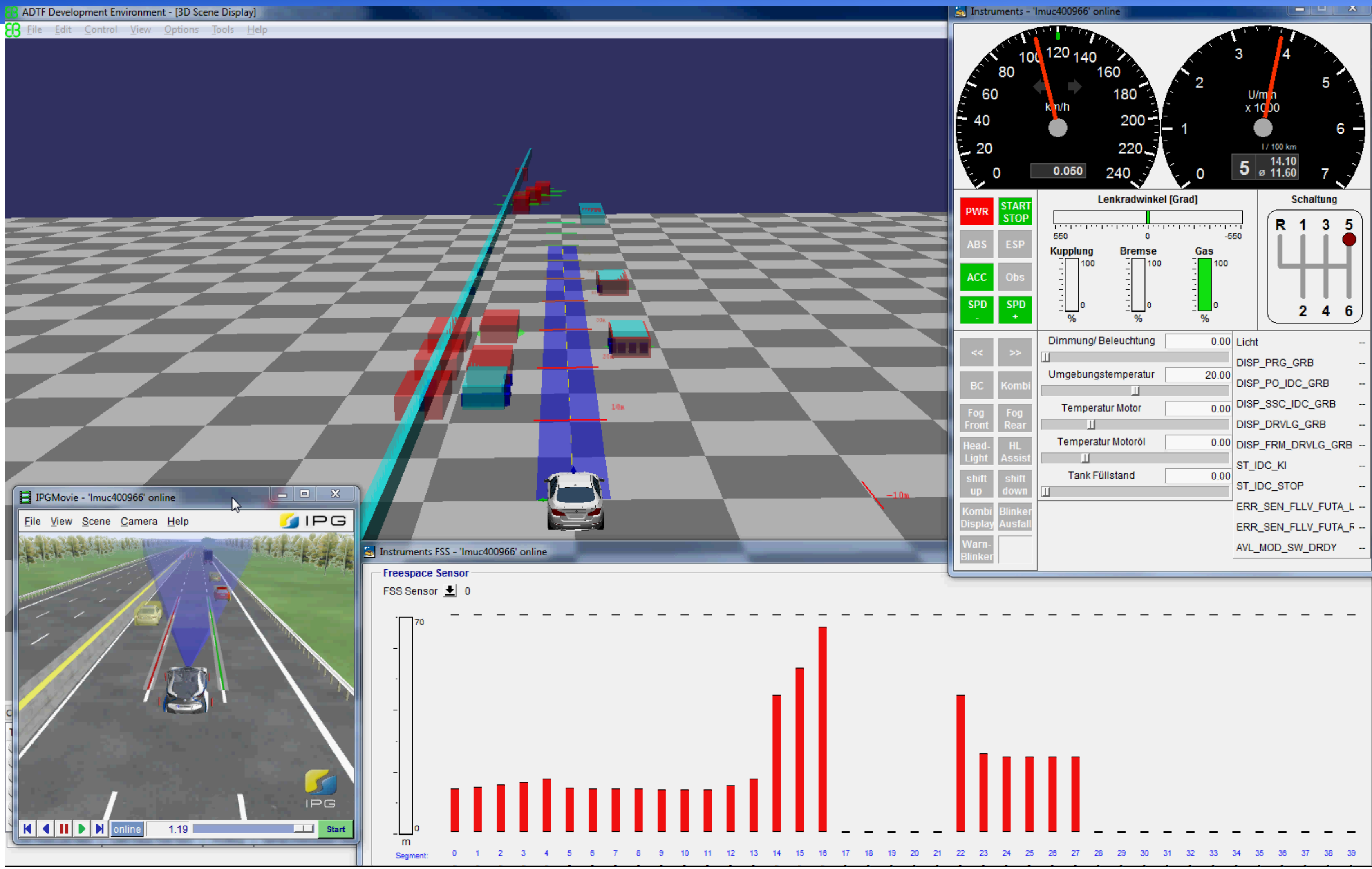
Dashboard: velocity, engine speed, gear,
gas level, clutch, status of ACC/EBA controllers



Raw signal of free range sensor

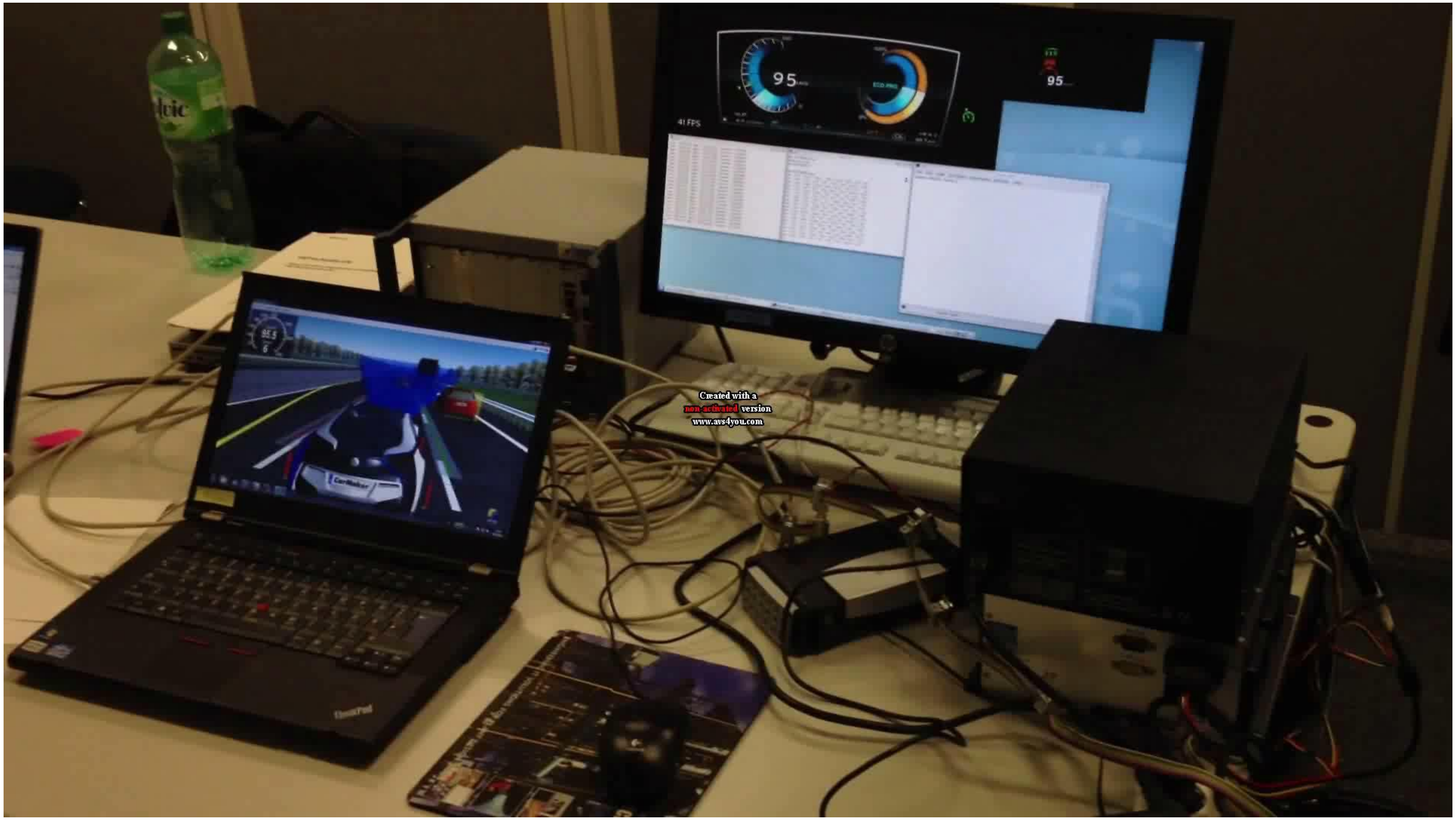
Simulator: birds eye scene view (CarMaker)

ADAPTIVE CRUISE CONTROL IN ACTION



ILLUSTRATE THE USE CASES

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



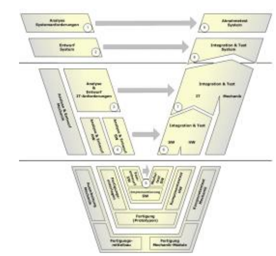
CAN WE ALSO EXCHANGE SYSTEMS?

- Now we can safely exchange one model – as FMU
- What about the system level?

- New project by Modelica Association to develop a standard to represent the system structure

Main Purposes of SSP – Based on FMI standard

- Define a standardized format for the connection structure of a network of components (FMUs in particular).
- Define a standardized way to store and apply parameters to these components.
- The developed standard / APIs should be usable in all stages of development process (architecture definition, integration, simulation, test in MiL, SiL, HiL).
- The work in this project shall be coordinated with other standards and organizations (FMI, ASAM, OMG).



List of features

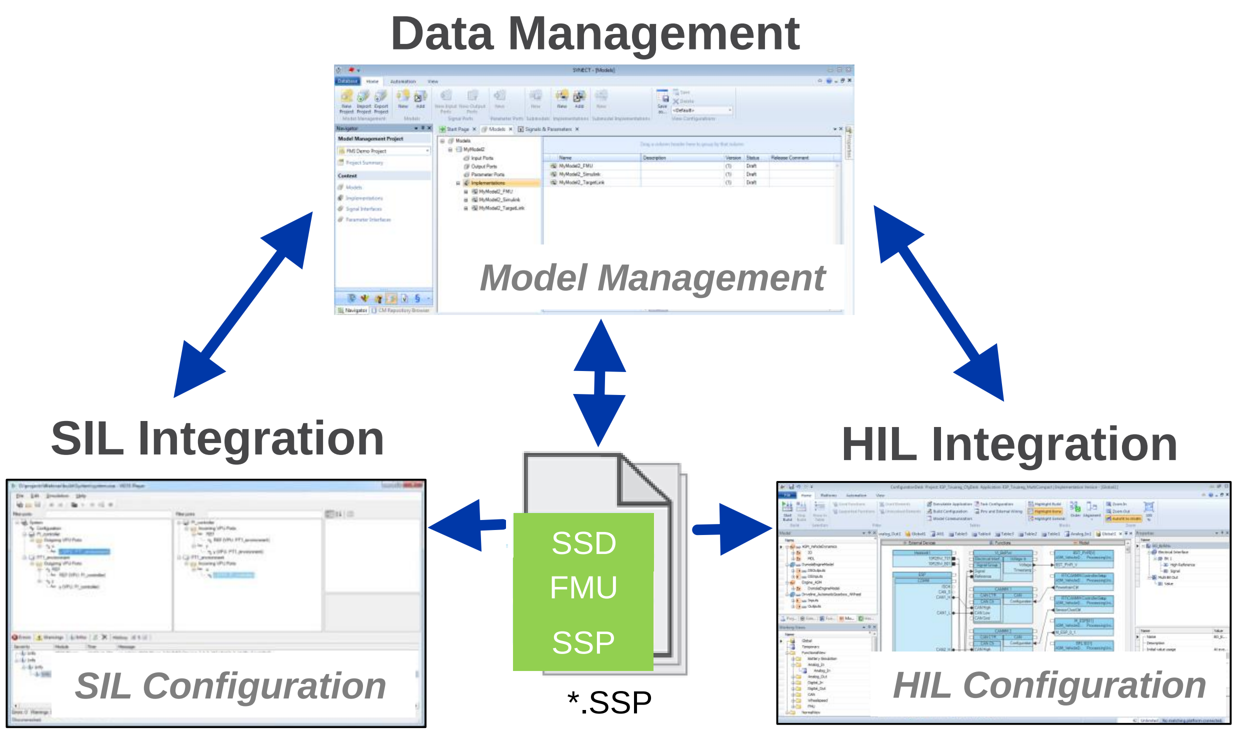
- Hierarchical description of **systems** of connected components
- Components: FMUs and external SSPs/SSDs,
- Parameter bindings both at component and system-level,

  including transformations and name/unit-mapping
- Signal dictionaries support cross-hierarchical data pools (e.g for busses)
- Packaging of SSDs, FMUs, Parameters, … into one bundle (SSP)
- Light-weight support for variant handling at SSP level
- Optional exchange of graphical information (similar display across tools)
- URI references to all resources: Integration with other systems via URIs

Integration of FMUs for SIL & HIL with SSP: Reuse of the System Structure

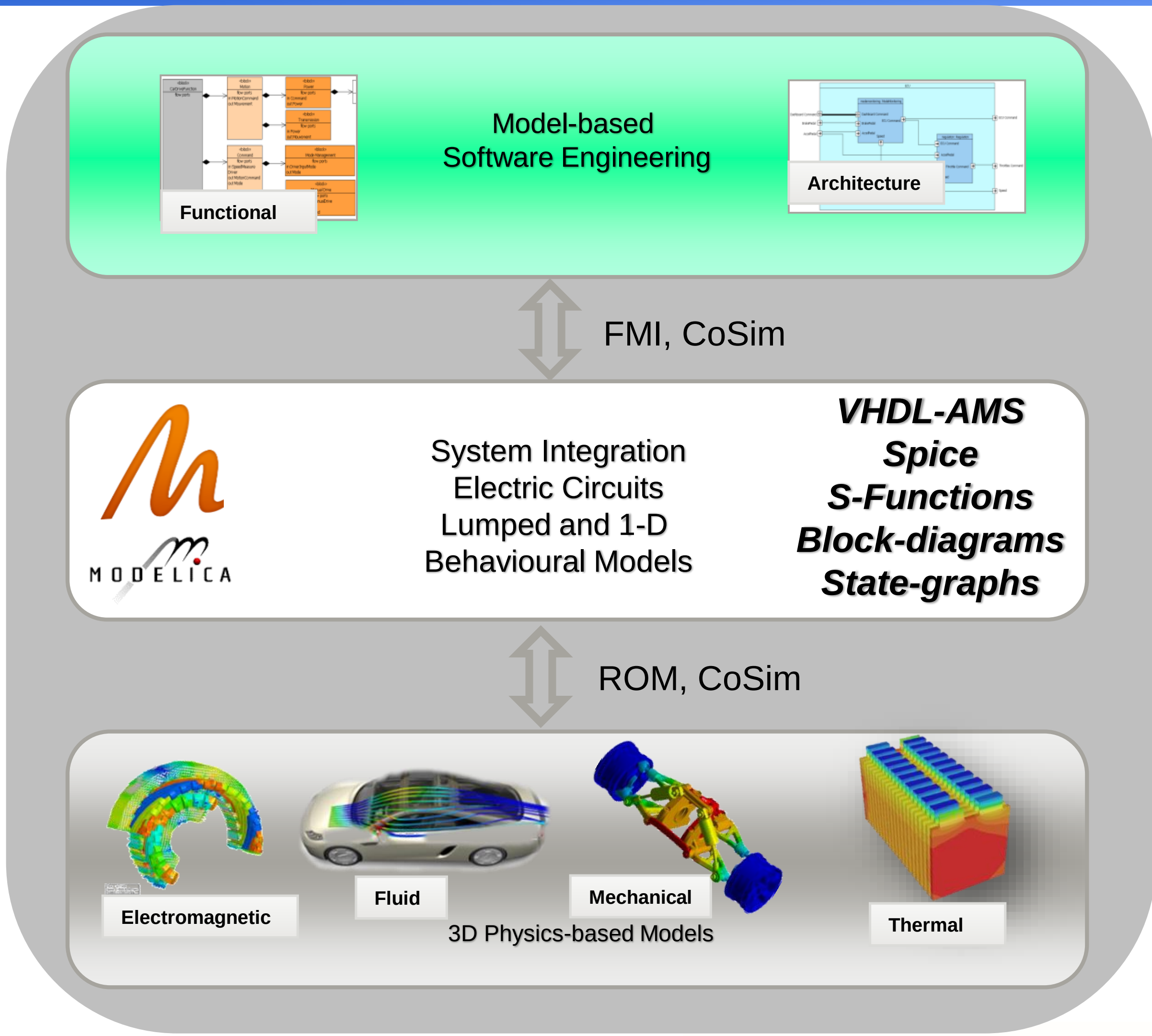
Global Product Data Interoperability Summit | **2017**

- The System Structure defined for SIL can be reused for HIL testing
- It becomes possible to reuse more models, configurations, tests, layouts and parameters
- A Data Management tool controls the lifecycle of the SSP

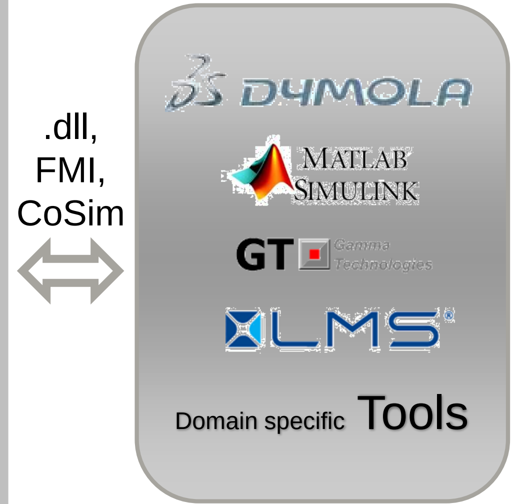


Roadmap for first release: Version 1.0, with a handful of tools supporting it, by end of 2017

FMI & SSP: System Level Integration



ROM = Reduced-order model
FMI = Functional mock-up interface
CoSim = Co-Simulation



Conclusions

- CAE-tools exist in silos: per domain and zoom-level
- With FMI (> 100 tools), finally a broadly accepted standard simplifies collaboration and exchange!
- Processes and business relations should adapt to take full advantage of collaborative MBSE
- We should not stop there!
- The Modelica Association and vendors like Modelon are working to address standardization at the system level: the System Structure and Parameterization standard, SSP in short
- SSP Standard and first tools to be released in Q4 2017