FMI TUTORIAL

Hubertus Tummescheit

INCOSE/NAFEMS SMSWG

INCOSE IW

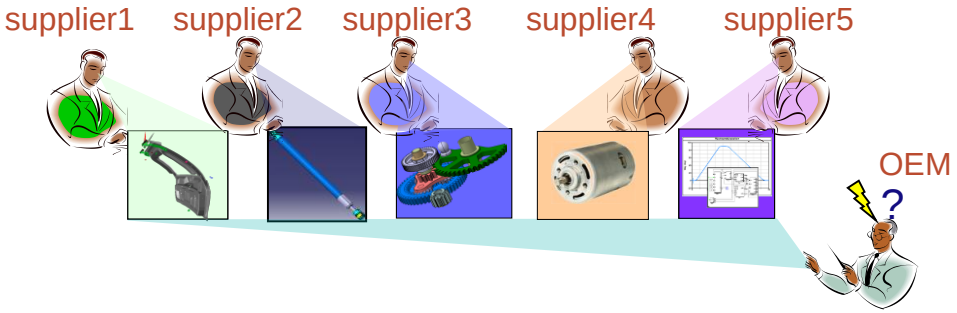
January 26th, 2015

Torrance / CA

1. WHY FMI?

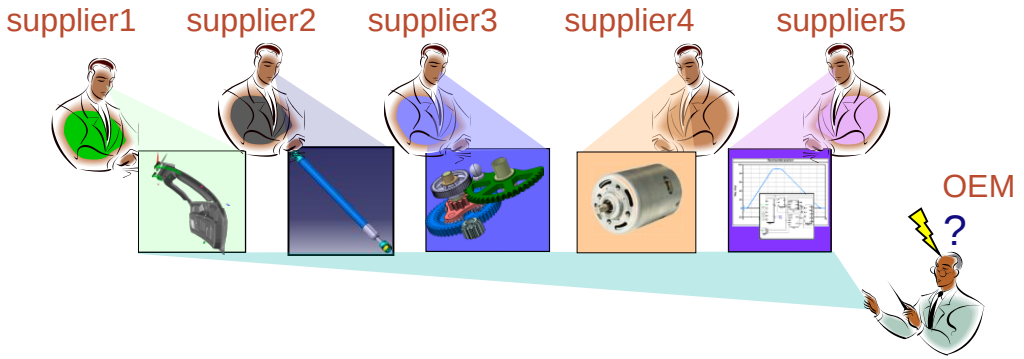
Problem

- 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.

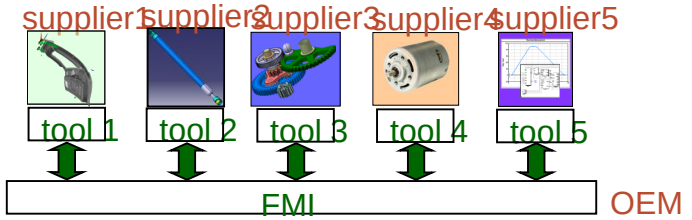
USE CASE I:



Combined simulation for system integration

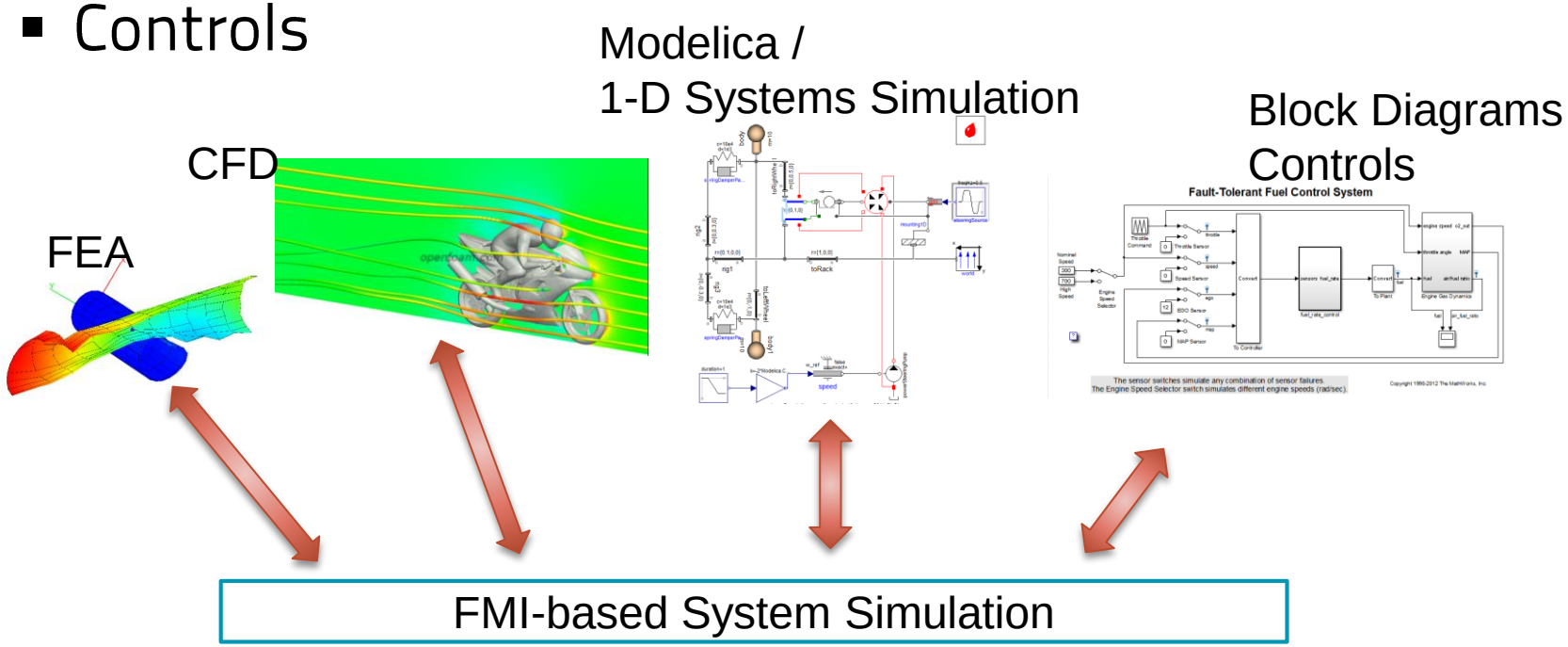
Solution

- 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 formalisms into coherent co-simulation
  - Physical models, 1D-3D
  - Controls



USE CASE III: VIRTUALIZATION FOR CONTROLS

Virtualization: Objectives

Running accurate closed-loop simulation of the complete system – On a PC

All engineers equipped with a virtual vehicle

System integration and feed-back within minutes

FMI

Silver

FMI

Virtual ECUs
Simulation of the ECUs, working as in the vehicle

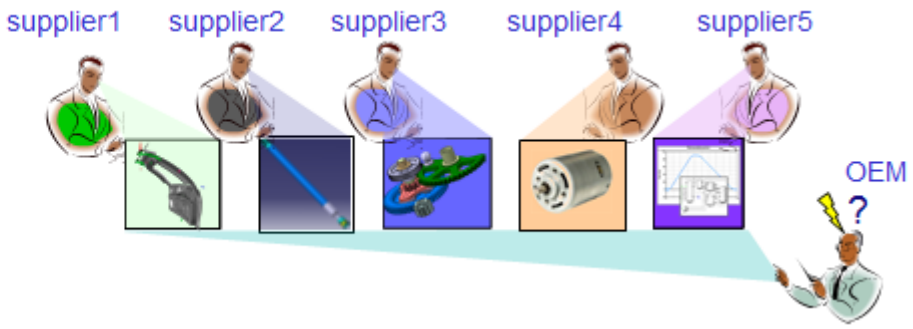
Vehicle Simulation
High-fidelity, configurable

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 over 60 tools (see www.fmi-standard.org for most up to date list)
- Active development as Modelica Association project
- FMI 2.0 just released 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

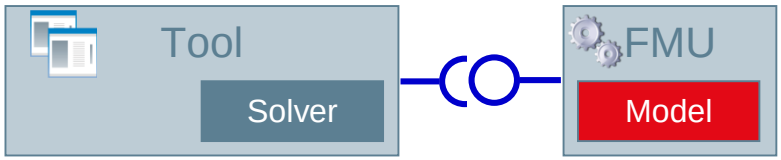
- A component which implements the FMI standard is called *Functional Mockup Unit (FMU)*
- Separation of
  - Description of interface data (XML file)
  - Functionality (C code or binary)
- A FMU is a zipped file (*.fmu) containing the XML description file and the implementation in source or binary form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

Modelon

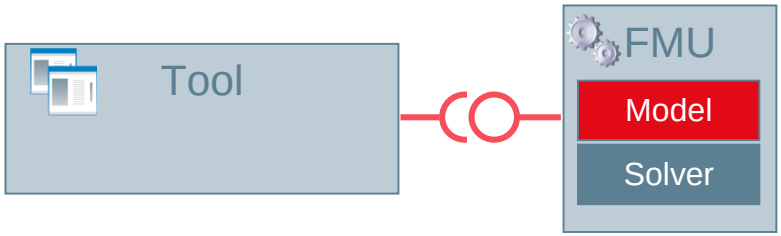
From the official FMI presentation (adapted)

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

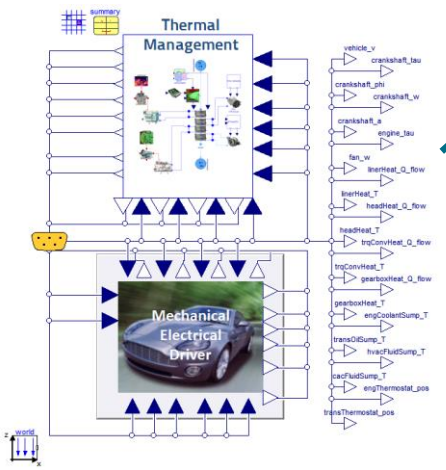
FMI: A BUSINESS MODEL INNOVATION

- FMI-compliant tools often 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)

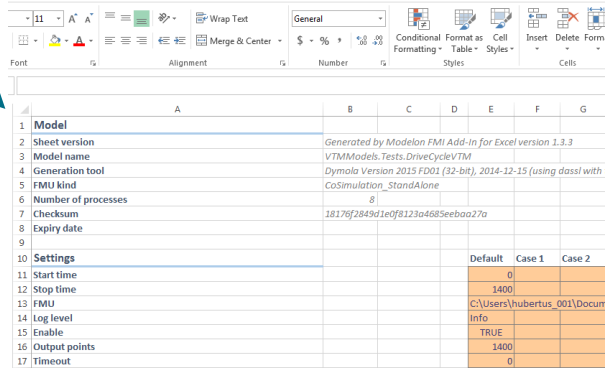
FMI: A BUSINESS MODEL INNOVATION

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)

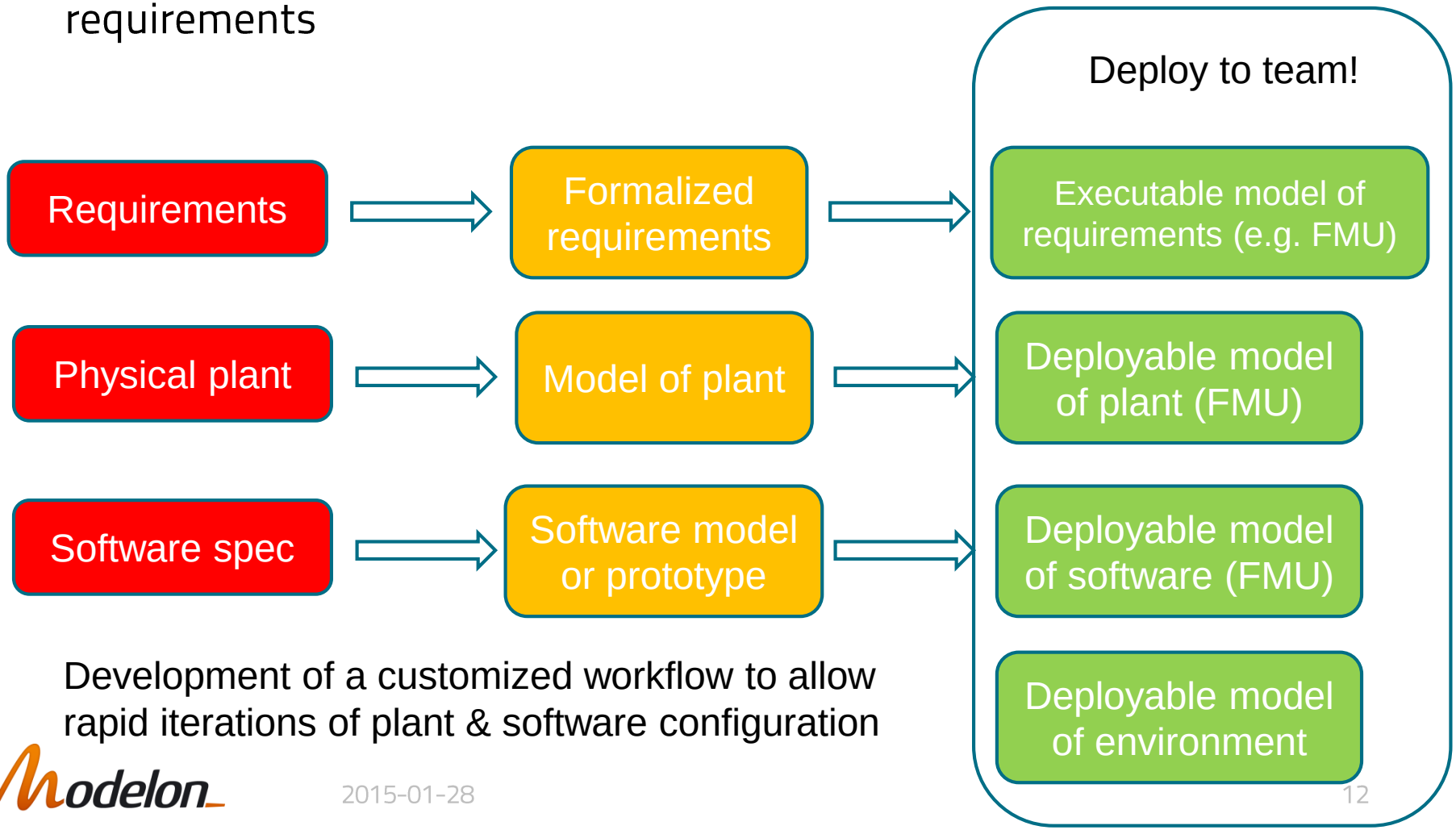
Low-cost Model Execution Platform
May combine FMUs from several tools

- 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

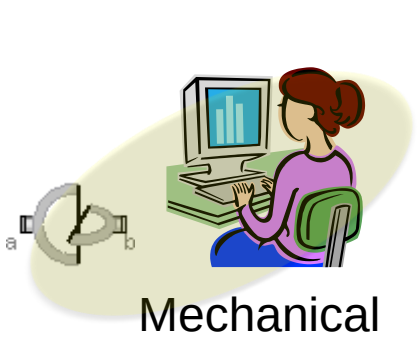
- Systems Engineering centric FMI-based workflow example: automated requirements verification for hardware and software requirements



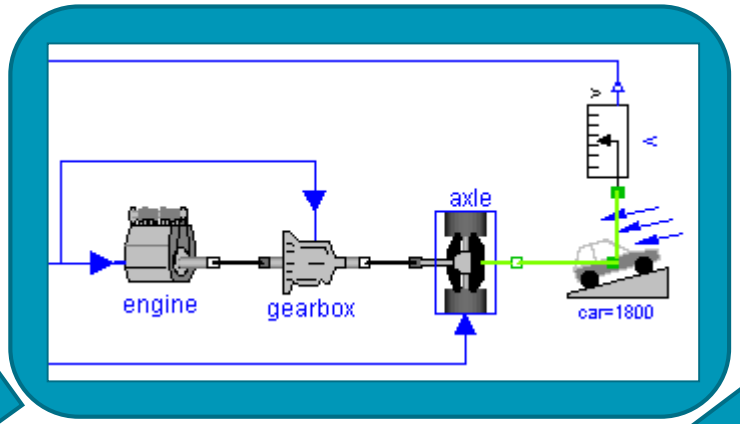
Development of a customized workflow to allow rapid iterations of plant & software configuration

MODEL DEPLOYMENT

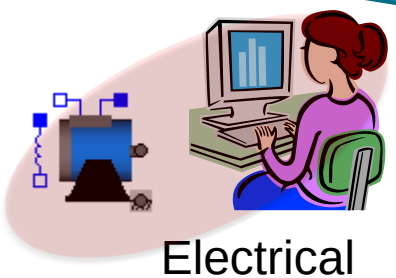
- FMU deployed (native tool) to support multiple applications



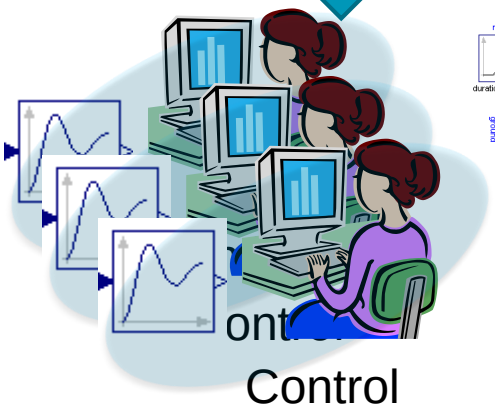
Mechanical



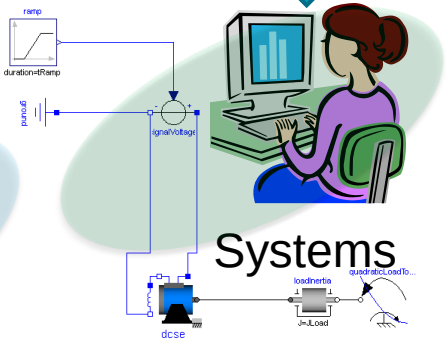
Mechanical System FMU



Electrical



Control

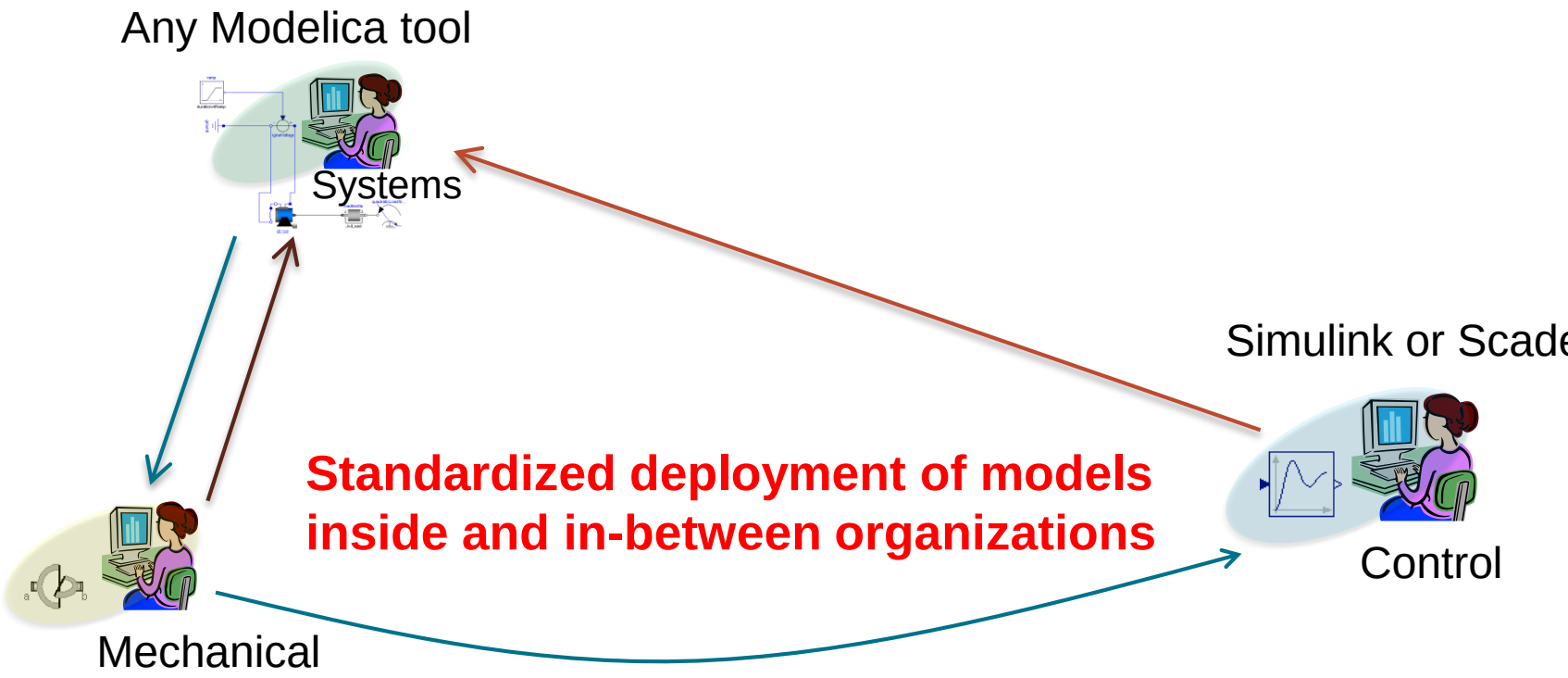


Systems

Thermal

Modelon

ENTERPRISE MODEL DEPLOYMENT

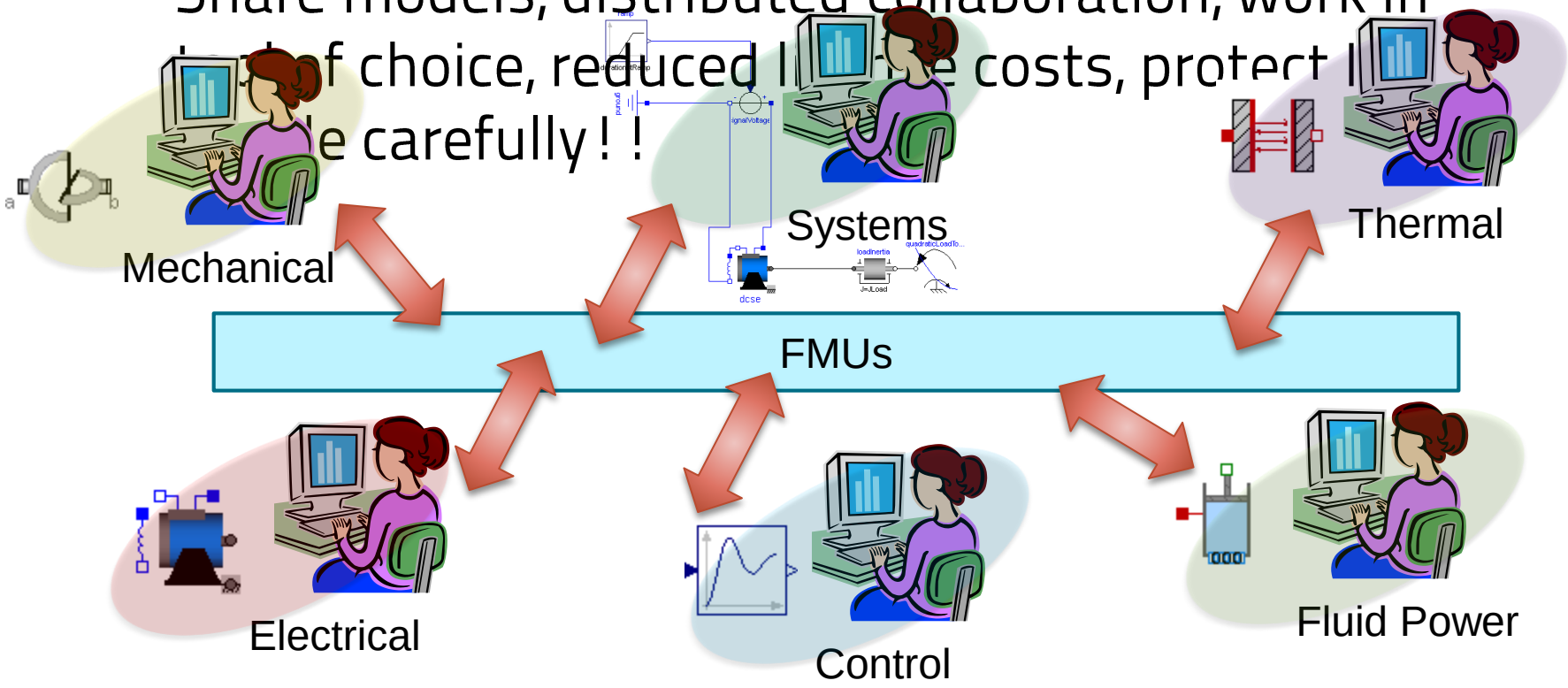


"Daimer, QTronic and Vector describe how Mercedes-Benz currently uses virtual ECUs to validate transmission control software for about 200 variants of the Sprinter series in a highly automated way on Windows PC"

MULTIDOMAN COLLABORATION

with FMUs

- Engineers in different domains work ~~in one formalism/tool~~
  - Share models, distributed collaboration, work in ... of choice, reduced ... costs, protect ... le carefully ! !

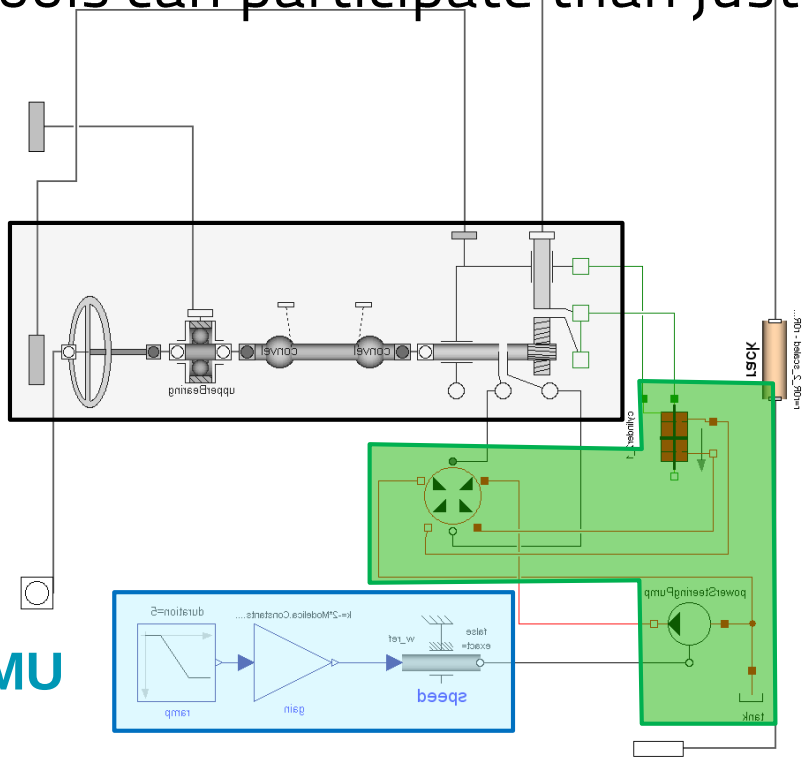


Modelon

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

Mechanical FMU

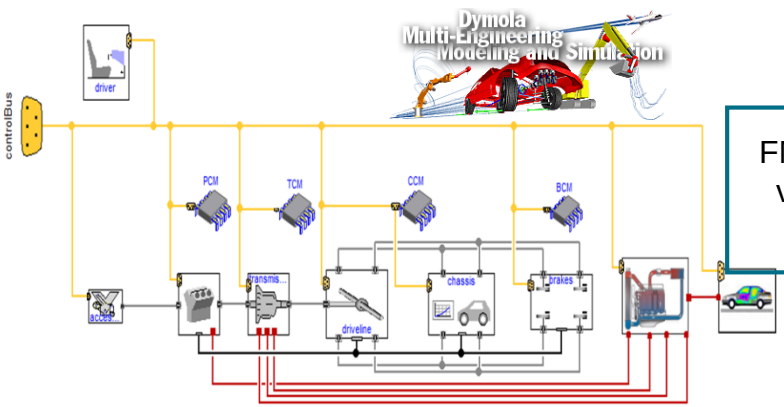


Hydraulics FMU

Control/Actuation FMU

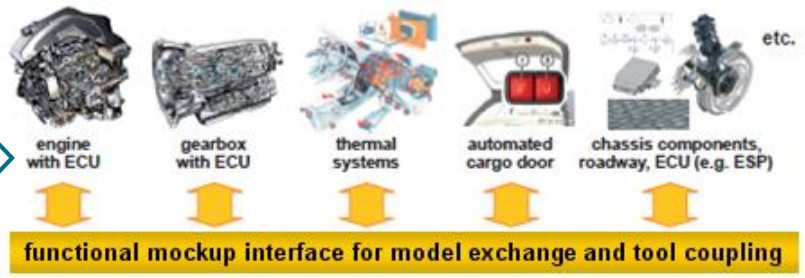
Modelon

DEVELOPMENT TO DEPLOYMENT



FMU Export via Model Export

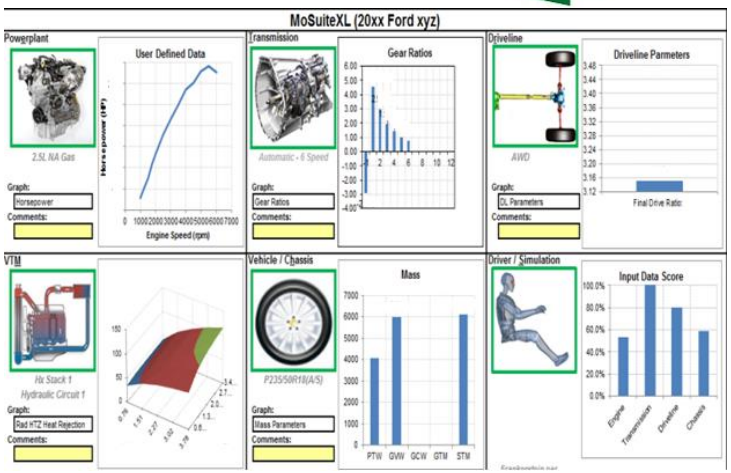
Functional Mockup Interface (FMI)



PyFMI

Custom GUI

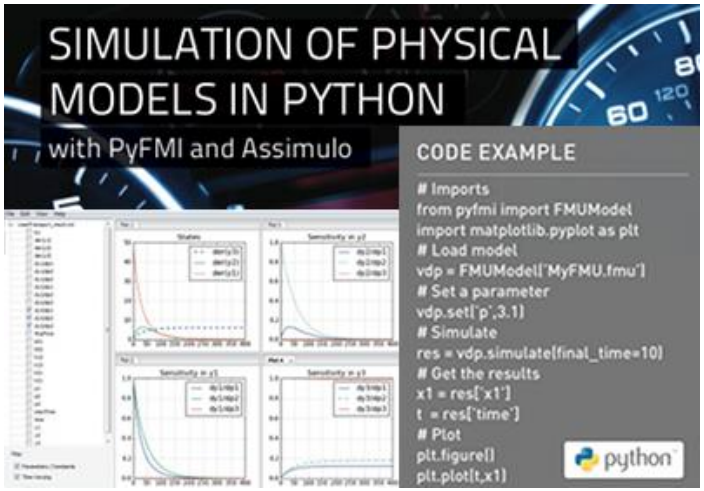
Excel



Parameters

Results

FMU Simulator



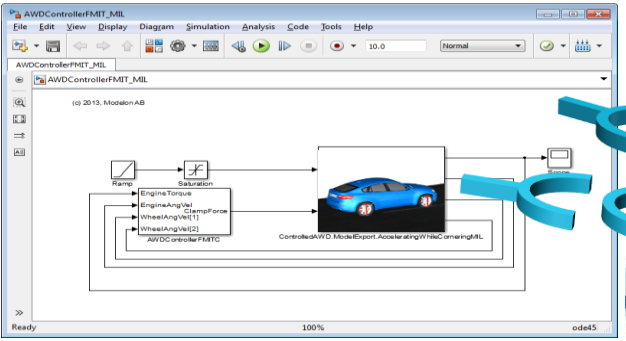
WHAT IS NEW IN FMI 2.0?

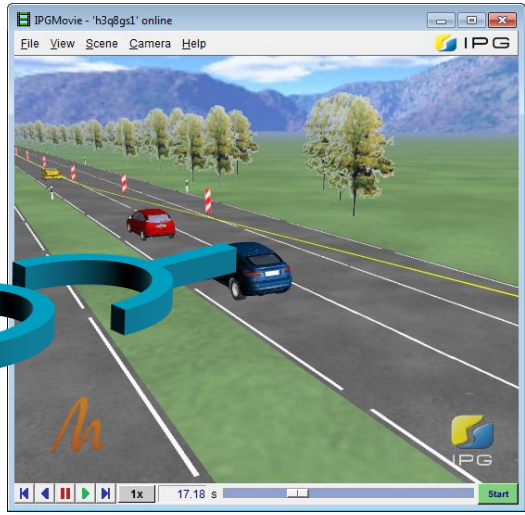
- Unification of the two flavors FMI-CS and FMI-ME
- Clarifications of the specification
- Improvement of the cross-checking protocol
- Interface improvements that allow for much higher quality coupling of simulations
  - Dependency information for inputs/outputs/states
  - Derivatives & analytic Jacobians
  - Combinations of discrete/continuous systems
  - Initialization

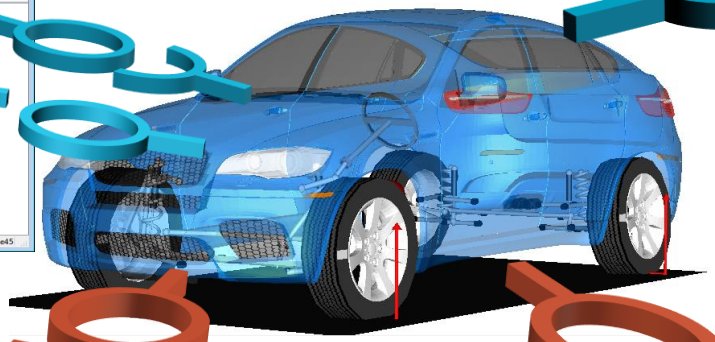
Modelon

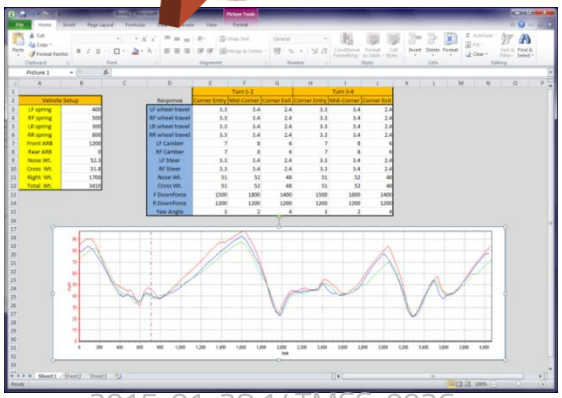
FMI ADVANTAGE

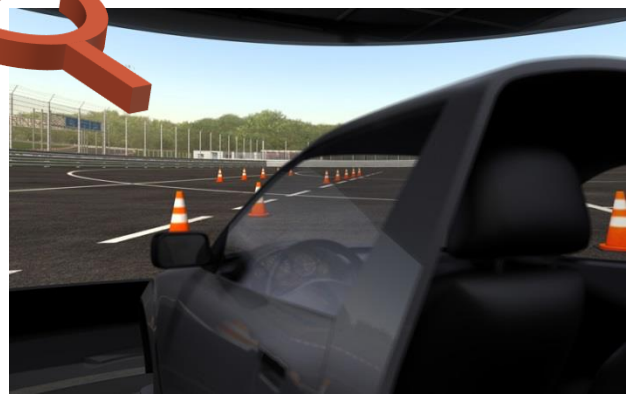
- Same model – different applications





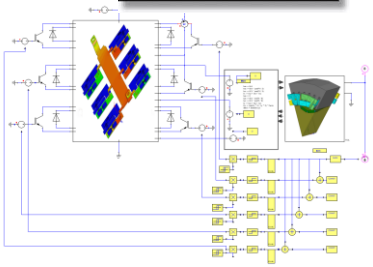




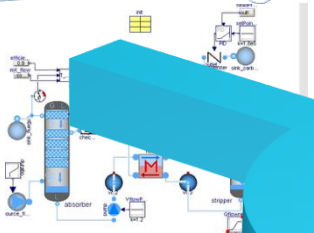


SCADE SUITE™

ANSYS



SIM PACK

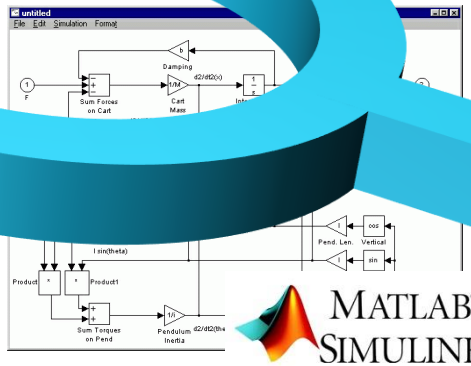


DASSAULT SYSTEMES

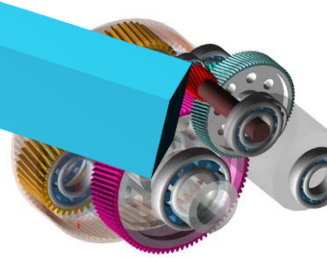
C/C++

Wolfram SystemModeler

AUTOSAR Builder™



QTronic
SIMULATION FOR ENGINEERING

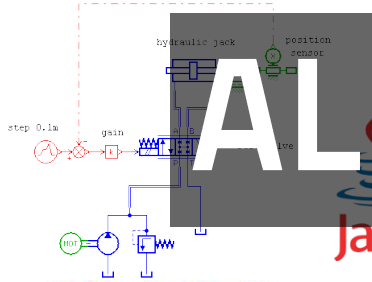


Excel

MATLAB SIMULINK

OpenModelica

IPG AUTOMOTIVE



ALL CONNECTED!

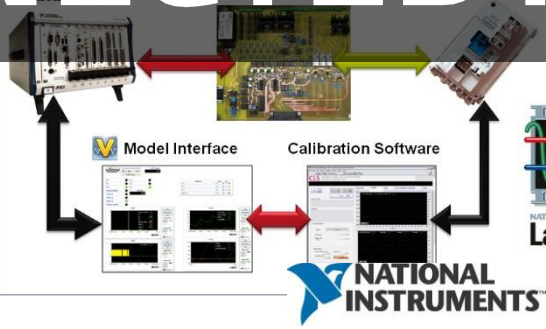
Java

Adams™

MSC Software

LMS
A Siemens Business

SIMULATION X®
Powered by ITI



LabVIEW

python

Modelon

NATIONAL INSTRUMENTS™

MapleSim