The Functional Mockup-Interface:
Innovation through Open Standards

Hubertus Tummescheit, Modelon
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
- System complexity increases
- 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 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

supplier1 supplier2 supplier3 supplier4 supplier5

OEM or systems integrator
FUNCTIONAL MOCKUP INTERFACE (FMI) IN A NUTSHELL

Global Product Data Interoperability Summit | 2017

• What is FMI?
  o an application programming interface and its semantics
  o an xml schema that describes the model structure and capabilities
  o 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.

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

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

Legend

FMI-CS coupling
FMI-ME coupling
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


From the official FMI presentation (adapted)
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)

\[
\text{FMU} = \text{Model w/ Standard Interface}
\]
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

Global Product Data Interoperability Summit | 2017

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

10/18/2016
TYPICAL FMI-BASED WORKFLOWS

Model Authoring Tool(s)

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

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

- True democratization of simulation
- Greatly improved utilization of models

Export: exported FMU freely licensed
EXAMPLE: DEVELOPMENT TO DEPLOYMENT

Functional Mockup Interface (FMI)

FMU Export via Model Export

Functional mockup interface for model exchange and tool coupling

FMU Simulator

Custom GUI

Excel

SIMULATION OF PHYSICAL MODELS IN PYTHON

with PyFMI and Assimulo

CODE EXAMPLE

# Imports
from pyfmi import FMUModel
import matplotlib.pyplot as plt
# Load model
vdp = FMUModel('MyFMI.fmu')
# Set a parameter
vdp.set('p1', 3.1)
# Simulate
results = vdp.simulate(final_time=10)
# Get the results
x1 = results['x1']
# Plot
plt.figure()
plt.plot(x1)

PyFMI
AUTOMATED REQUIREMENTS VERIFICATION

Global Product Data Interoperability Summit | 2017

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

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

Mechanical

Electrical

Control

Systems

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

Combining FMUs with **wisely chosen** boundaries
Any FMI-compliant tool

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

Global Product Data Interoperability Summit | 2017

<table>
<thead>
<tr>
<th>Tools supporting FMI</th>
<th>FMI Version</th>
<th>Export</th>
<th>Import</th>
<th>Slave</th>
<th>Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMI Library</td>
<td>FMI_2.0</td>
<td>Planned</td>
<td>Available</td>
<td>Planned</td>
<td>Available</td>
</tr>
<tr>
<td>FMI Toolbox for MATLAB/Simulink</td>
<td>FMI_2.0</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>FMUSDK</td>
<td>FMI_2.0</td>
<td>Available</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Energy Systems (GES)</td>
<td>FMI_2.0</td>
<td>Planned</td>
<td>Planned</td>
<td>Planned</td>
<td>Planned</td>
</tr>
<tr>
<td>GT-SUITE</td>
<td>FMI_2.0</td>
<td></td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>IGNITE</td>
<td>FMI_2.0</td>
<td>Planned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JavaFMI</td>
<td>FMI_2.0</td>
<td></td>
<td>Available</td>
<td></td>
<td>Available</td>
</tr>
<tr>
<td>JModelica.org</td>
<td>FMI_2.0</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
</tbody>
</table>

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

Global Product Data Interoperability Summit | 2017

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

Brake Model
Modelica

Controller
SIMULINK

Environment Mode
with Sensors
ADTF

Navigation
NAVTEQ

Software
Autosar Builder

Virtual ECU (VAP)

Instrument Cluster – Head up
Display

Universal VAP

OS: RT Linux
BUS: CAN, Ethernet

HW: x86, Nvidia GPU

AUTOSAR
Builder

Virtual ECU (VAP)
ILLUSTRATE A USE CASE

Simulator: birds eye scene view (CarMaker)

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

Development environment for reconstruction the world from raw sensor signal

Raw signal of free range sensor
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
• 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

- 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

Model-based Software Engineering

FMI, CoSim

System Integration
Electric Circuits
Lumped and 1-D
Behavioural Models

VHDL-AMS
Spice
S-Functions
Block-diagrams
State-graphs

ROM, CoSim

Electromagnetic
Fluid
Mechanical
3D Physics-based Models
Thermal

Domain specific Tools

.dll, FMI, CoSim
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