POTENTIAL AND BENEFITS OF FUNCTIONAL MOCK-UP INTERFACE - FMI FOR VIRTUAL VEHICLE INTEGRATION
New Challenges in vehicle development
• Hybrid and electric cars, networking functions...
• Drivetrain Topologies are literally turned upside down
• But usual/available vehicle topology not sufficient

➤ flexible and versatile extension is required

Interaction of components
• System optimization needs to be done in realistic setup (traffic, realistic road, GPS based route planning,...)
• Complex systems with complex dependencies
• Fuel saving techniques make use of all available information e.g. navigation, driver behaviour,...
• But detailed, manual component modeling very costly or not possible

➤ special tools required

Short iterations in the development cycle
• CM allows to combine all aspects: Sensors, Traffic, Driver, Road, Navigation, Interaction/Networking...
• Early validation of design

➤ Virtual Vehicle Integration
• OEMs usually use lots of different tools for detailed analysis of various system components
• Co-Simulation quite common requirement
• No Standard for Co-Simulation or Model Interfacing
• But lots of proprietary interfaces:
  - Simulink: S-function
  - Modelica interface: external function, external object
  - SimulationX: External Model Interface
  - LabVIEW: External Model Interface Simulation Interface Toolkit
  - Simpack: uforce routines
  - ADAMS: user routines
  - ...
The FMI development is part of the ITEA2 MODELISAR project (2008 - 2011; 29 partners, Budget: 30 Mill. €)

FMI development initiated, organized and headed by Daimler AG

Improved Software/Model/Hardware-in-the-Loop Simulation, of physical models from different vendors.

Open Standard

14 Automotive Use-Cases to evaluate FMI.

Further Information: www.modelisar.com
The Functional Mockup Interface defines

(a) C-header files to interact with the equations of a model or to perform co-simulations with other simulators and

(b) XML schema files to inquire information about model and interface variables.

The Functional Mockup Interface definition is one result of the ITEA2 project MODELISAR. The intention is that dynamic system models of different software systems can be used together for software/model/hardware-in-the-loop simulation and for embedded systems.”

Source: www.modelisar.com
A component which implements the interface 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

Interface specification: www.functional-mockup-interface.org
FMI enables

• High level modeling with specialized tools (instead of handcoding)

• Black box models for IP protection (suppliers can provide component models)

• Detailed studies, evaluation and analysis of components or systems in early development stage

• Identification and analysis of interdependencies at early stage

• Transfer experiment and testing into simulation by providing detailed models

• Flexible and automatic extension of CM Vehicle hierarchy

• Seamless integration of complex models into CM ModelManager
XML - Model description allows:

- Component type checking
- Automatic component parameter update
- Automatic and customized signal propagation
- User comfort functions like e.g. preparation of FMU-specific Infofiles for batch parameterization
Multi-Physics-Modeling within Dymola

- Composition of e.g. electric, mechanic or thermal models
- Arbitrary driveline architectures (conventional, electric, hybrid)

example DC machine
Library Concept

- Use existing Libs and Models (VDL, PowerTrain, ...)
- Develop own Libraries tailored to suit your requirements
- BlackBox FMUs for IP protection of suppliers

VDL example driveline
BENEFITS FOR CARMaker - SCHEME

**FMI for ModelExchange**

- $t_0, p$, initial values (a subset of $v(t_0)$)
- time
- discrete states (constant between events)
- parameters of type Real, Integer, Boolean, String
- inputs of type Real, Integer, Boolean, String
- all exposed variables
- continuous states (continuous between events)
- outputs of type Real, Integer, Boolean, String
- event indicators

**External Model (FMU Instance)**

- $t$
- $x$
- $x, z$

**Solver**

---

**FMI for CoSimulation**

- $t, p, v_0$
- time
- discrete states (constant between events)
- parameters of type Real, Integer, Boolean, String
- inputs of type Real, Integer, Boolean, String
- all exposed variables
- continuous states (continuous between events)
- outputs of type Real, Integer, Boolean, String
- event indicators

**External Model (FMU Instance)**

- $t$
- $x$
- $x, z$

**Solver**

**Co-Simulation Slave (FMU Instance)**

---

Source: FMI – standard specification
BENEFITS FOR CARMAKER - OPTIONS

FMI for ModelExchange

- Described by differential-, algebraic-, discrete equations, with time-, state, and step-events
- No computational overhead by solver – very fast
- Requires stable/robust model (especially for initialization)

FMI for CoSimulation

- Data exchange is restricted to discrete communication points
- Subsystems are solved independently between communication points
- Simulation of heterogeneous systems
- Partitioning and parallelization of large systems
- Derivatives of inputs, outputs w.r.t. time can be set/retrieved for supporting of higher order approximation
- allows use of advanced solvers and therefore fast system dynamics
- Comp. Robustness
- Enables complex models, fast dynamics and Event handling

Source: www.modelisar.com
BENEFITS FOR CARMaker - CHALLENGE

MODELING CONCEPTS

Signal based/procedural
- Order of calculation is set
- Can always be calculated
- Easy access and model comprehension
- Full control of model equations

Physical/acausal
- Acausal physical modeling (equation is interpreted as such, not as assignment)
- System is largely defined by boundary conditions not just model
- Requires continuous signals as modeling is based in physical equations (algebraic and differential)
- Automatic manipulation of obtained equation system
Iterative
• Time integration on predefined grid
• States and derivatives can therefore easily be achieved at grid points
• Signals are piecewise constant
• Fast integration
• Basis for HIL-use

Continuous
• Adaptive stepsize control possible
• Requires continuous signals for states and derivatives
• States and derivatives of states must match at communication points
• Fast dynamics possible
• Enables event handling
- Requires the user to understand and consider both tool concepts
- Advantages and drawbacks of both concepts must be taken into account to achieve effective simulation
- Enables the tool developer to exploit the advantages of both concepts
- Despite complex models with fast dynamics high performance can be achieved
- State signals for communication should be chosen wisely
- ME and CS option offer high flexibility
  - ModelExchange allows high performance but requires robust modeling
  - CoSimulation enables easy access, robust simulation (also for stiff systems) and proper event handling
• MAN Truck&Bus AG evaluated FMI based interface with TruckMaker
• Modelon GmbH provided FMI based software interface to TruckMaker
• Goal: verify if TruckMaker FMI extension meets MANs tool requirements.
• Object: Conventional reference drivetrain
• Criteria:
  - Accessibility of tool
  - Ease of use
  - Performance of simulation
  - Quality of solution
• Tool choice to provide FMUs: Dymola
• Conventional drivetrain model as basis
• Methodology in focus
• Key aspect was to verify the toolchain
• Derived method serves as prototype for further developments
• Basis for detailed model integration and functional extension
• TruckMaker OpenXWD interface powertrain
Components:

- Table-based engine with starter
- Clutch model with friction loss calculation
- 12-speed gearbox with gear depended efficiency
- Rear axle with differential
- OpenXWD drivetrain (no tire, no brakes modeled)
Model integration required the following adaptations:

- Additional, required signals must be provided, e.g. PT.Engine .rotv for Driver.
- Proper vehicle configuration must be provided (e.g. number of gears, idle engine speed,...) to match TruckMaker component requirements.
- Manual shifting for Open XWD had to be activated.
- Starting up was implemented via Mini-Maneuver "Kl15" und "Kl50".
- Driver parameterization done via driver adaption to obtain proper shifting behaviour.
Comparison of TruckMaker default drivetrain with Dymola Model

- Good compliance due to similar models
- Dymola model based on mechanical components for conventional drivetrain with loss

Engine speed

Gear number

Vehicle speed
Comparison of detailed reference drivetrain with simplified model

- Reference resembles high detailed and validated longitudinal dynamics model
- Different behaviour due to strongly simplified Turbocharger model
Insights and conclusion for project partner MAN Truck & Bus AG

• CoSimulation yielded instant results, ModelExchange instable at initialization for this model.
• Replacing the default model by the more detailed and lossy Dymola (FMI) model was easily achieved
• Proper parameterization of the driver is crucial to achieve realistic shifting behaviour
• Realistic results especially in terms of power and consumption
• Combined benefits of Dymola modeling (via FMI) and CarMaker at hand: physical modeling in real-time capable environment with driver and traffic
• The interfacing tool met desired requirements
• Project will be continued and enhanced to investigate more detailed models
FMI and tool future

- Steadily growing support of FMI in industry and user community
- Standard is being maintained and developed, Version 2.0 in beta status available, Key Features:
  - Unified standard
  - Improved event handling
  - Enhanced treatment of Jacobian matrices
  - ...
- Commercial product Modelon FMI Toolbox for Car/TruckMaker available by end of 2012.
  For further information please contact: sales@modelon.com or visit: http://www.modelon.com/products/fmi-toolbox-for-carmaker/
- DEMO at Appy&Innovate exhibition