

Tutorial:

Functional Mockup Interface 2.0 and HiL Applications

Torsten Blochwitz
R&D Manager, ITI Dresden

FMI MAP Leader

Agenda

FMI at Automotive OEMs and Suppliers

	Automotive Use Cases, Workflows, HiL	S. Schneider (BMW)
	FMI Applications at Daimler	B. Relovsky (Daimler)
	Benefits of using Standards for Model Exchange	M. Kuebler (ZF)

FMI in Hardware-in-the-Loop (HiL) applications

	FMI in CarMaker HiL	R. Pfeffer (IPG)
	FMI in LabCar HiL	C. Mitrohin (ETAS)
	FMI Support in the dSPACE Toolchain	A. Pillekeit (dSPACE)
	FMI in Concurrent Real-Time HiL	R. Praveenkumar (Concurrent RT)
	FMI in dSPACE DS1006 HiL	J. Akesson (Modelon)

Agenda

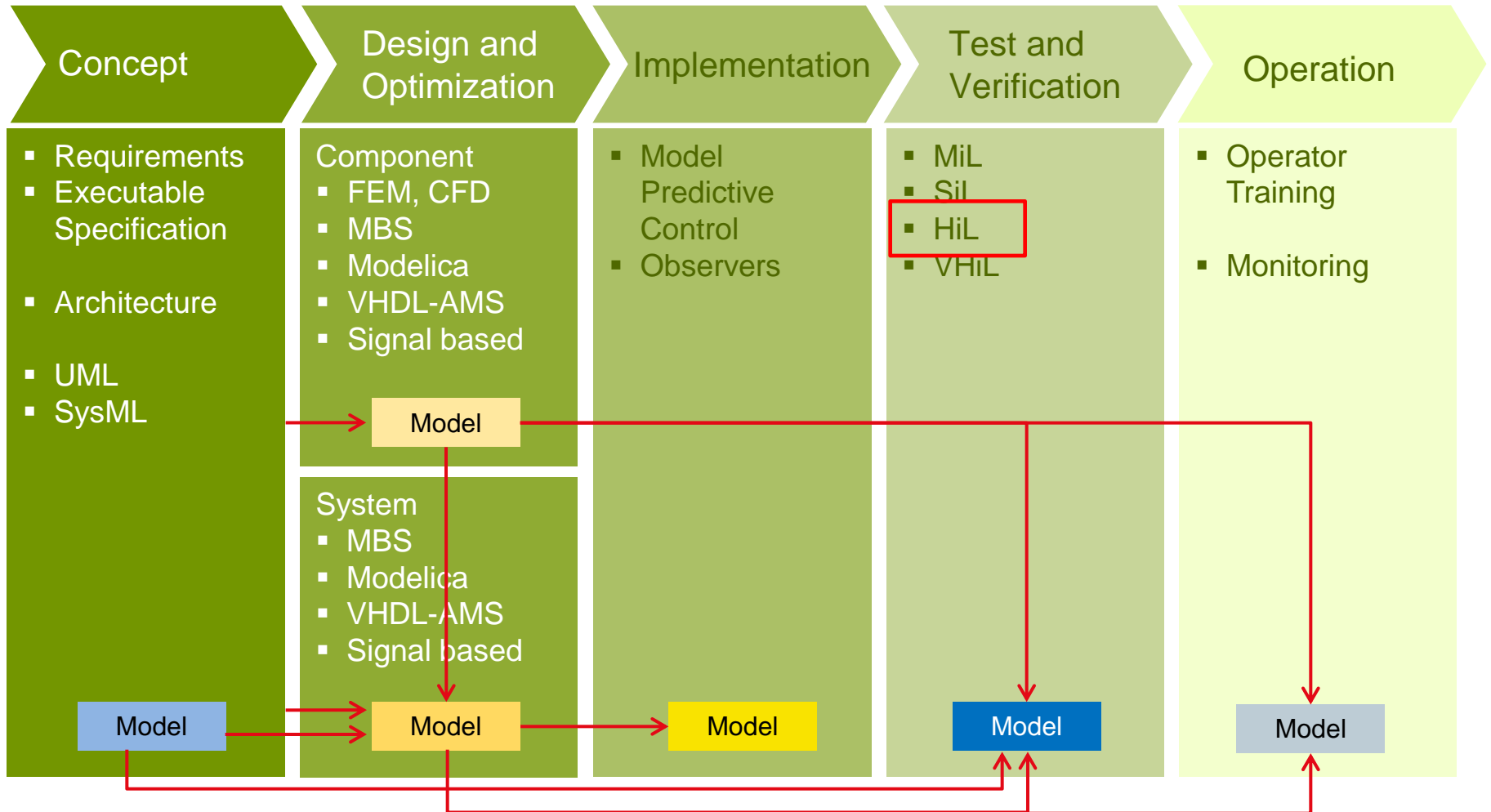
FMI 2.0 Features

	New Features of FMI 2.0 and beyond	T. Blochwitz (ITI)
	Connecting tightly coupled FMUs	M. Otter (DLR)
	Modelica FMI Test Library	M. Otter (DLR)
	Co-Simulation with variable step-size	H. Elmqvist (Dassault Systèmes)

FMI 2.0 Demonstrations

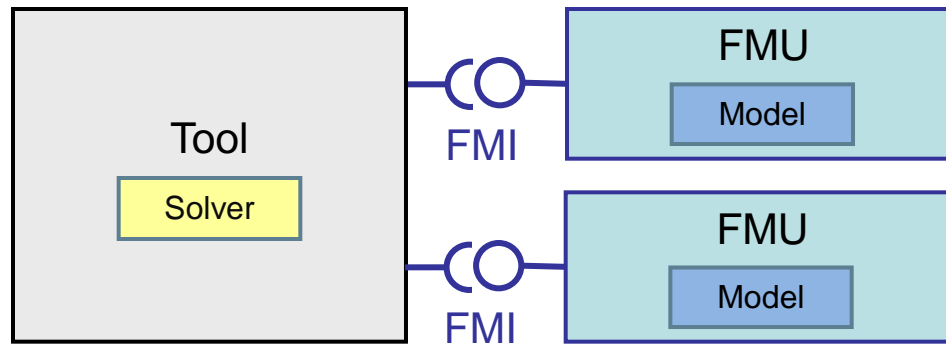
	FMU Compliance Checker	J. Akesson (Modelon)
	FMUs from Dymola and SimulationX on dSPACE SCALEXIO HiL	A. Pillekeit (dSPACE), et.al.
	FMUs from Dymola and AUTOSAR Builder on Concurrent RT platform	H. Elmqvist (Dassault Systèmes), et.al.

Introduction

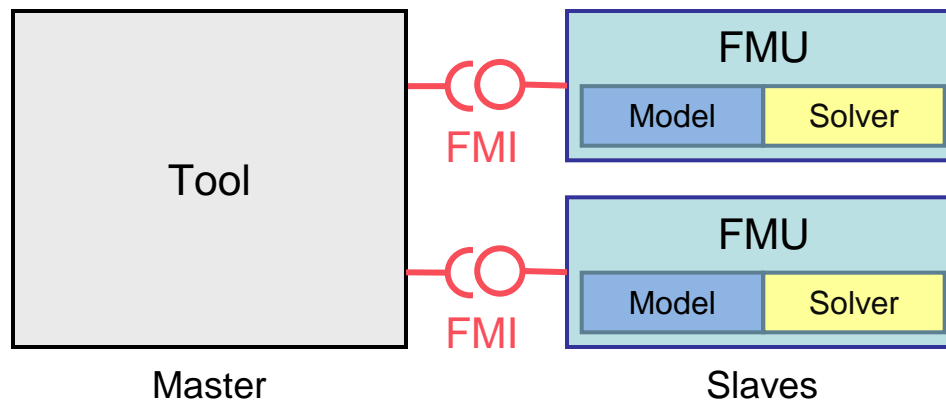


FMI – Main Design Idea

- FMI for Model Exchange

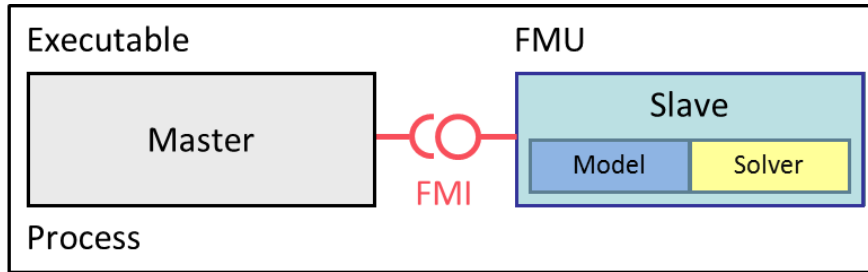


- FMI for Co-Simulation

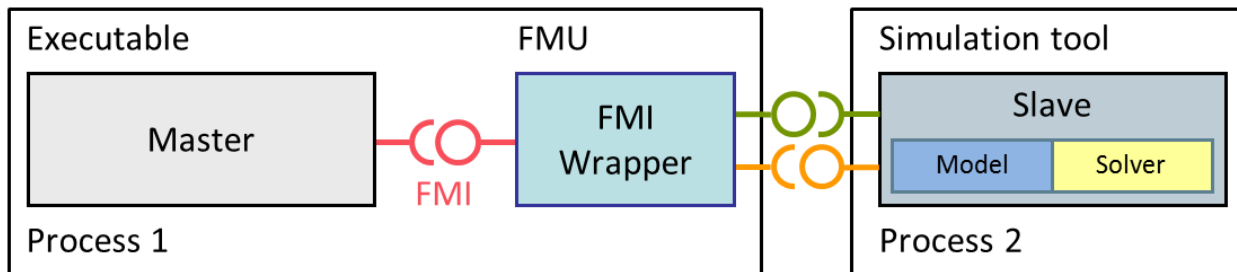


FMI Use Cases

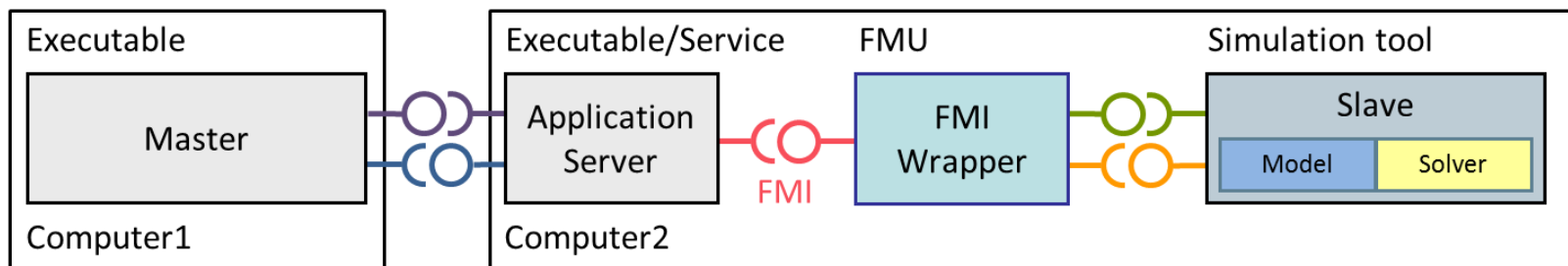
- Standalone:



- Tool Based:

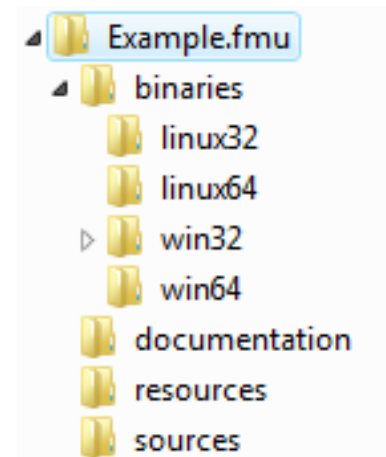


- Distributed:

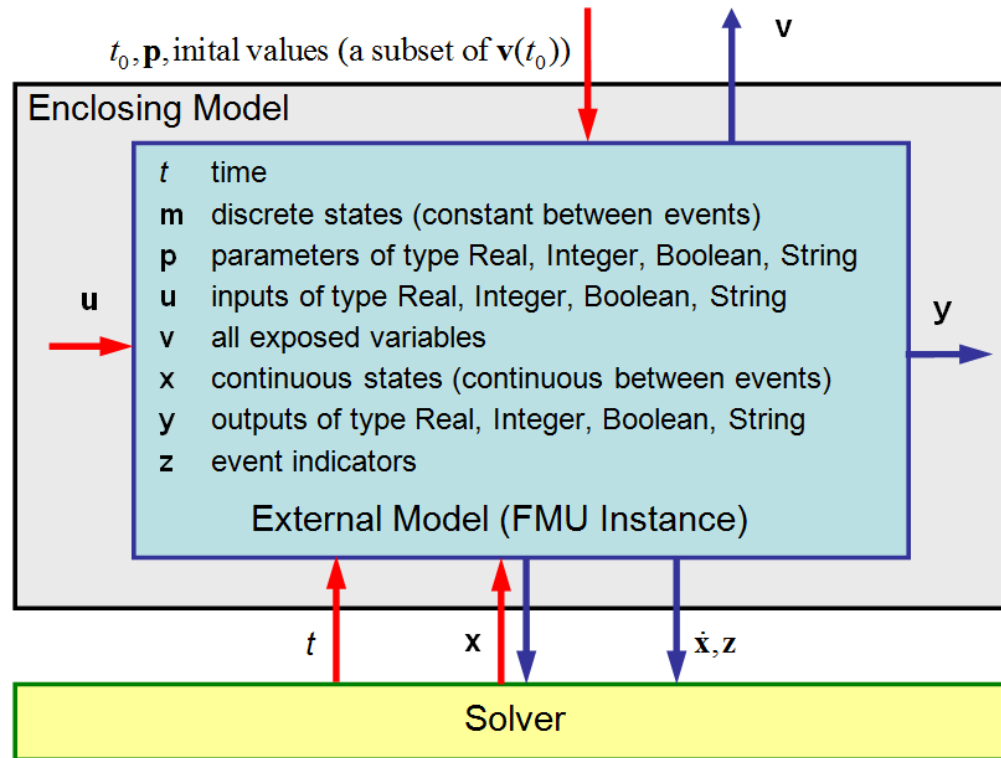


FMI – Main Design Idea

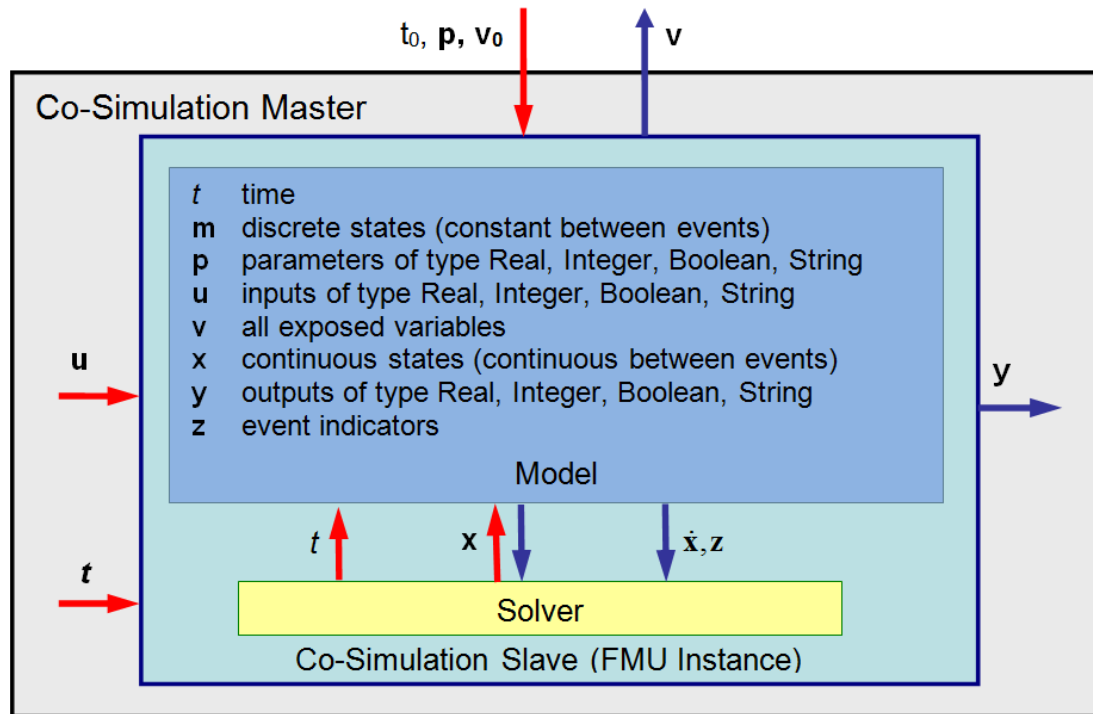
- A component which implements the interface is called a *Functional Mockup Unit (FMU)*
- Separation of:
 - Description of interface data (XML file)
 - Functionality (API in C)
- An FMU is a zipped file (*.fmu) containing:
 - modelDescription.xml
 - Implementation in source and/or binary form
 - Additional data and functionality
- One FMU can contain implementations of both interfaces



FMI for Model Exchange Signals



FMI for Co-Simulation Signals



Additional:

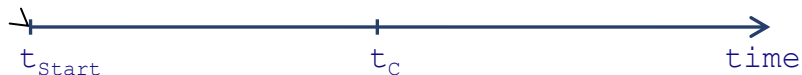
- Status information
- Derivatives of inputs, outputs w.r.t. time for support of higher order approximation between communication steps

FMI for Model Exchange and Co-Simulation

Sample Code

- Model Exchange:
(One model evaluation)

```
/* Set inputs*/  
fmiSetReal(m, id_u, u, nu);  
fmiSetTime(m, tC);  
fmiSetContinuousStates(m, x, nx);  
/* Get results */  
fmiGetDerivatives(m, derx, nx);  
fmiGetEventIndicators (m, z, nz);  
fmiGetReal(m, id_y, y, ny);
```



- Co-Simulation:
(One communication step)

```
/* Set inputs*/  
fmiSetReal(s, id_u, u, nu);  
/* Do computation*/  
fmiDoStep(s, tC, hC, fmiTrue);  
/* Get results */  
fmiGetReal(s, id_y, y, ny);
```

