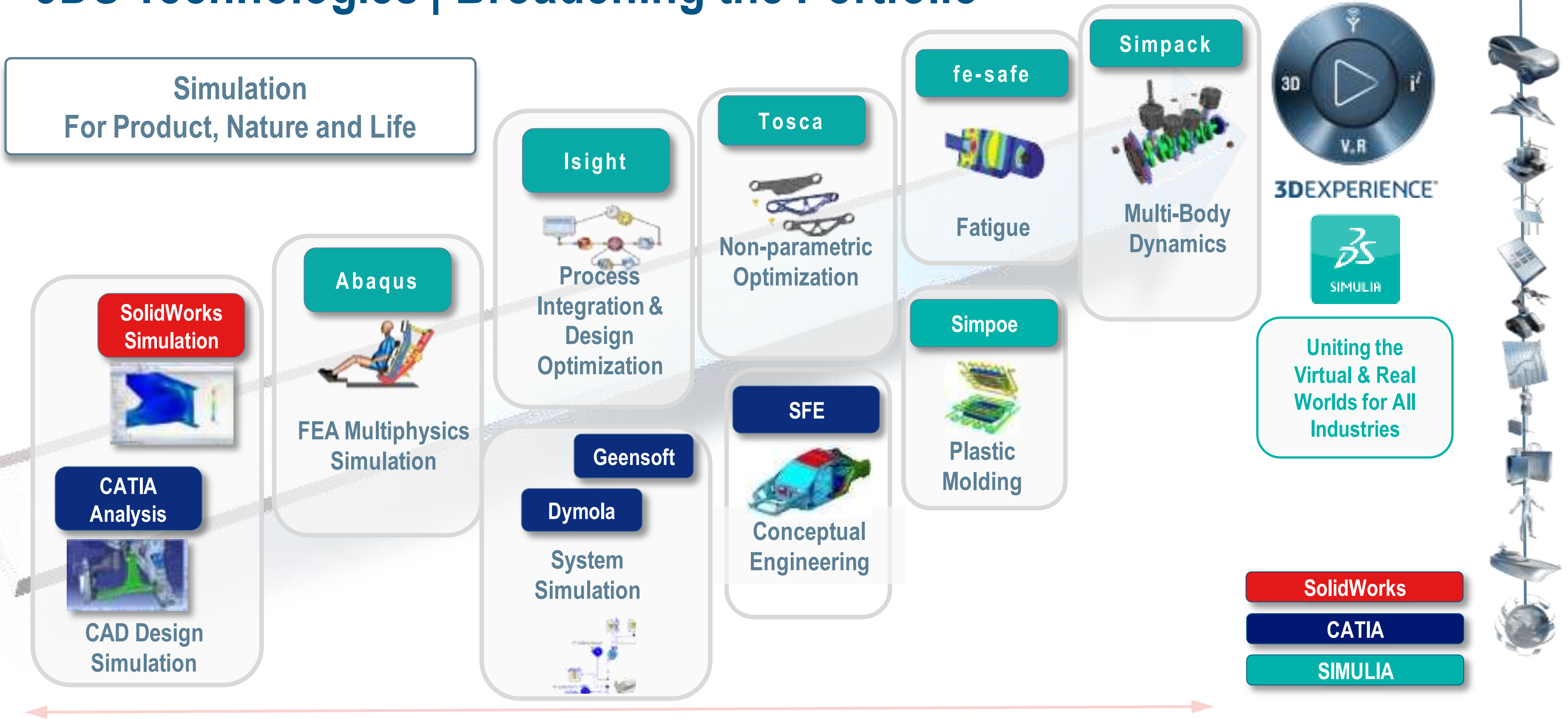
MODELICA / DYMOLA and SIMULIA Abaqus working together

Tomáš Ivančík
SIMULIA Sales
tomas.ivancik@3ds.com

MODELICA seminář 2015
26.6.2015

3DS Technologies | Broadening the Portfolio



Simulation
For Product, Nature and Life

SolidWorks Simulation

CATIA Analysis

CAD Design Simulation

Abaqus

FEA Multiphysics Simulation

Isight

Process Integration & Design Optimization

Geensoft

Dymola

System Simulation

Tosca

Non-parametric Optimization

SFE

Conceptual Engineering

fe-safe

Fatigue

Simpoe

Plastic Molding

Simpack

Multi-Body Dynamics

3DEXPERIENCE

Uniting the Virtual & Real Worlds for All Industries

- 🟥 SolidWorks
- 🟦 CATIA
- 🟩 SIMULIA

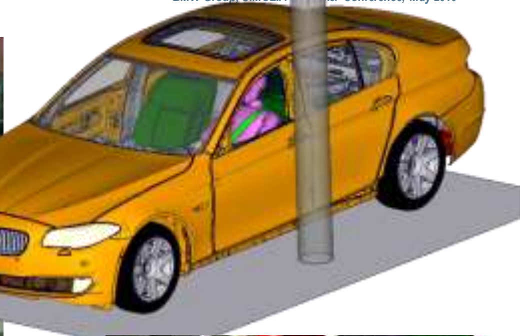
IF WE ask the right questions we can change the world

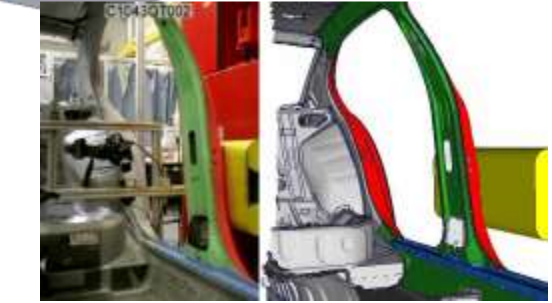
What is Realistic Simulation?

“Realistic Simulation” is a simulation that is physically realistic and “life like” in every way

Courtesy Mechanical Design and Analysis Corporation, 2010 SCC

“Predictive Crashworthiness Simulation in a Virtual Design Process without Hardware Testing”, Jurgen Lescheticky, Hariaokto Hooputra and Doris Ruckdeschel, BMW Group, SIMULIA [illegible] Conference, May 2010





Courtesy of BMW Group, 2010 SCC

Statistical distribution of impact damage

Courtesy of University of Zagreb, 2010 SCC

Why simulate instead of test?

Physical Testing

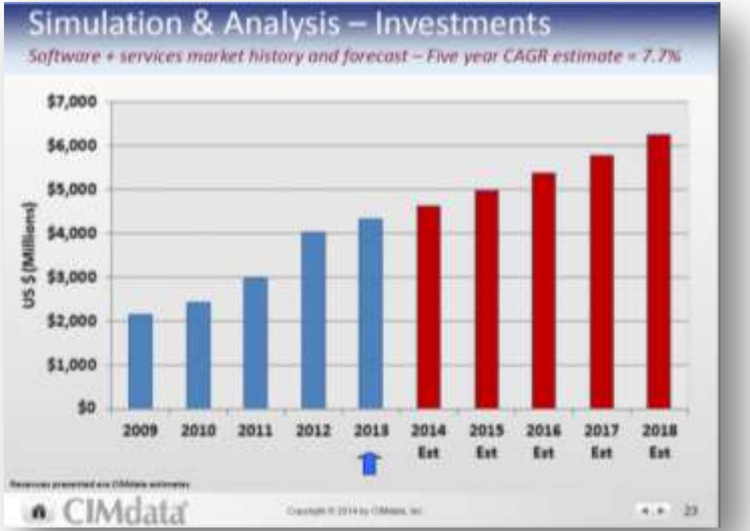
- Very expensive (often $100,000/test)
- Time-consuming (weeks or months)
- Can only be done in an expensive lab
- Requires exotic equipment
- Can only be done at certain times
- Says *what* happened (“pass/fail”), but not *why*

Virtual Testing (Realistic Simulation)

- Cheap ($500/simulation-result)
- Quick (hours)
- Can be done anywhere
- Uses only a computer and software
- Can be done 24/7/365
- Says *what* happened AND *why*

Simulation & Analysis is a Central, Up-front Role

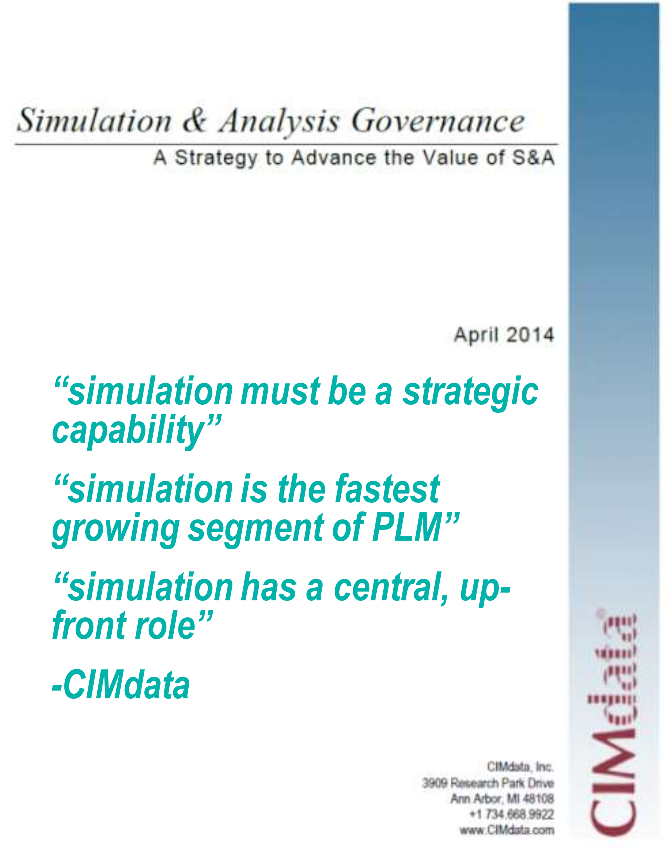
It must be… recognized and managed as a strategic capability…



Simulation & Analysis – Investments

Software + services market history and forecast – Five year CAGR estimate = 7.7%

Courtesy CIMdata



Simulation & Analysis Governance

A Strategy to Advance the Value of S&A

April 2014

"simulation must be a strategic capability"

"simulation is the fastest growing segment of PLM"

"simulation has a central, up-front role"

-CIMdata

CIMdata, Inc.
3909 Research Park Drive
Ann Arbor, MI 48108
+1 734.668.9922
www.CIMdata.com

Living Heart Project

- Simulation is at the forefront of a revolution in cardiac care

- Visit the Living Heart Project in 3D

The Living Heart

SIMULIA Spearheads The Living Heart Project

Bringing the medical community together for improved patient care

3DXcite
Realtime Deltagen Visualization

SIMULIA
COMMUNITY NEWS

IN HARMONY
SIMULATING PRODUCT, NATURE, AND LIFE

SIMULIA
Community News
May 2014

Metóda konečných prvků – MKP (FEM)

...je numerická metoda sloužící k simulaci průběhů napětí, deformací, vlastních frekvencí, proudění tepla, jevů elektromagnetismu, proudění tekutin atd. na vytvořeném fyzikálním modelu. Její princip spočívá v diskretizaci spojitého kontinua do určitého (konečného) počtu prvků, přičemž zjišťované parametry jsou určovány v jednotlivých uzlových bodech.

Solutions for the entire range of industries

Consumer Packaged Goods

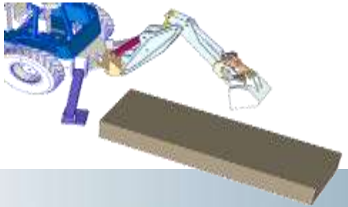
Plastic and Glass Forming
Conveyor Systems
Container Drop
Pressure Analysis
Thermal Analysis
Bottle Sealing
Adhesives

Automotive & Transportation

Chassis
Body
Tires
Interiors
Crashworthiness
Brake Systems
Powertrain
Electronics

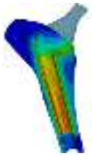
Industrial Equipment

Nonlinear Stress Analysis
Thermal Analysis
Cyclic Loading
Flexible Multibody Dynamics
Soil-Structure Interaction



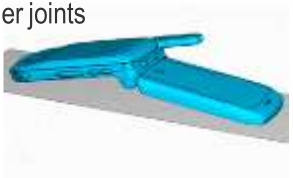
Life Sciences

Tissue Modeling
Surgical Equipment
Stents
Drug Delivery
Orthopedics
Medical Packaging



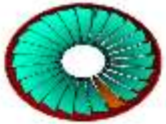
High Tech

Thermal cycling of solder joints
Drop Testing
Vibration Analysis
Semiconductors
Circuit Boards
Hand-held Devices
Computers & Peripherals



Aerospace & Defense

Avionics
Landing Gear
Aerostructures
Aeroengines
Composites
Defense Systems
Space Systems

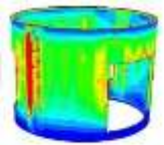


Architecture & Construction

Earthquake loading
Structural integrity due to fire
Concrete analysis
Soil-pore interaction
Failure limits

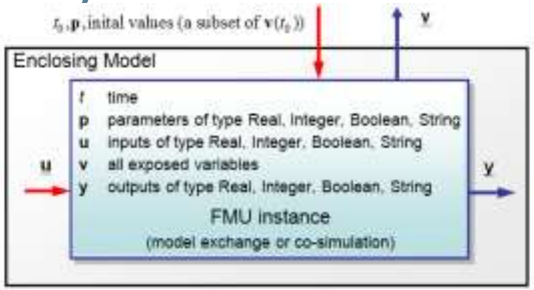
Energy

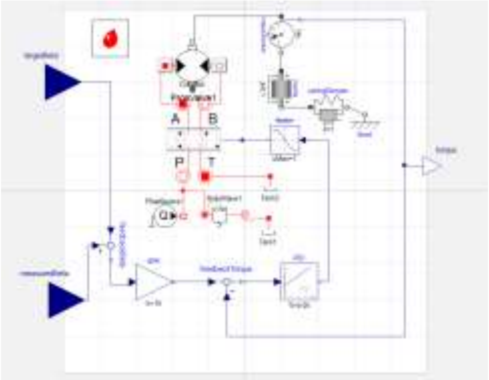
Wave loading on rigs and pipelines
Piping and pressure vessels
Thermal analysis
Blast loading
Drop or Impact



What is a Functional Mockup Unit (FMU)?

- A self-describing simulation model
  - Adhering to an open standard titled Functional Mockup Interface (FMI)
- With scalar input and output variables
- Commonly provided in file format (e.g. mymodel.fmu)
- Conceptually similar to
  - User subroutines
  - Dymola DLL files used in the Abaqus / Dymola interface
  - Substructure / superelement files





Dymola component

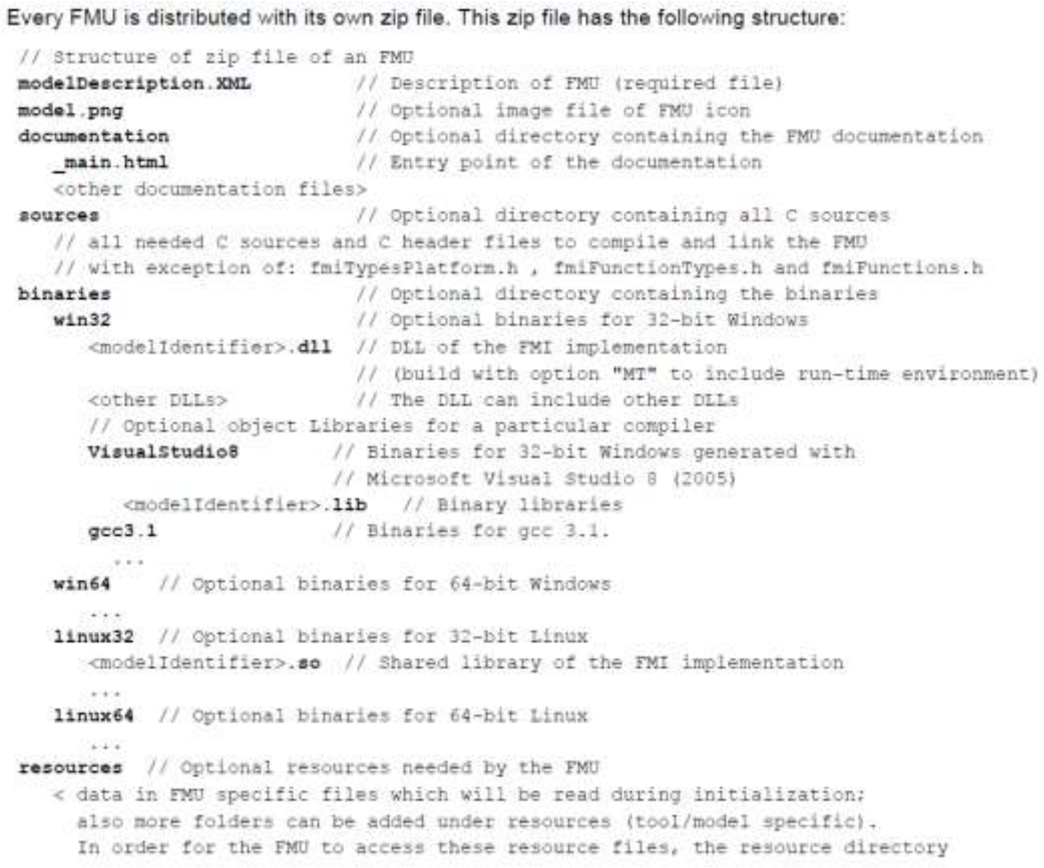
What is a Functional Mockup Unit (FMU)?

- Two general categories of FMUs are available
  - Model Exchange format – the FMU provides equation information
    - More difficult to handle
    - Easier to generate
  - Co-Simulation format – the FMU is capable of performing its own time integration
    - Easier to handle
    - More difficult to generate
- We will discuss only the Co-Simulation format

What is a Functional Mockup Unit (FMU)?

File format (e.g. mymodel.fmu)

- Important components
  - modelDescription.xml
    - Describes the model, including
      - Input / output variables
      - Parameters available
      - Unit definitions
  - Shared libraries
  - Source files



Every FMU is distributed with its own zip file. This zip file has the following structure:

```
// Structure of zip file of an FMU
modelDescription.XML         // Description of FMU (required file)
model.png                    // Optional image file of FMU icon
documentation                // Optional directory containing the FMU documentation
   _main.html                // Entry point of the documentation
   <other documentation files>
sources                      // Optional directory containing all C sources
   // all needed C sources and C header files to compile and link the FMU
   // with exception of: fmiTypesPlatform.h , fmiFunctionTypes.h and fmiFunctions.h
binaries                     // Optional directory containing the binaries
   win32                     // Optional binaries for 32-bit Windows
      <modelIdentifier>.dll  // DLL of the FMI implementation
                             // (build with option "MT" to include run-time environment)
      <other DLLs>           // The DLL can include other DLLs
      // Optional object Libraries for a particular compiler
      VisualStudio8        // Binaries for 32-bit Windows generated with
                           // Microsoft Visual Studio 8 (2005)
         <modelIdentifier>.lib   // Binary libraries
      gcc3.1               // Binaries for gcc 3.1.
        ...
   win64    // Optional binaries for 64-bit Windows
      ...
   linux32  // Optional binaries for 32-bit Linux
      <modelIdentifier>.so  // Shared library of the FMI implementation
      ...
   linux64  // Optional binaries for 64-bit Linux
      ...
resources  // Optional resources needed by the FMU
   < data in FMU specific files which will be read during initialization;
     also more folders can be added under resources (tool/model specific).
     In order for the FMU to access these resource files, the resource directory
```

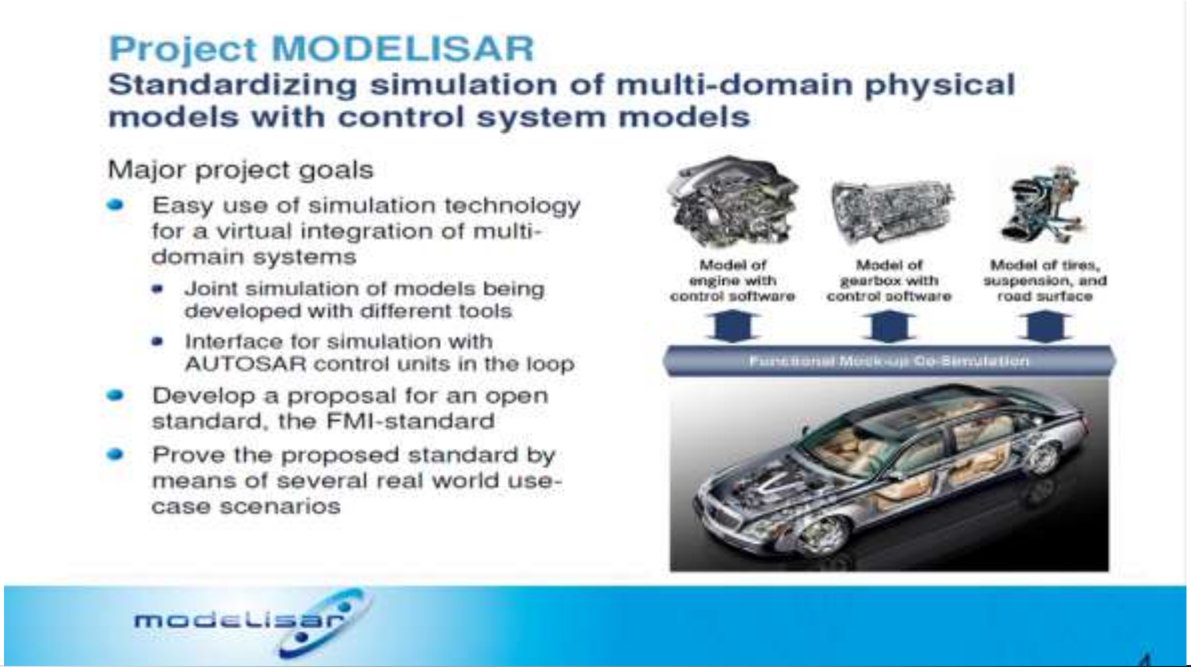
FMU Attractions

- Open standard
- Increasing vendor adoption
- Intellectual property hiding

Tools supporting FMI	Model-Exchange		Co-Simulation	
	Export	Import	Slave	Master
Adams		Planned	Available	Available
AMESim	Available	Available	Available	Planned
ANSYS Simplorer		Planned	Planned	
ASim - AUTOSAR Simulation	Available		Available	
Atego Ace		Available		Available
@Source	Available			
Building Controls Virtual Test Bed				Available
CATIA	Available	Available	Available	Available
ControlBuild	Available	Available	Available	Available
CosiMate		Available		Available
Cybernetica CENIT		Available		Planned
Cybernetica ModelFit		Available		Available
DSHplus	Planned		Planned	
Dymola	Available ▾	Available	Available ▾	Available
EnergyPlus			Planned	Available
FMI Add-in for Excel				Available ▾

History of the standard

Began as “Modelisar Project”



Project MODELISAR

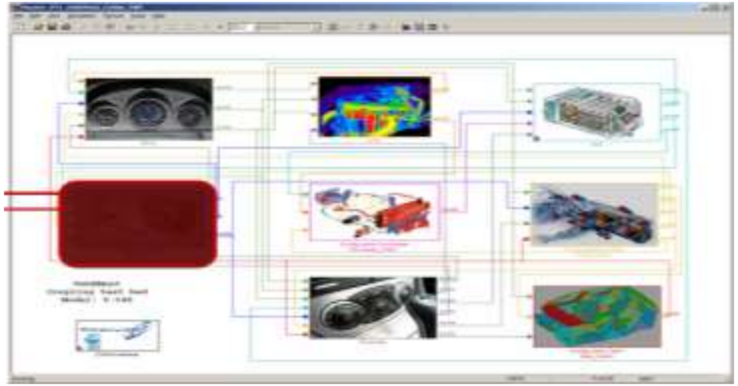
Standardizing simulation of multi-domain physical models with control system models

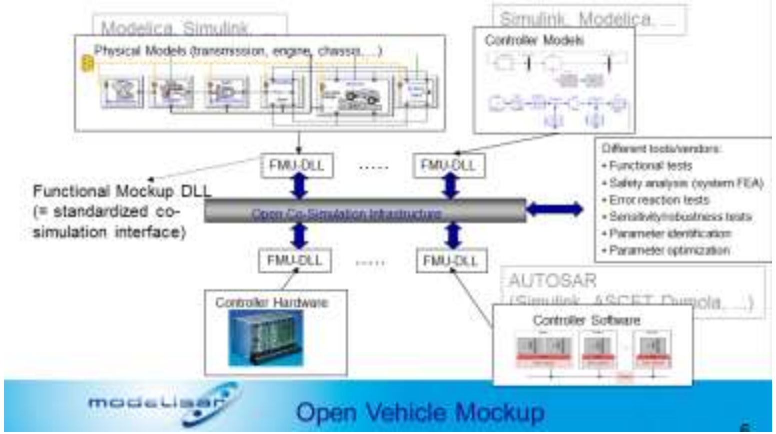
Major project goals

- Easy use of simulation technology for a virtual integration of multi-domain systems
  - Joint simulation of models being developed with different tools
  - Interface for simulation with AUTOSAR control units in the loop
- Develop a proposal for an open standard, the FMI-standard
- Prove the proposed standard by means of several real world use-case scenarios

Use of FMUs

- FMUs are designed for simulations where they are integrated into a system of components, comprised of either
  - Other FMUs or
  - Other simulation codes (e.g. Abaqus)
- FMUs are designed to be “slaves” in a co-simulation, with control over only their internal state.
  - They must be directed by a co-simulation “master”





Modelica, Simulink, ...

Physical Models (transmission, engine, chassis, ...)

Simulink, Modelica, ...

Controller Models

FMU-DLL ..... FMU-DLL

Functional Mockup DLL
(= standardized co-simulation interface)

Open Co-Simulation Infrastructure

FMU-DLL ..... FMU-DLL

Different tools/vendors:
- Functional tests
- Safety analysis (system FEA)
- Error reaction tests
- Sensitivity/robustness tests
- Parameter identification
- Parameter optimization

AUTOSAR
(Simulink, ASCET, Dymola, ...)

Controller Hardware

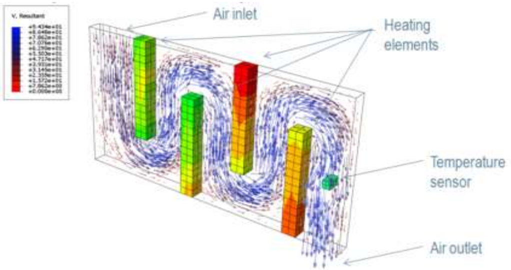
Controller Software

modelisar

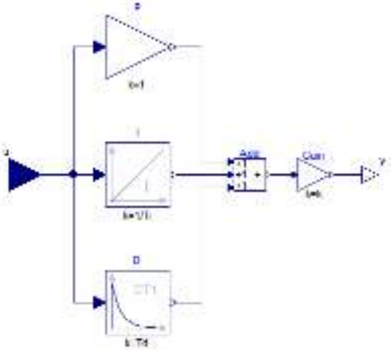
Open Vehicle Mockup

Example

- This example simulation considers a simple electrically controlled heat exchanger illustrated below

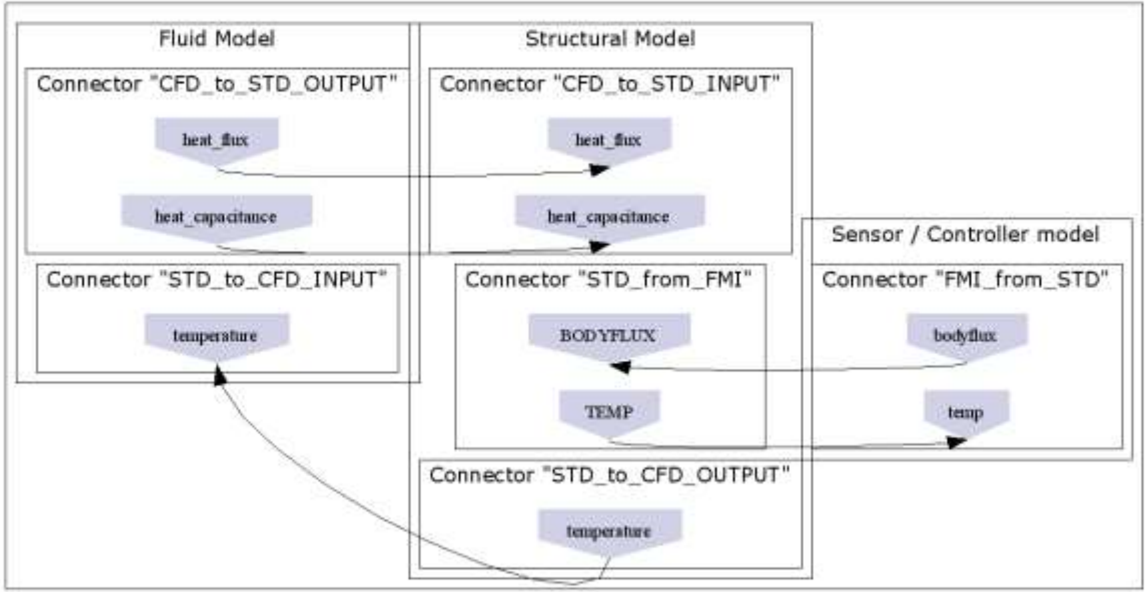


- The simulation comprises an Abaqus/Standard model of the solid heating element, an Abaqus/CFD model of the airflow through the passages, and an FMU representation of a controller. The FMU implementation models a traditional Proportional Integral Derivative (PID) controller that seeks to obtain an air outlet temperature of 50degC quickly without overshoot.



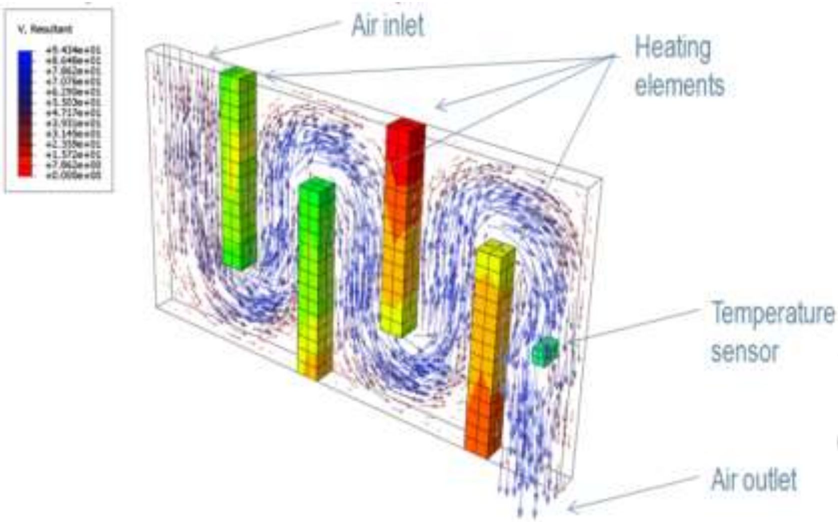
Example

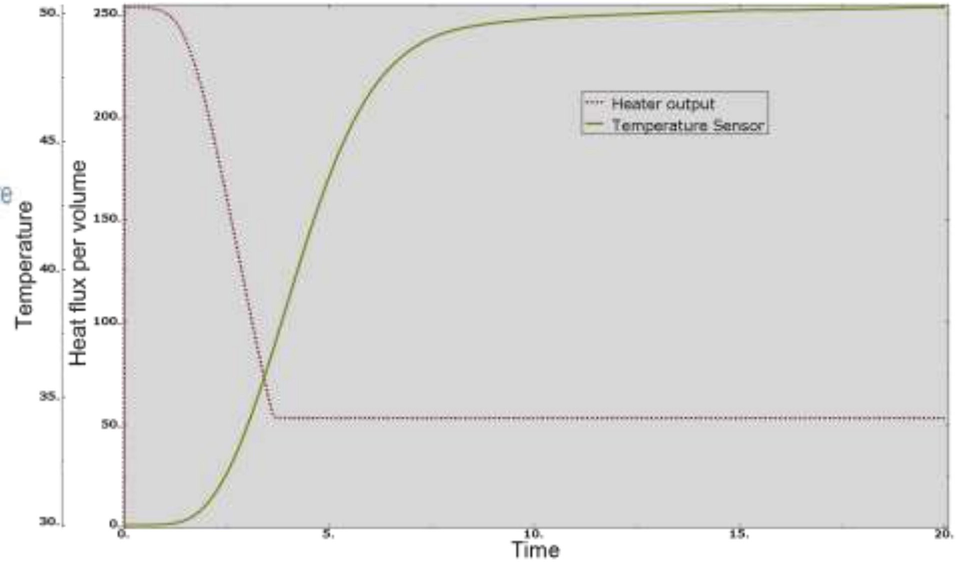
- The configuration file defines the following relationships



Example

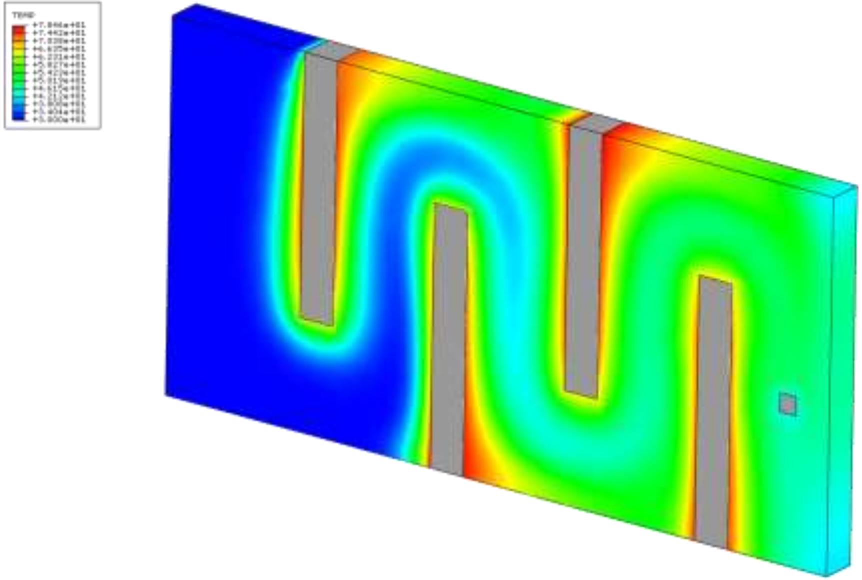
Results

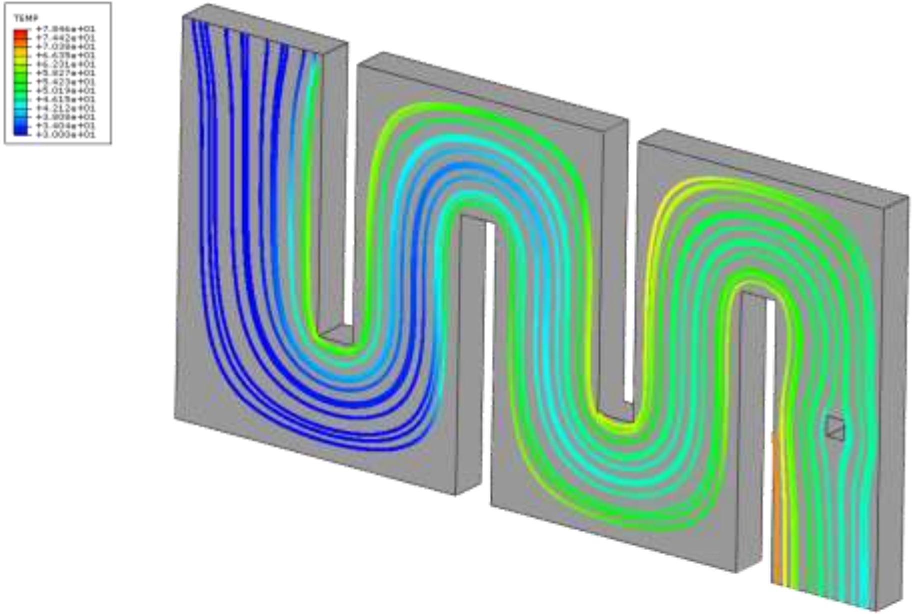




Example

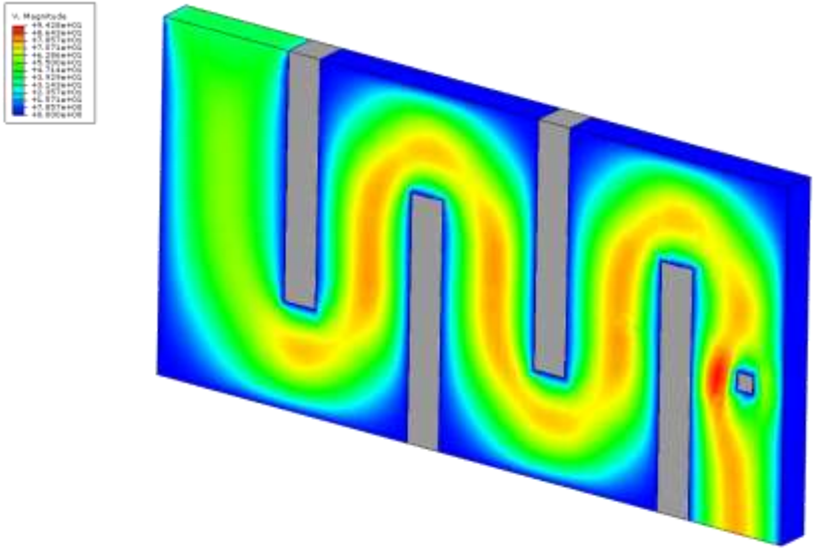
Results: Temperature

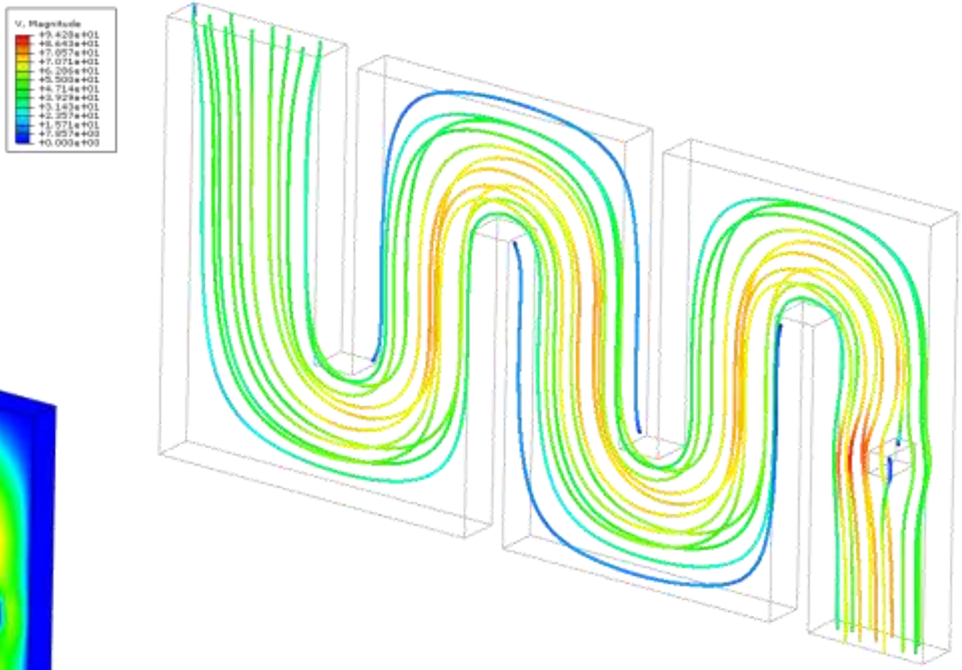




Example

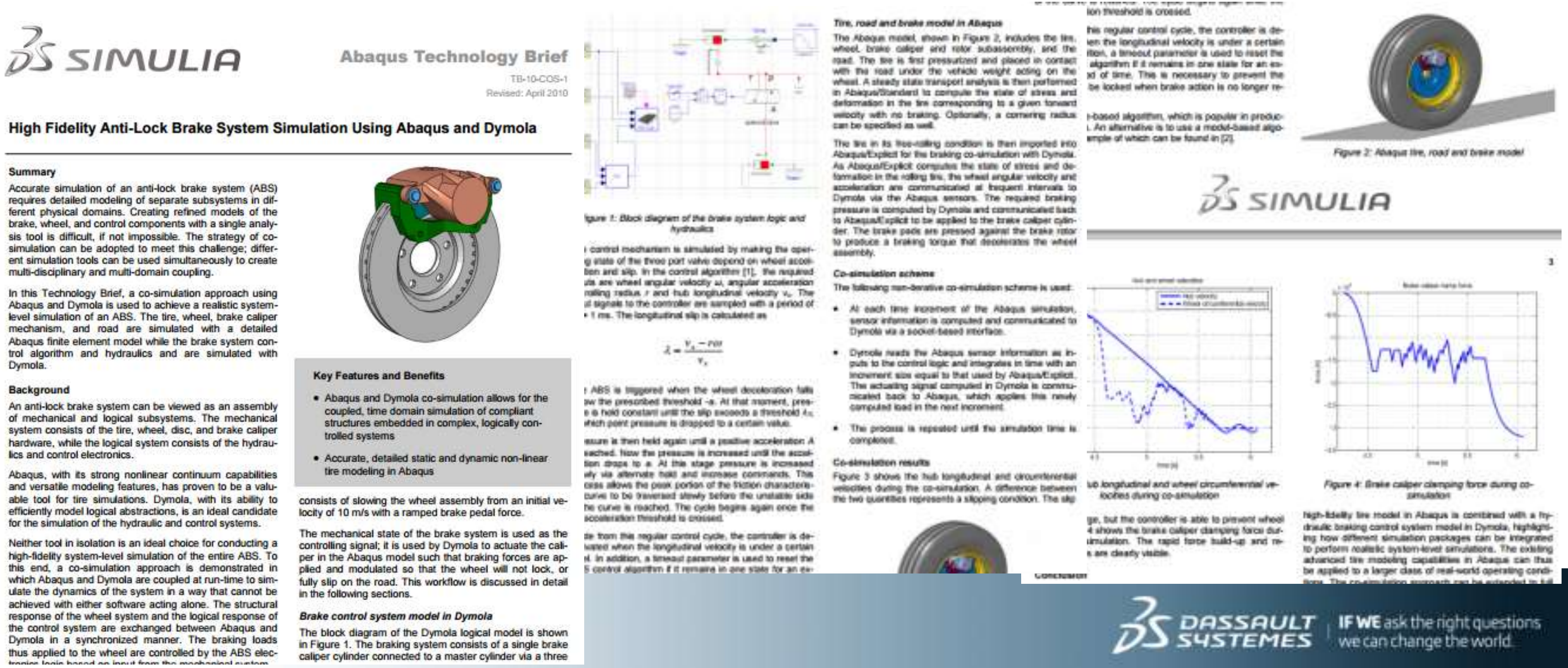
Results: Velocity





Example

http://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/tech-briefs/Auto-High-Fidelity-Anti-Lock-Brake-System-Simulation-10.pdf



SIMULIA

Abaqus Technology Brief

TB-10-COS-1
Revised: April 2010

High Fidelity Anti-Lock Brake System Simulation Using Abaqus and Dymola

Summary

Accurate simulation of an anti-lock brake system (ABS) requires detailed modeling of separate subsystems in different physical domains. Creating refined models of the brake, wheel, and control components with a single analysis tool is difficult, if not impossible. The strategy of co-simulation can be adopted to meet this challenge; different simulation tools can be used simultaneously to create multi-disciplinary and multi-domain coupling.

In this Technology Brief, a co-simulation approach using Abaqus and Dymola is used to achieve a realistic system-level simulation of an ABS. The tire, wheel, brake caliper mechanism, and road are simulated with a detailed Abaqus finite element model while the brake system control algorithm and hydraulics and are simulated with Dymola.

Background

An anti-lock brake system can be viewed as an assembly of mechanical and logical subsystems. The mechanical system consists of the tire, wheel, disc, and brake caliper hardware, while the logical system consists of the hydraulics and control electronics.

Abaqus, with its strong nonlinear continuum capabilities and versatile modeling features, has proven to be a valuable tool for tire simulations. Dymola, with its ability to efficiently model logical abstractions, is an ideal candidate for the simulation of the hydraulic and control systems.

Neither tool in isolation is an ideal choice for conducting a high-fidelity system-level simulation of the entire ABS. To this end, a co-simulation approach is demonstrated in which Abaqus and Dymola are coupled at run-time to simulate the dynamics of the system in a way that cannot be achieved with either software acting alone. The structural response of the wheel system and the logical response of the control system are exchanged between Abaqus and Dymola in a synchronized manner. The braking loads thus applied to the wheel are controlled by the ABS elec-

Key Features and Benefits

- Abaqus and Dymola co-simulation allows for the coupled, time domain simulation of compliant structures embedded in complex, logically controlled systems
- Accurate, detailed static and dynamic non-linear tire modeling in Abaqus

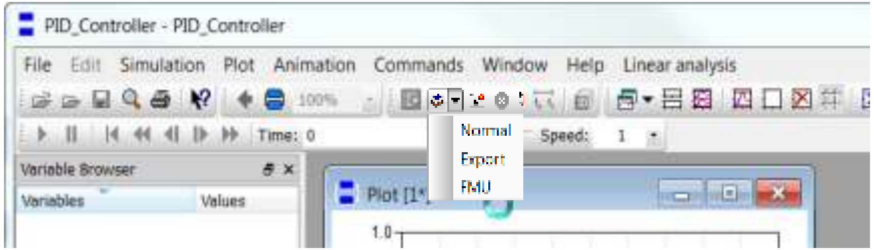
consists of slowing the wheel assembly from an initial velocity of 10 m/s with a ramped brake pedal force.

The mechanical state of the brake system is used as the controlling signal; it is used by Dymola to actuate the caliper in the Abaqus model such that braking forces are applied and modulated so that the wheel will not lock, or fully slip on the road. This workflow is discussed in detail in the following sections.

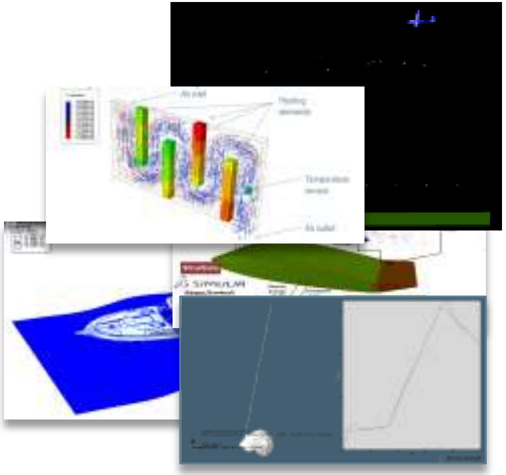
Brake control system model in Dymola

The block diagram of the Dymola logical model is shown in Figure 1. The braking system consists of a single brake caliper cylinder connected to a master cylinder via a three

Using FMUs in your work



- Create them with Dymola
  - Note that the FMU interface will replace the current Dymola DLL interface
  - All Dymola DLL QC tests have equivalent FMU based tests
- Write your own
  - An SDK is provided by http://www.qtronic.de/en/fmusdk.html
    - This kit is used by Jeff, Jaesu, David Fox for creating FMUs from c-code



Our Values

IF WE SHOW THE **DREAM** IS POSSIBLE

› WE CAN **INSPIRE** PEOPLE ‹
TO CREATE IT

IF WE **CHALLENGE** THE STATUS QUO

› WE CAN **IMAGINE** NEW HORIZONS ‹
TO IMPROVE THE WORLD

IF WE HAVE THE **PASSION** TO LEARN

› WE CAN **EXPAND** CREATIVITY ‹
TO NAVIGATE THE FUTURE

IF WE BRING OUR **COMMUNITY** TOGETHER

› WE CAN **BUILD** HARMONY ‹
TO ACHIEVE OUR GOALS

IF WE™