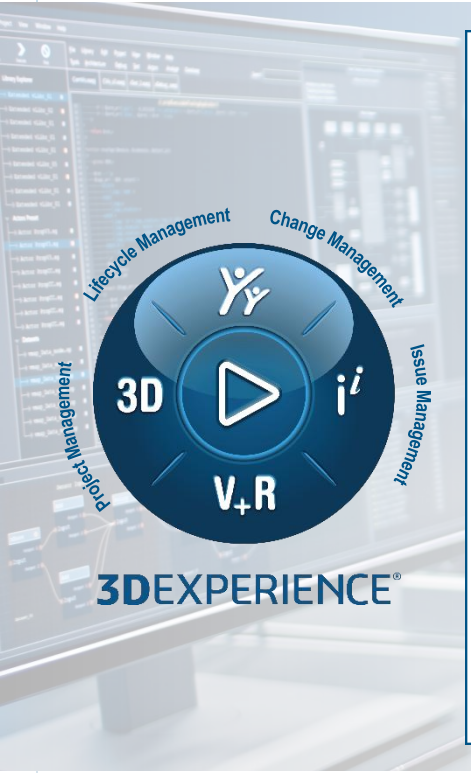
A BRIEF INTRODUCTION TO MODELICA, DYMOLA

JYOTHI MATAM
WW Autonomous driving / ADAS system simulation Industry Process Senior Manager

3DS SYSTEMS ENGINEERING PORTFOLIO



System Management

- Requirements Management
- System Traceability
- Report Generation

System Specification

- System of Systems Specification
- System Specification

System Design

Architecture & Behavior Design

- System Architecture Design
- Multi-Physics Behavior Design
- Control Behavior Design

Software Design

- Software Architecture Design
- Software Code Generation

Hardware Design

- Embedded Architecture Design
- Electronics Communication Design

Domain-Specific System Design

- Mechanical
- Fluidics
- Electrical
- Electronics
- Software

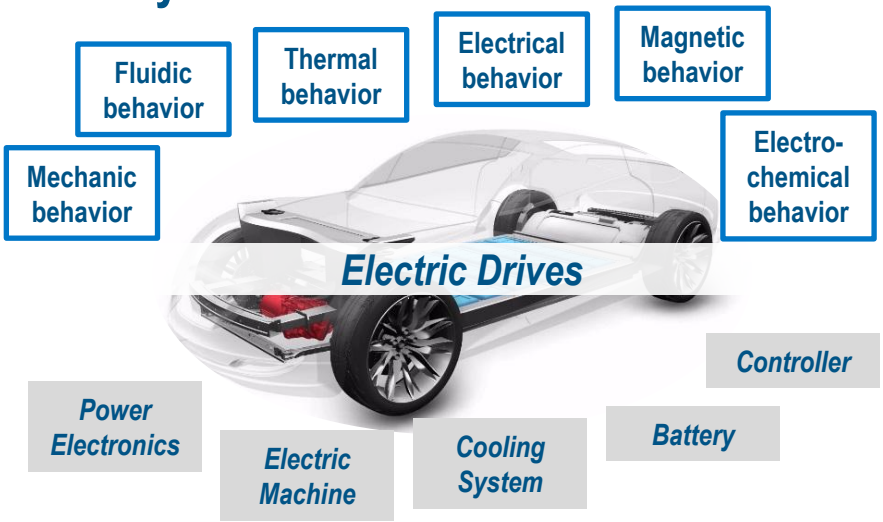
System Validation

- Requirements Validation
- System Safety
- Test Design
- Test Experience

SYSTEMS SIMULATION

Systems

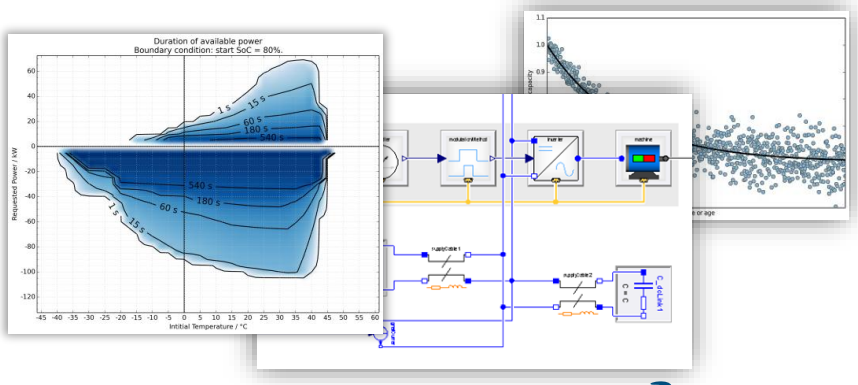
Physical Domains



Components

Simulation

- Optimization of system architecture
- Selection of appropriate technologies
- Design of control algorithms
- Minimization of energy consumption
- Understanding system-internal interactions



DASSAULT SYSTEMES

CATIA SYSTEMS SIMULATION

Framework

SERVICES

Solution & content related **services to support a smooth & fast introduction** and sustainable usage

DYMOLA BEHAVIOR MODELLING *

3DEXPERIENCE **applications connected to** power of **platform** solutions

MODELICA

Open modeling language for multi-physic simulation-models

MODELICA SIMULATION APPS *

Easily explore and experiment multi-physics behavior models **for trade-off analysis**

FUNCTIONAL MOCK-UP INTERFACE

Tool independent standard to support both model **exchange** and **co-simulation** of **dynamic models**

DYMOLA STANDALONE *

Stand-alone **solution to model and simulate** dynamic **systems**

LIBRARIES

Domain-specific libraries (mechanical, electrical, fluid, thermal, hydraulic, etc.)

*** shares the same kernel**

MODELICA & DYMOLA

A brief overview

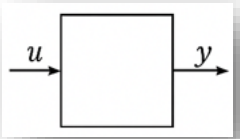
0D VS 1D SYSTEM MODELS

0D (Lumped Parameter Models)

- Variables depend only on time (t)
- No spatial resolution (uniform values)
- Suitable for: concept design, controls, quick simulations

Examples in Dymola:

  – Battery modeled as an equivalent circuit
  – Room temperature (single node thermal model)
  – Rigid body dynamics

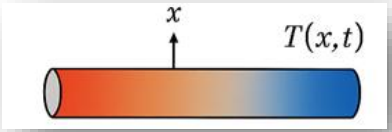


1D (Distributed Parameter Models)

- Variables depend on space + time ⟶ PDEs discretized into ODEs
- Captures gradients (temperature, pressure, flow) along one dimension
- Higher fidelity; useful for detailed system performance

Examples in Dymola:

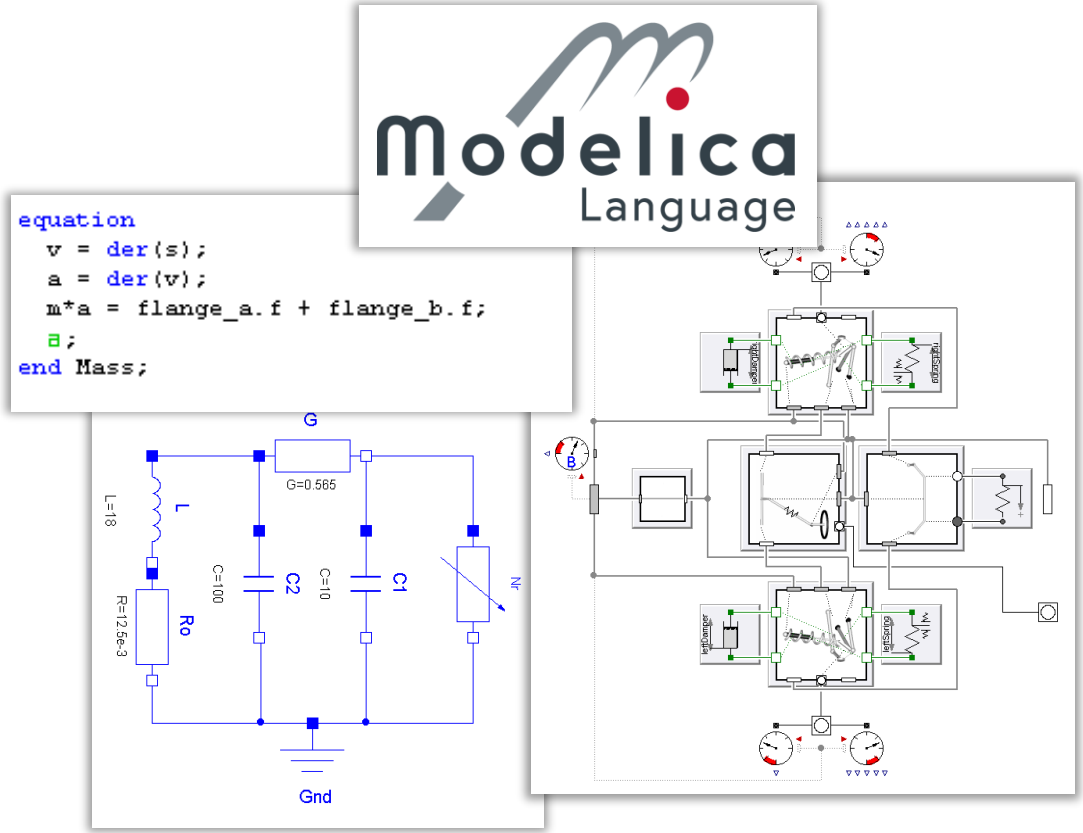
  – Heat conduction along a pipe or rod
  – Fluid dynamics in pipelines
  – 1D driveline torsional vibration



MODELICA

A Language to Model Physical Behavior

- Equation-based A-causal modelling language
- Declarative
- Object-oriented
- Inherently multiphysical
- Non-proprietary
- Maintained by the Modelica Association

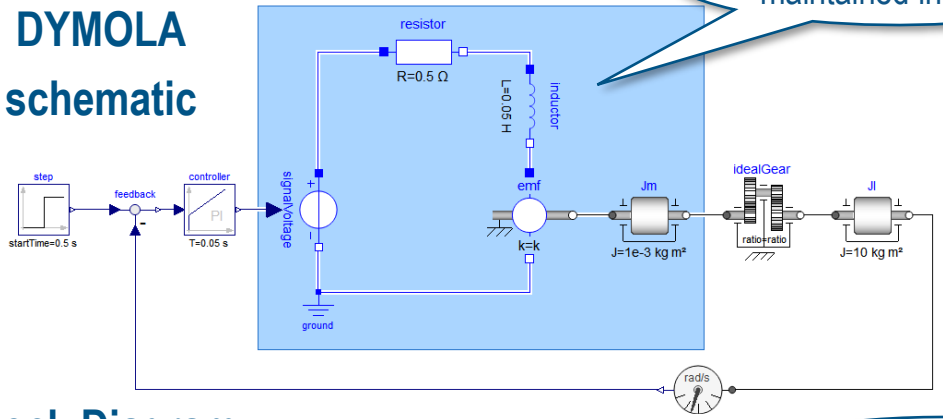


DASSAULT SYSTEMES

ACAUSAL MODELING VS. BLOCK DIAGRAMS

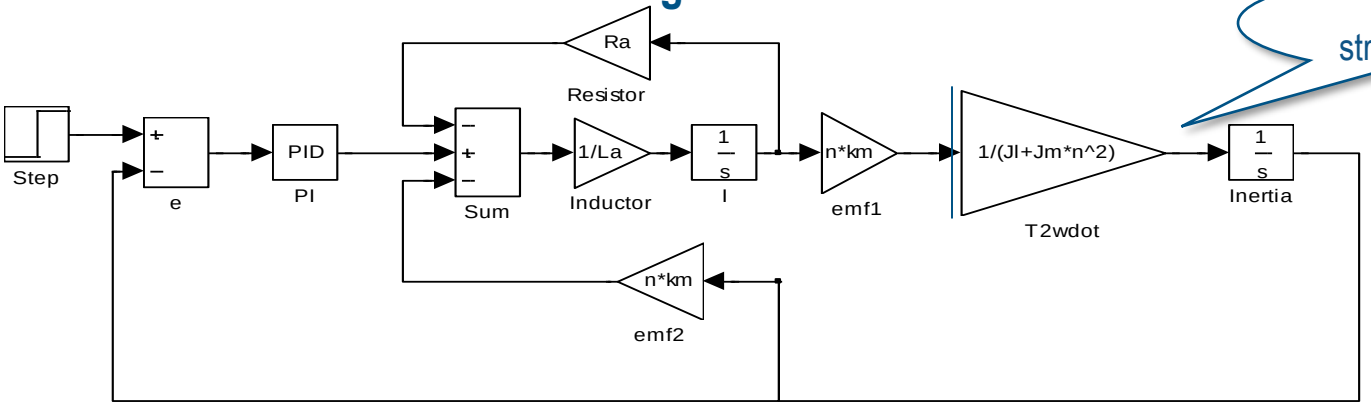
Example:
Simplified electric drive

DYMOLA schematic



Physical structure maintained in model

Block Diagram



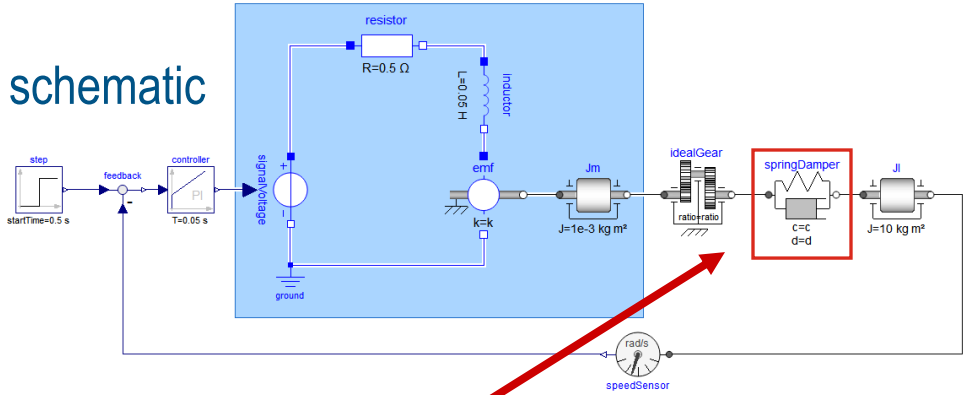
No physical structure in model

ACAUSAL MODELING VS. BLOCK DIAGRAMS

Engineering problem
Study vibration effects of the shaft

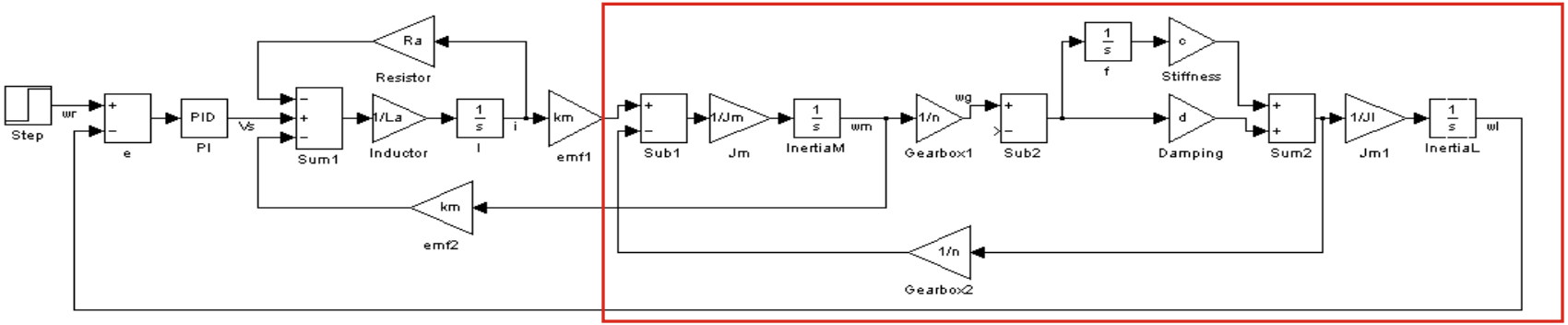
Engineering solution
Add *flexible* shaft to model

DYMOLA schematic



Block Diagram

Change needed



DYMOLA

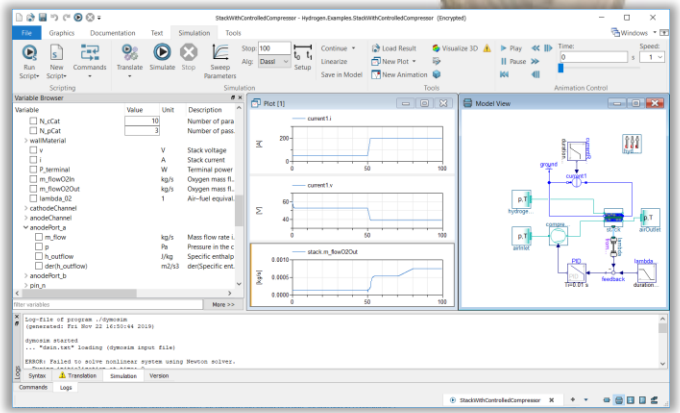
Dymola is a Modelica compliant solution to model & simulate dynamic systems efficiently

- Powerful Multi-Disciplinary System Simulation
- High-fidelity Modeling of Complex Systems
- Open model libraries enables fast result generation
- FMI support for Windows and Linux

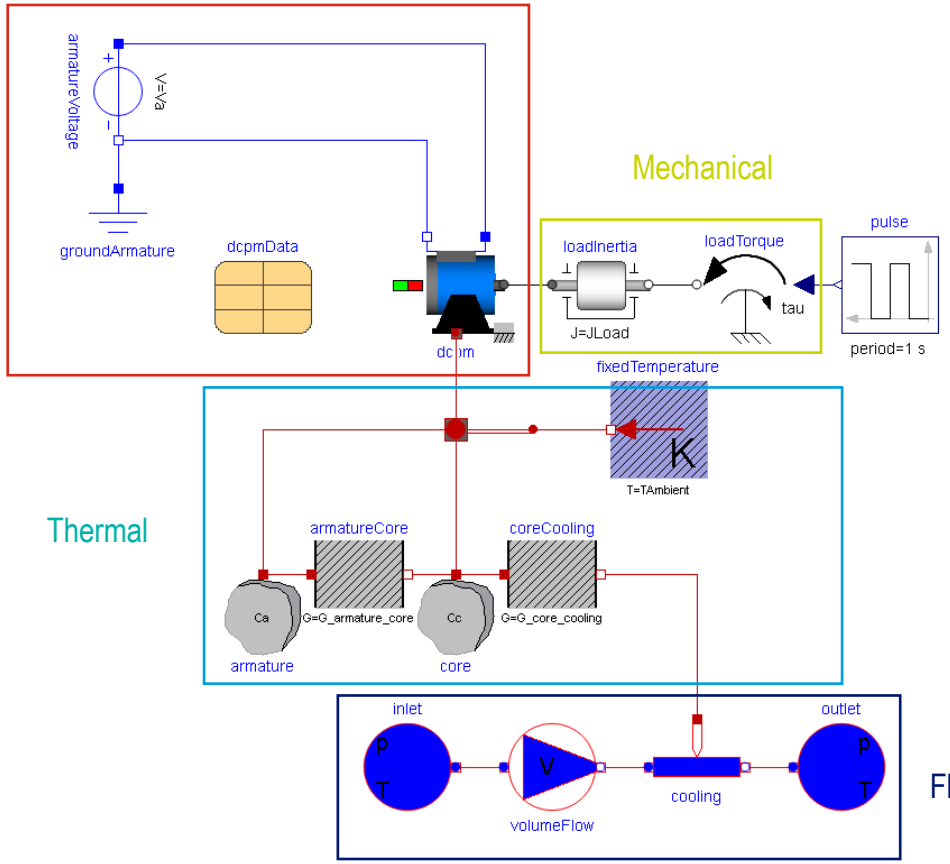
DYMOLA

Key Advantages

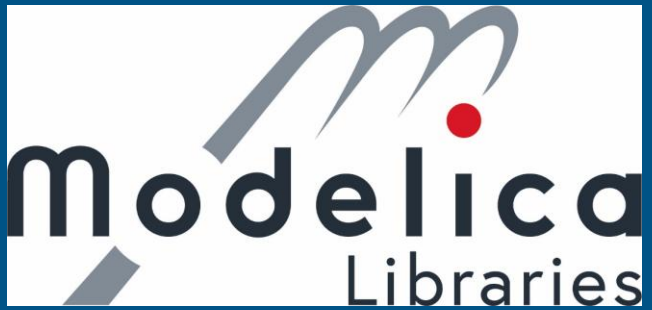
- **Multi-engineering** Compatible model libraries for many engineering fields enable high-fidelity modeling of complex integrated systems.
- **Modelica** A powerful, object-oriented and formally defined modeling language.
- **Free and commercial libraries** User can easily build own or adapt existing components to match unique needs. Comprehensive portfolio of model libraries.
- **Reuse** Acausal, equation-oriented models allow a component to be used in different contexts and a model to be used for different studies.
- **Outstanding Performance**
  Unique and outstanding performance for solving systems of differential algebraic equations due to **symbolic equation processing** which optimizes and solves the systems of equations.
- **Open and Flexible**
  The Modelica based environment is completely open in contrast to many modeling tools that have a fixed set of component models and proprietary methods for introducing new components.



MULTI-DISCIPLINARY SYSTEMS MODELLING

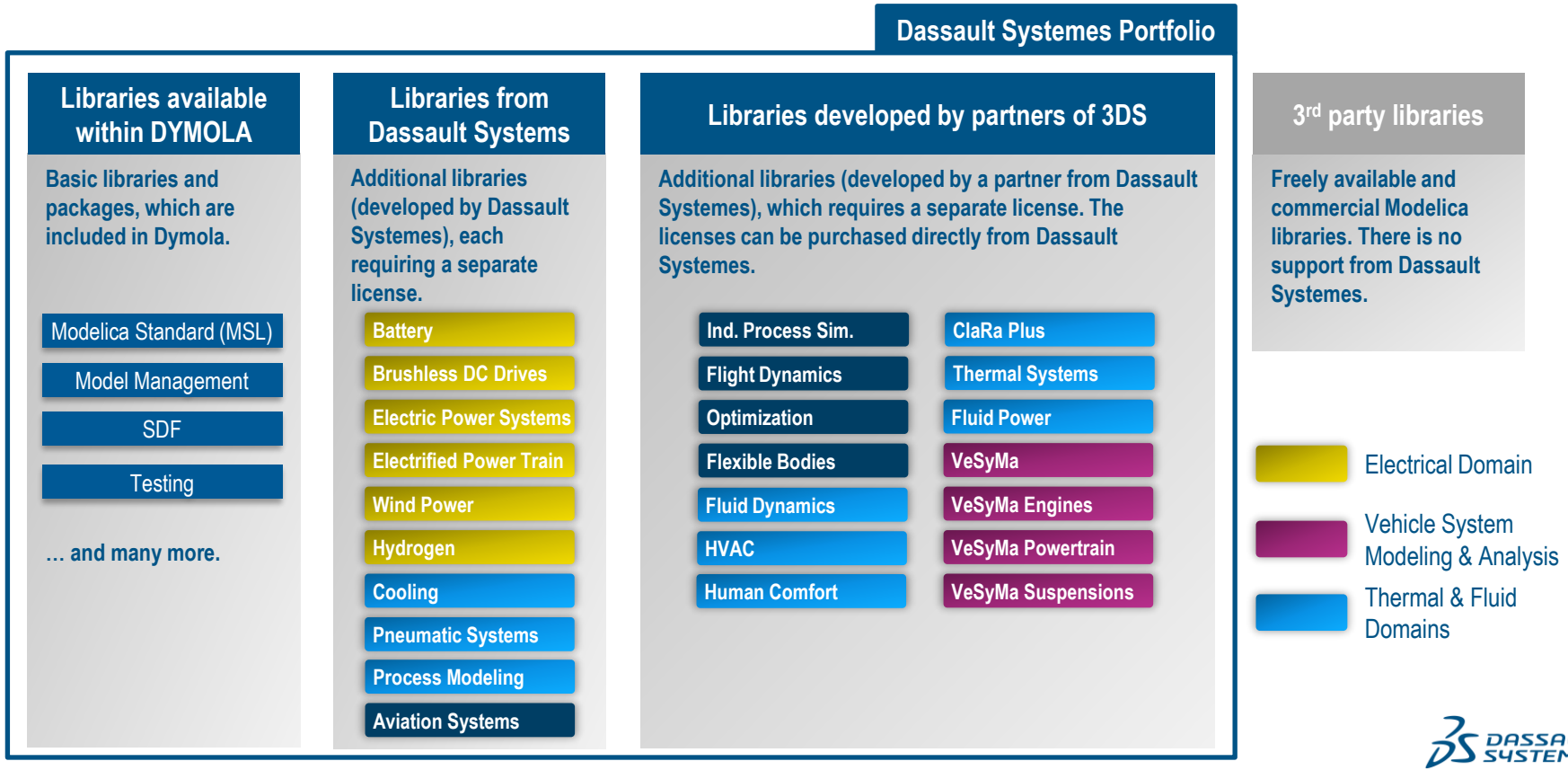


MODELICA LIBRARIES



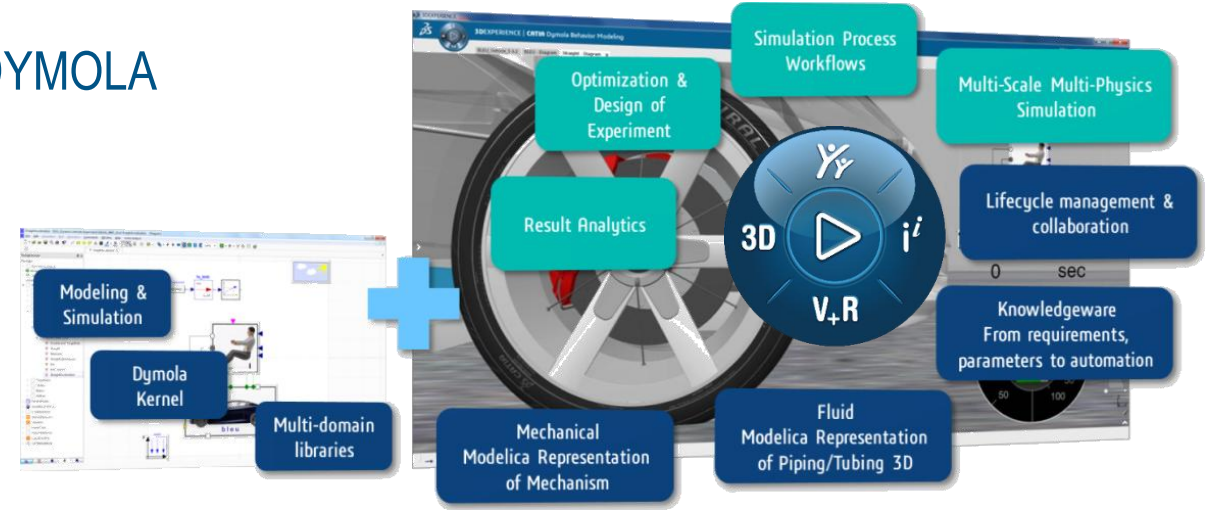
SYSTEMS SIMULATION PORTFOLIO

Libraries



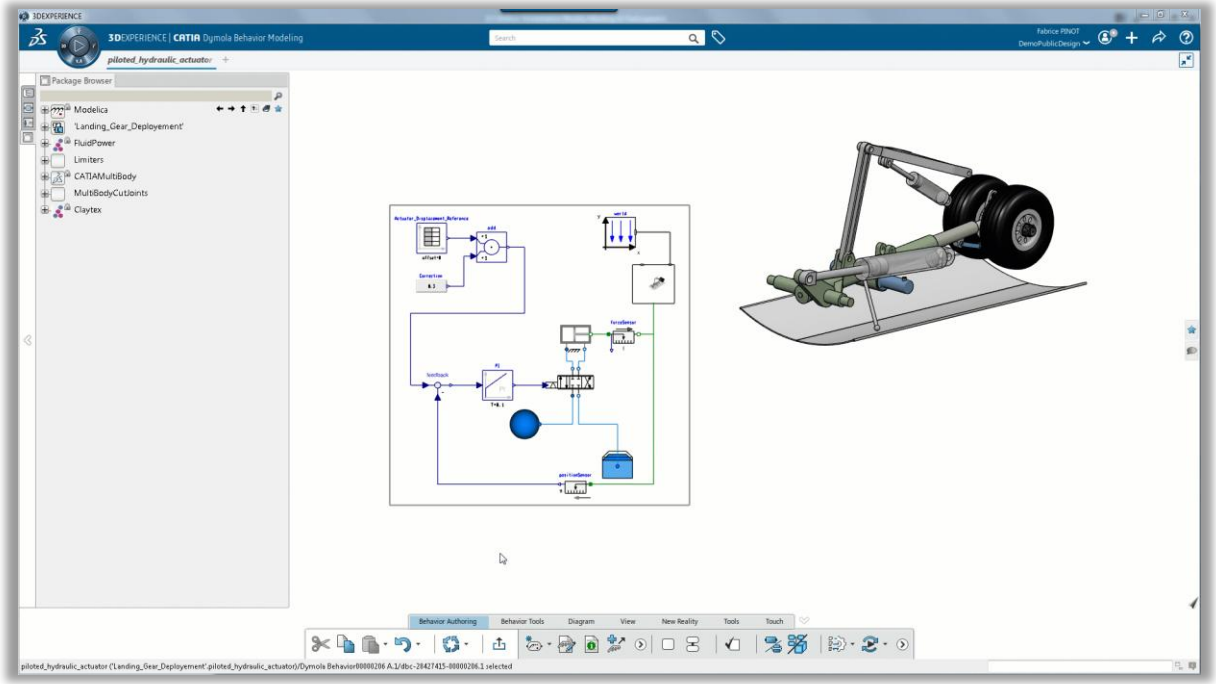
DYMOLA AS A 3DEXPERIENCE APPLICATION

- Shares the same kernel as DYMOLA
- Models and libraries are fully compatible
  - Same libraries and extensions available
- Create a virtual twin of the system
  - Efficient & exact 3D modeling combined with Modelica multi-physics behaviors.
- Quickly converge to the optimal solutions in a single platform
  - Automation, multi-scale simulation, process workflow and result analytics



MODELICA CREATION BASED ON KINEMATICS

- Create and Play 3D Mechanisms
- 3 clicks to generate Modelica Model
- Create Multi-Physics Simulations
- Modify 3D and Update Models
- Add Flexible Bodies



DASSAULT SYSTEMES

Systems Simulation Design

.dsone.3ds.com

3DEXPERIENCE | Systems Simulation Design

Search In Current Dashboard

BasicTemplate

Simulation up-to-date
BasicTemplate

CockpitView

Type: Default

Compressor

Cylinder Capacity	125	cm3

Exchanger

Condenser Length	0.210 - 2*0.00(	m
Evaporator Length	0.210 - 2*0.00(	m
AirCooler Height	340	mm

Battery

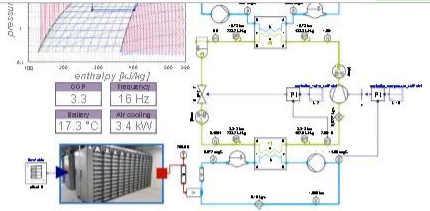
Number of Cells (serial)	90	
Cooling Loop (Temp. set point)	10	Cdeg

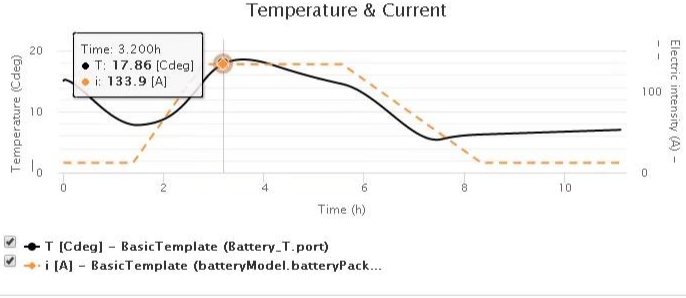
Add a category

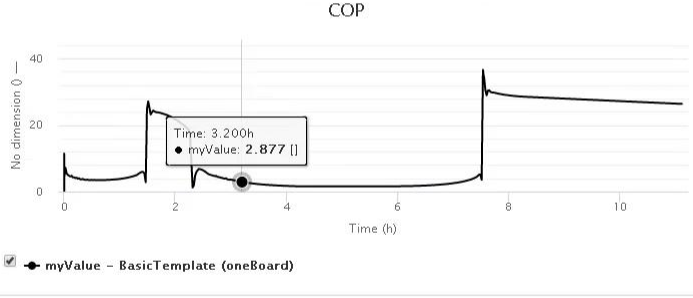
Add Free text

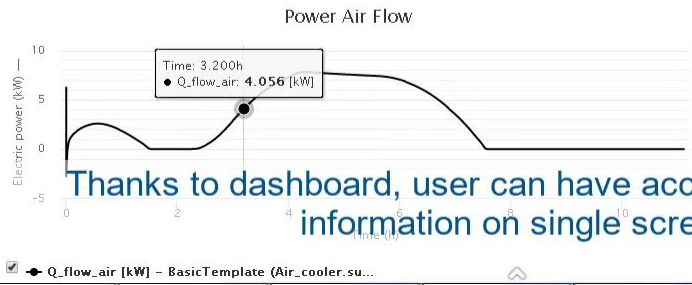
Add diagram

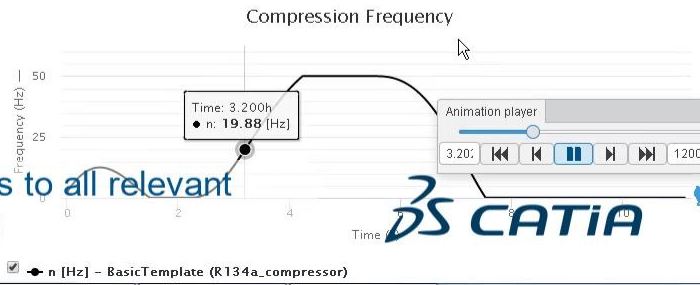
Air Flow Power	4.056	kW
Battery Temperature	17.86	Cdeg
Battery output current	133.9	A











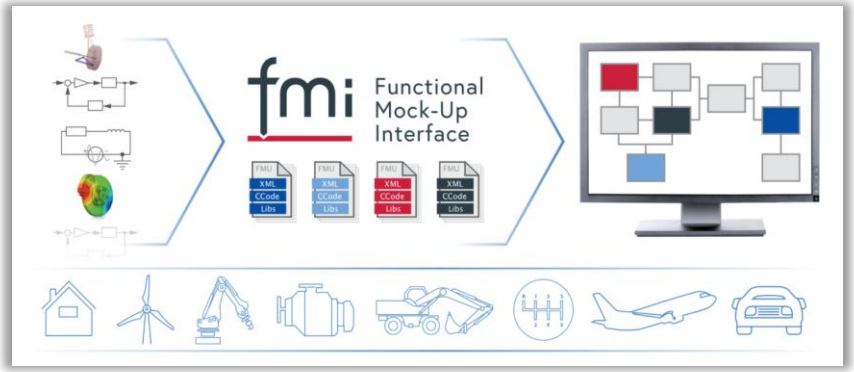
Thanks to dashboard, user can have access to all relevant information on single screen

DS CATIA

FUNCTIONAL MOCK-UP INTERFACE

FUNCTIONAL MOCK-UP INTERFACE

Functional Mock-up Interface (FMI) is an **open** and **tool independent standard** to support both **model exchange** and **co-simulation of dynamic models** using a combination of xml-files and compiled C-code.



The development of the standard is being progressed with the participation of 16 companies and research institutions under the roof of the Modelica Association (https://fmi-standard.org/) as the Modelica Association Project.

FUNCTIONAL MOCK-UP INTERFACE

Overview

- Enables **dynamic models of different software** systems to be **used together** for systems development and software-, model- and hardware-in-the-loop simulations.
- The first version FMI 1.0, was published in 2010 within the frame of the **ITEA2 E.U. MODELISAR Project.** Second version FMI 2.0 was released in 2014**.** New FMI version 3.0 is under development.
- FMI is currently **supported by over 100 tools** and is used by automotive and non-automotive organizations throughout Europe, Asia and North America.
- The FMI standard enables to **develop component models with different tools** and supports the exchange of customizable models between working groups and companies.
- FMI is the de-facto **industry standard** for the **exchange of simulation models**.

FMI

Motivation & Use Case

MOTIVATION

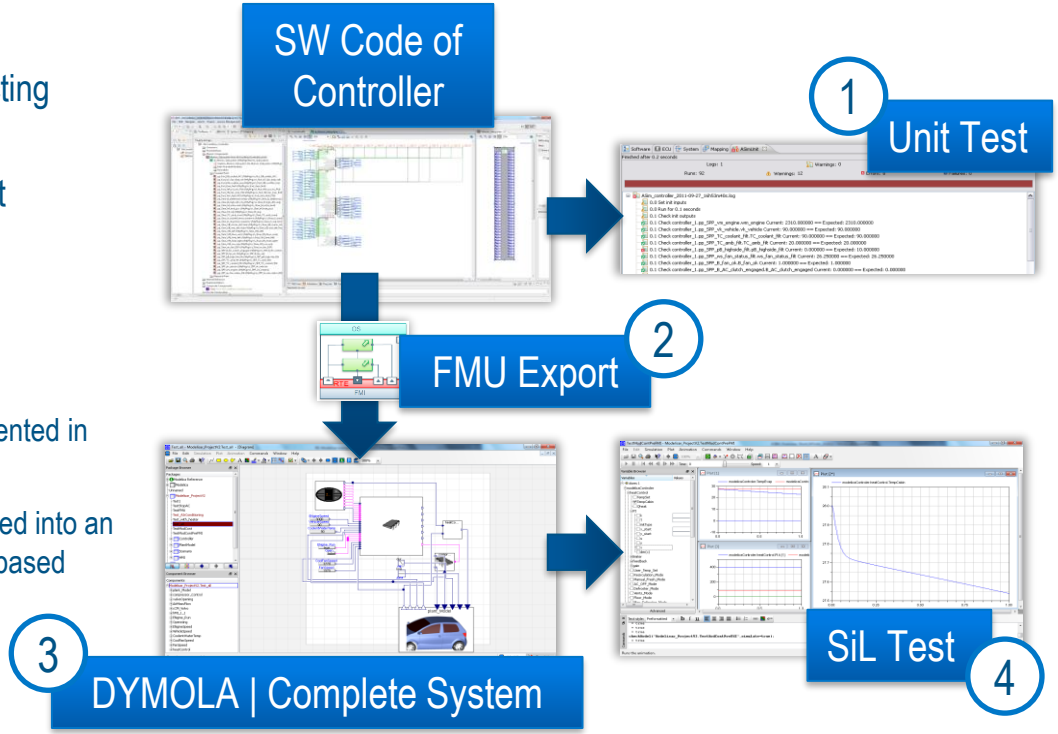
FMI standard allows virtual system integration by connecting simulation models from different sources.

- Model exchange between Tier1 and OEM and different departments with different roles.
- DASSAULT SYSTEMS fully supports FMI standard.

USE CASE

Air conditioning system developed in DYMOLA. Controller implemented in AUTOSAR BUILDER.

- Controller software component exported as a FMU and integrated into an air-conditioning plant model, which is modeled in the Modelica-based simulation tool DYMOLA.

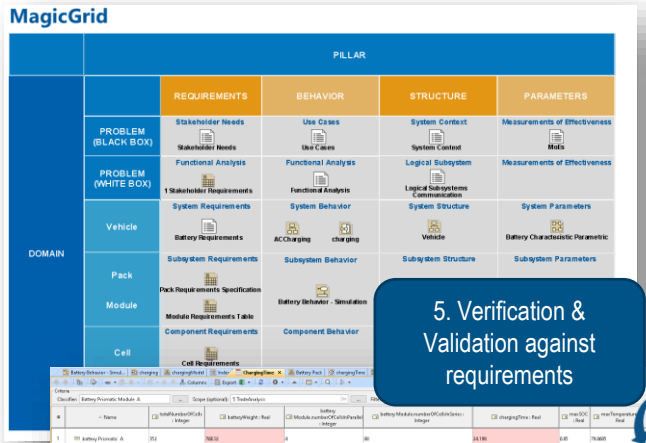


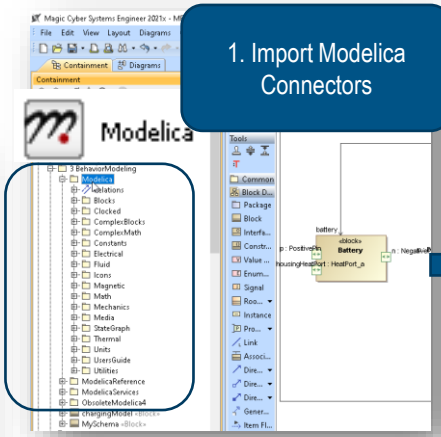
More information about FMI:
https://fmi-standard.org/

fmi Functional Mock-Up Interface

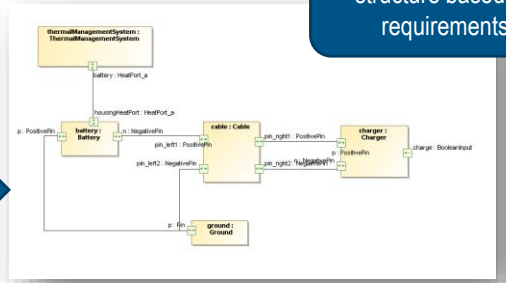
DASSAULT SYSTEMES

SYSML & MODELICA | DYMOLA INTEGRATION

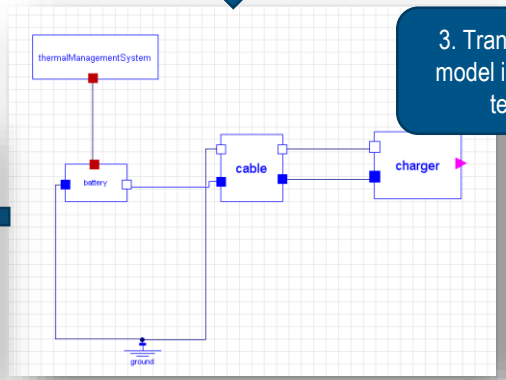




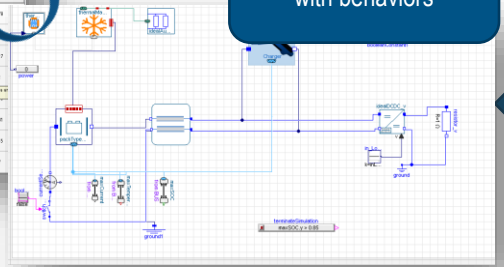
1. Import Modelica Connectors



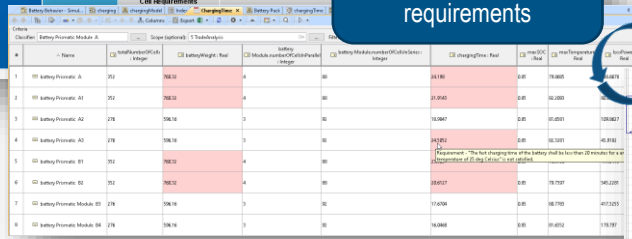
2. System/sub-system structure based on requirements



3. Translate SysML model into Modelica template



4. Extend the model with behaviors



5. Verification & Validation against requirements

DASSAULT SYSTEMES

Virtual Worlds for Real Life

DISCOVER MORE ON 3DS.COM

DASSAULT SYSTEMES