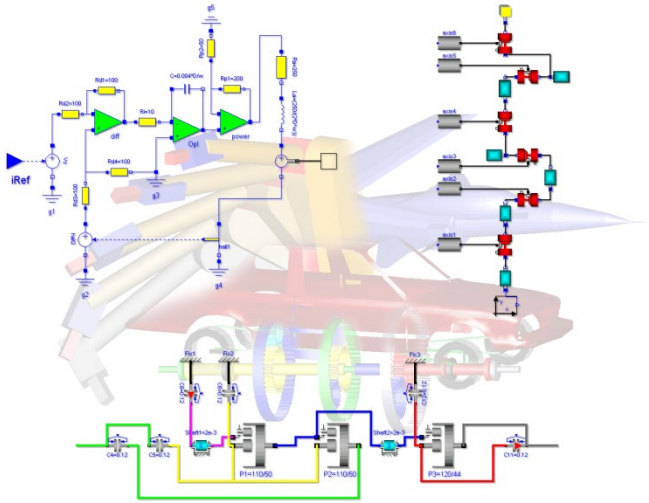
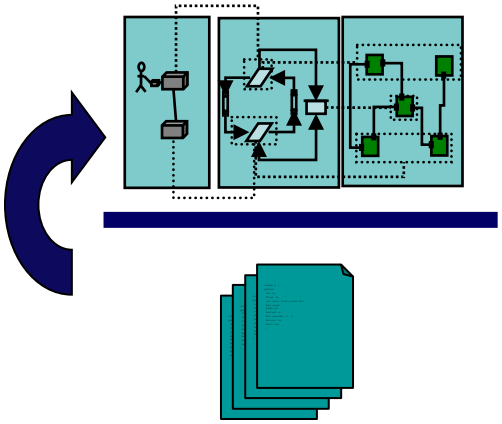
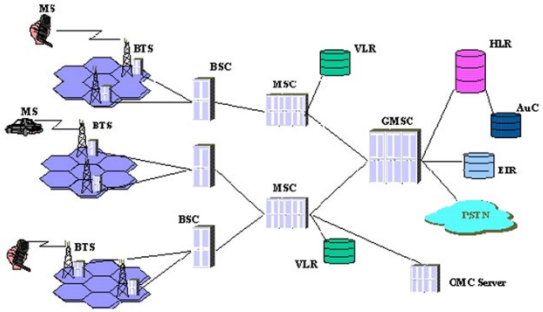
# Research in Model-Based Product Development at PELAB and RISE in the MODPROD Center

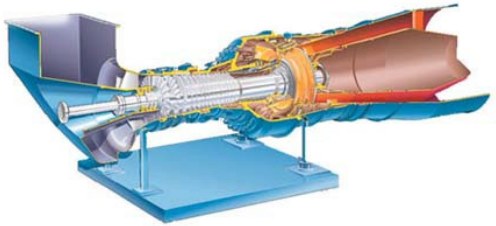
**Presentation at MODPROD'2025**
**Department of Computer and Information Science**
**Linköping University**
**2025-02-04**

Adrian Pop, Peter Fritzson, Martin Sjölund, Lena Buffoni, Lennart Ochel,
John Tinnerholm, Mahder Gebremedhin, Robert Braun, Arunkumar Palanisamy,
Per Östlund, Adeel Asghar, Abdelazim Hussien

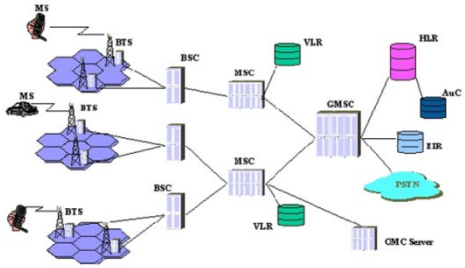




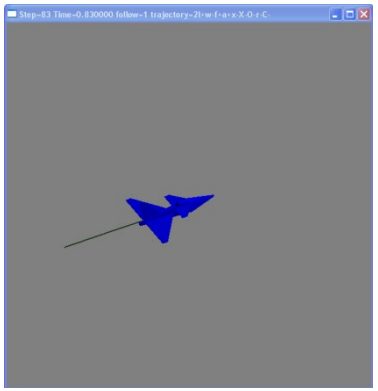
# Industrial Challenges for Complex Products of both Software and Hardware




- Increased Software Fraction
- Embedded and real time constraints
- Higher demands on effective strategic decision making

**Digitalization Revolution Happening Now!**

**Internet of Things, AI, CPS**





MODELICA pelab

# Research

**Large-Scale Modeling and Simulation**

**Modeling-Language Design**

**Model-Based Co-simulation with FMI and TLM**

**Model Debugging**

**Model-Based Fault Analysis**

**Embedded System Real-Time Modeling**

**Modeling in the Cloud**

**Energy Modeling**

**Sustainable Systems**

# Large-Scale, High Performance Model-Based Development

# 10 million equation goal!

**Per Östlund, Adrian Pop, Martin Sjölund, Mahder Gebremedhin, John Tinnerholm, Abdelazim Hussien**

**Peter Fritzson, et al**

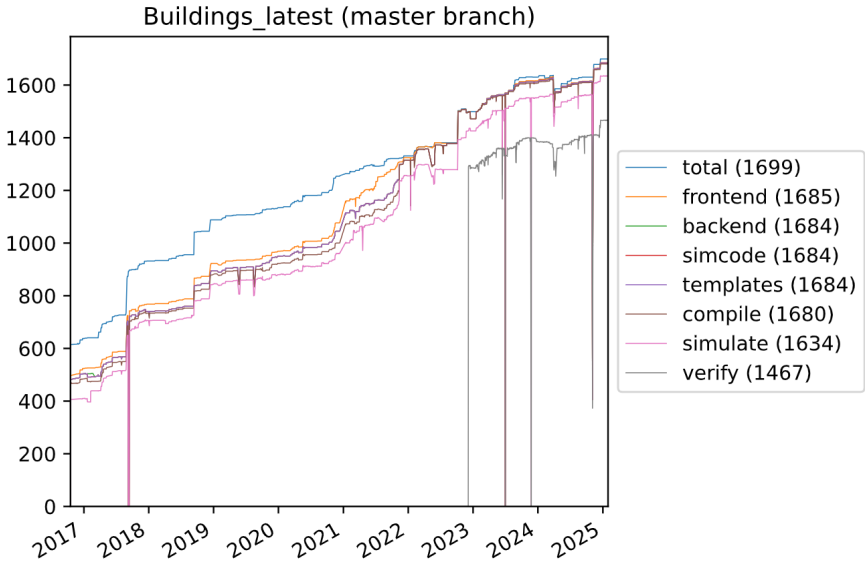
# High Performance Modelica Compilation Methods for Large Model Applications

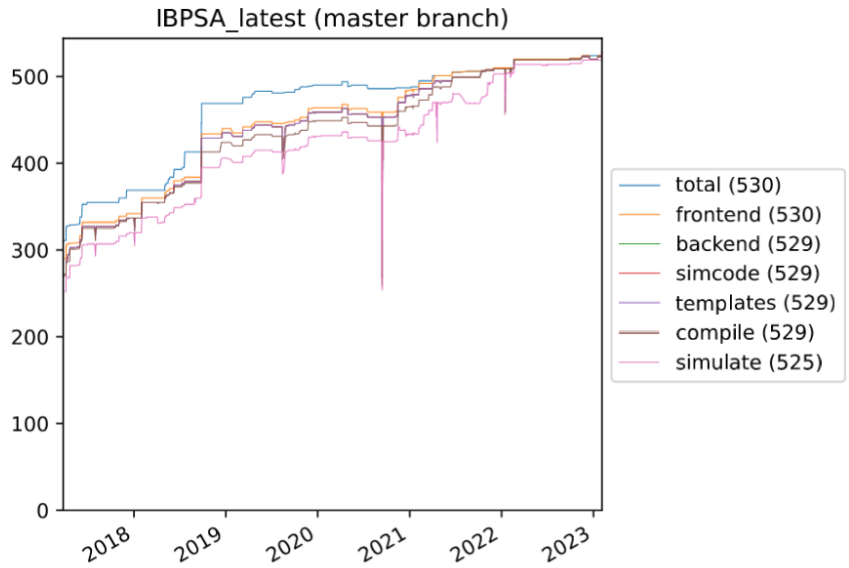
- The **OpenModelica new compiler frontend** – a **large** effort to **redesign** and rewrite more than half of the compiler to gain high compilation **performance** and 100% Modelica semantics
- Uses **Model-centric** and **multiple phases design** principles
- OpenModelica 1.14.1 December 2019 – First release with New Frontend. OpenModelica 1.18.x July 2021 on by default.
- The New frontend is about **10 to 100 times faster** than the old compiler frontend. It can also flatten more libraries: Buildings, Chemical, ClaRa, HanserModelica, HelmholtzMedia, IBPSA, MEV, ModelicaByExample, Modelica (3.2.3 & 4.0.0), ModelicaTest, Modelica_DeviceDrivers, OpenHydraulics, OpenIPSL, PNLib, PhotoVoltaics, PhotoVoltaics_TGM, PlanarMechanics, PowerGrids, PowerSysPro, PowerSystems, ScalableTestGrids, ScalableTestSuite, SystemDynamics, ThermoPower, TILMedia, ThermoFluidStream, VehicleInterfaces

- During 2024 – Further tuning and performance increases; coupling with the new backend in development at FA Bielefeld. New API to use the new frontend in model editing as well.

# Continuing collaboration with LBL on Buildings

- Strategic partnership started in 2021 with LBL (US gov't laboratory in Berkeley)
- Goal: provide open-source support for Modelica libraries (Buildings, IBPSA) involved in the Spawn of Energy Plus project



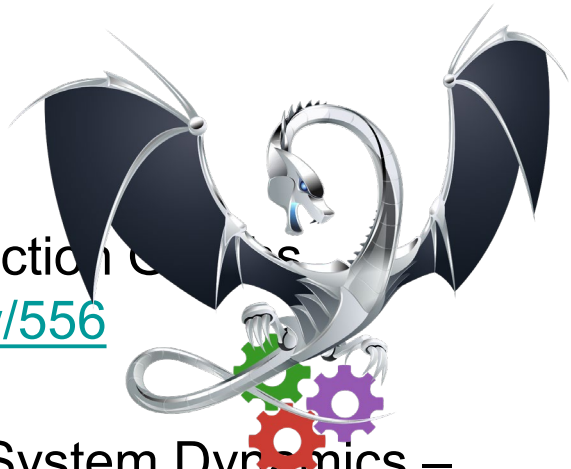




- **98% Build success on Buildings and IBPSA**
- **98% Simulation success on Buildings, 98% on IBPSA**
- **Started at version 7 now version 10**

# Experimental OpenModelica Compiler in Julia
## Goal – Flexible Just-in-time Compilation, variable structure

- During 2019 , Developed a preliminary MetaModelica to Julia translator
- Translated most of the previous OM frontend
- Able to execute some translated MetaModelica functions
- Goal – support variable structure systems (VSS) and large-scale models

- Status 2022 – new front-end translated,
  PoC for VSS using recompilation during runtime
- Status 2023 first release:
  https://github.com/JKRT/OM.jl
  new applications: Dynamic Overconstrained Connection Graphs
  https://ecp.ep.liu.se/index.php/modelica/article/view/556
- Status 2025 first release:
  new applications: ESCIMO/Earth3 translated from System Dynamics – VenSim
- John Tinnerholm - PhD thesis defence in autum

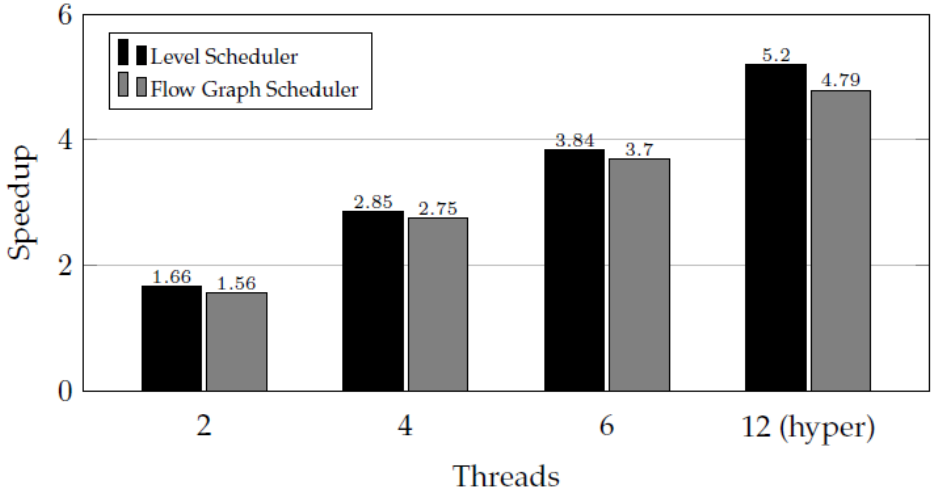


# ParModAuto Parallelization (Released spring 2020)

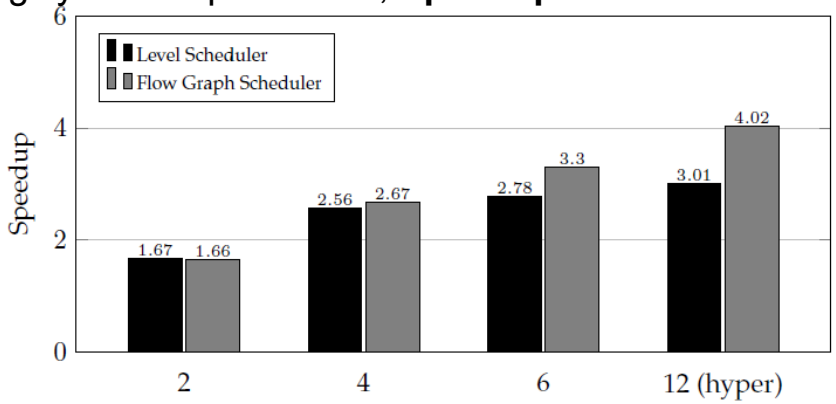
## Automatic AutoTuned Parallelization of Equation-based Models

- **Parallelization for higher performance**
- Automatic **Parallelization**
- Automatic **clustering** of small tasks
- **Automatic load balancing** based on measurements, automatically adapts to changing load
- **Shared-memory** task parallelization

SteamPipe640 model, **Speedup 5.2 on 6 cores**:




BranchingDynamicPipes model, **Speedup 4 on 6 cores**:




- 2022 – better integration in OM
- 2023 – testing different models
- 2025 – trying new metaheuristic optimizations for better speedup



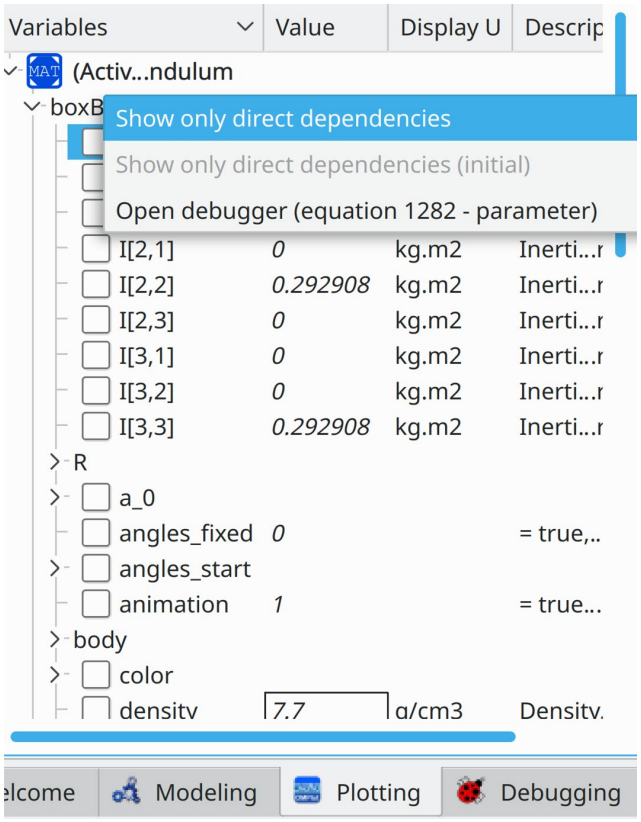
MODELICA pelab

# Enhance Modeling Ease-of-use! Model Debugging and Performance Analysis

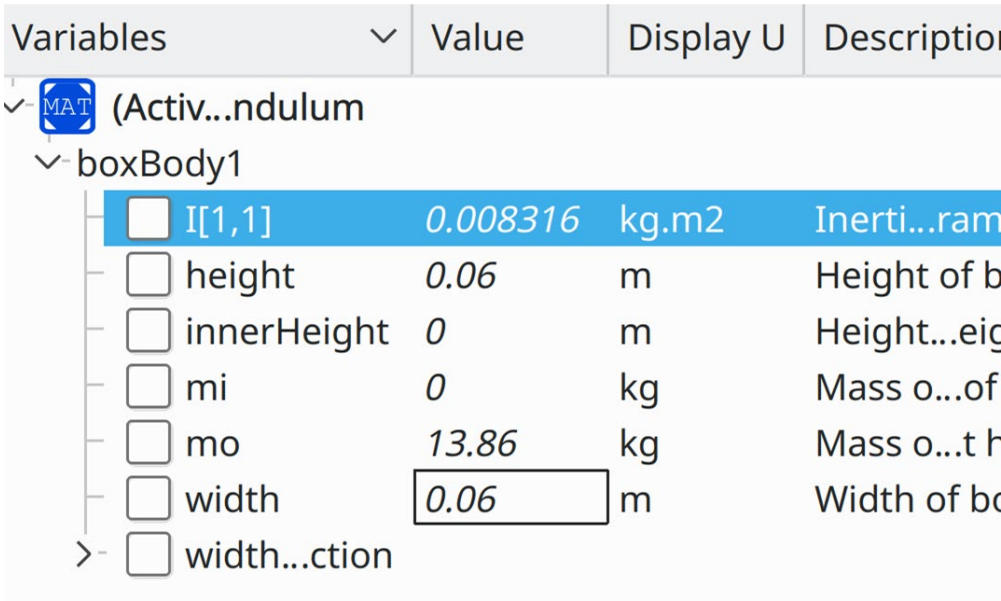
**Martin Sjölund,**
**Adrian Pop, Adeel Asghar**
**Dept Computer and Information Science**
**Linköping University**

# Enhanced OM Debugger that can trace (and plot) which variables and equations influence a variable

## New functionality to show direct variable dependencies



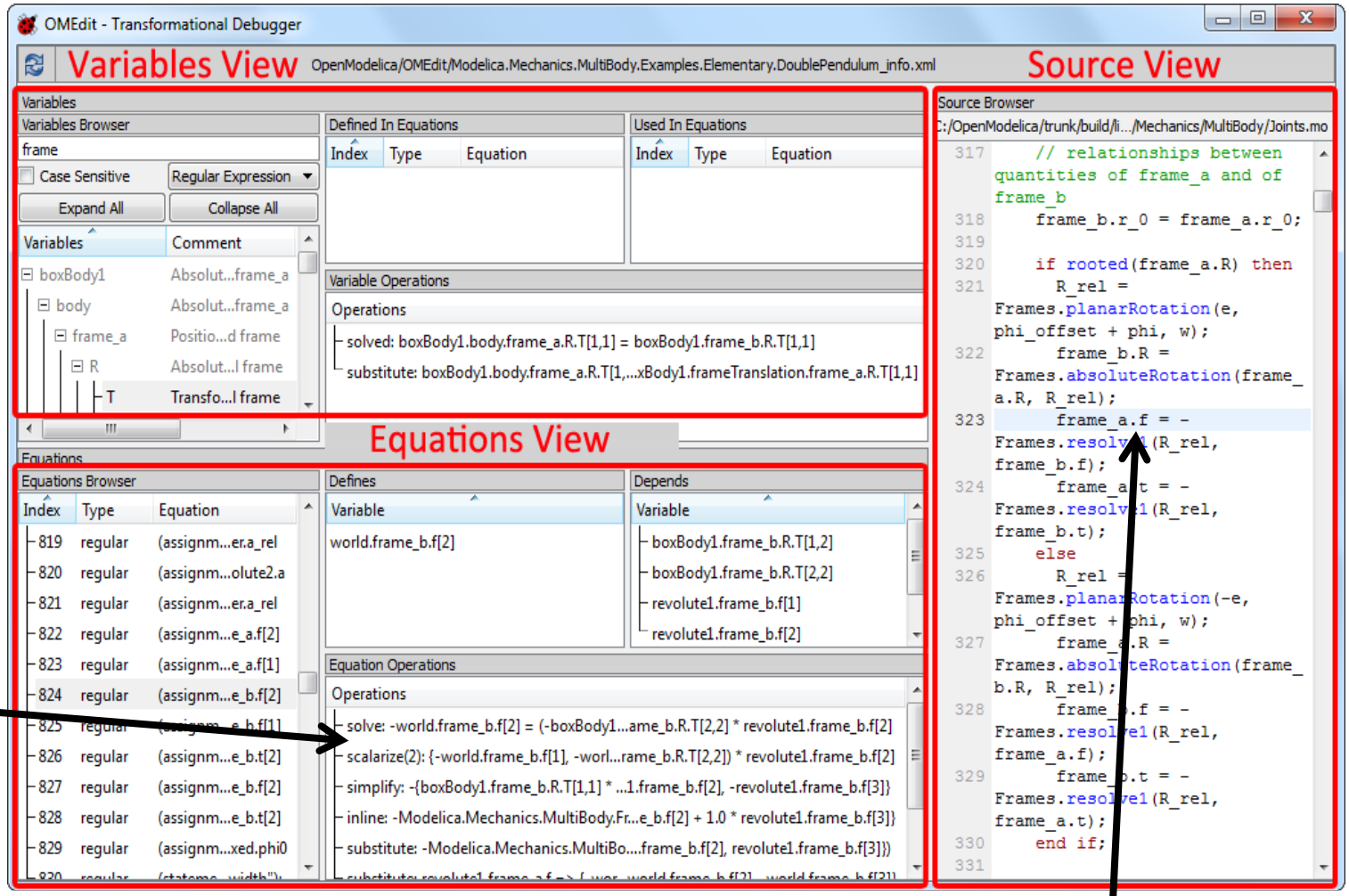

### List of Variables directly influencing:



# Integrated Static-Dynamic OpenModelica Equation Model Debugger

**Efficient** handling of **Large** Equation Systems

Showing equation transfor mations of a model:




**Mapping dynamic run-time error to source model position**



MODELICA pelab

# Digital Twins using Modelica and OpenModelica

**Collaboration with**
**Modelicon InfoTech, Bangalore, India**
**GI-LIFT AB, Linköping, Santa Anna**

**Adeel Asghar, Martin Sjölund, Peter Fritzson, Lennart Ochel, Arunkumar Palanisamy**

# More Sustainable Forestry – Digital Twin of Balloon-Assisted UAV – Collaboration with GI-LIFT AB and Modelicon

**Avoid clear-cut damage**

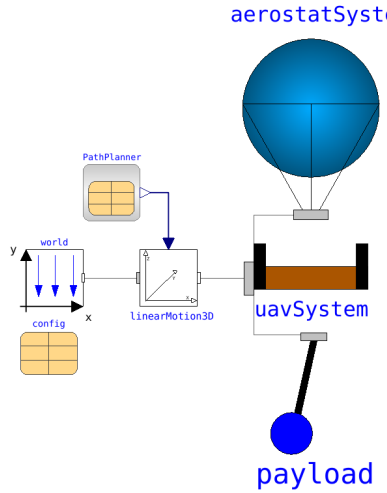
**Instead high-powered Electric Ballon-assisted UAV lifting system**
(patent pending, GI-LIFT)

Timber

Branches


Whole tree

**Digital Twin Using OpenModelica**



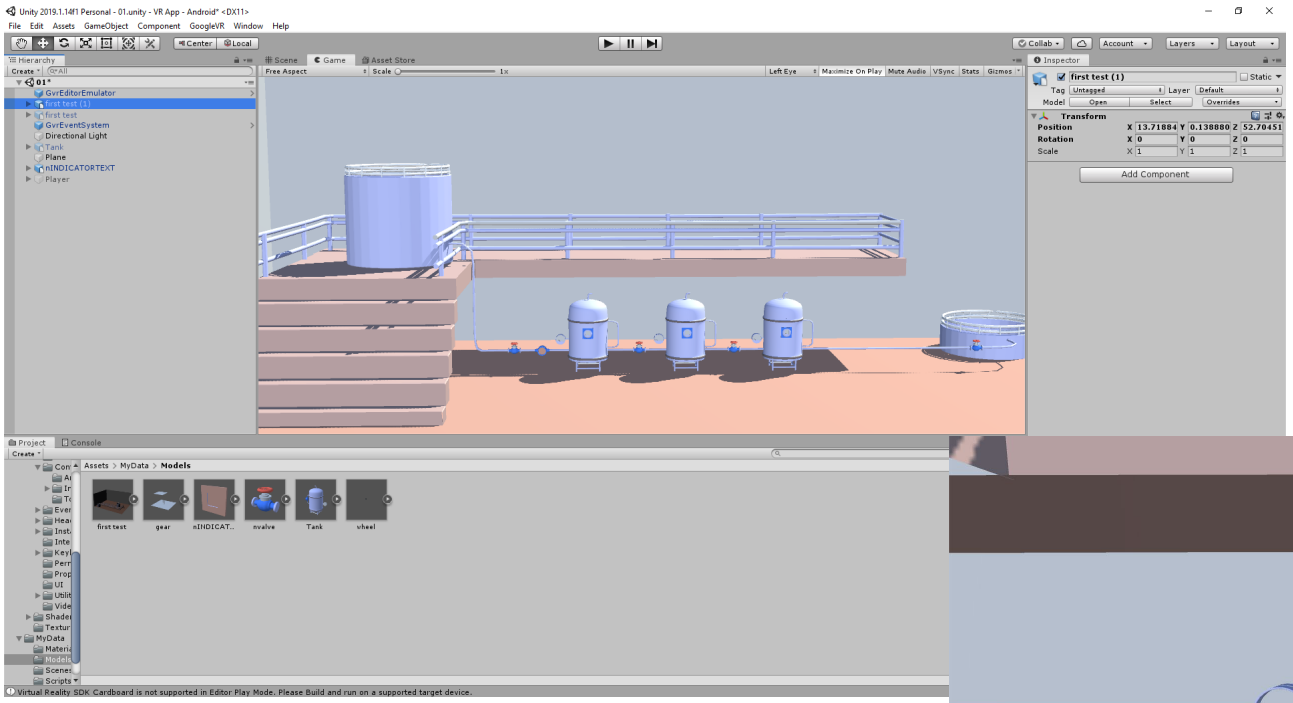



MODELICA pelab

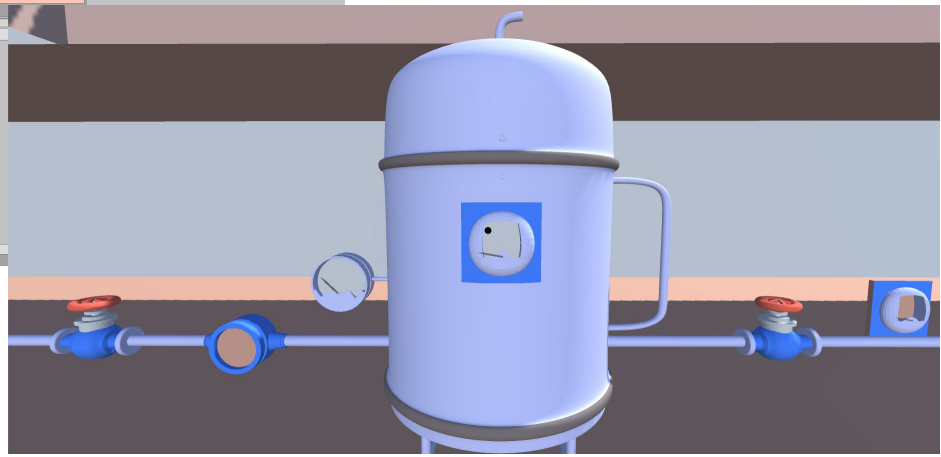
# Test-Flight of Balloon-Assisted UAV – Outside Linköping – by GI-LIFT AB

# Integration with Unity 3D Visualization in VAL – Virtual Automation Lab

Development environment integrated with OpenModelica



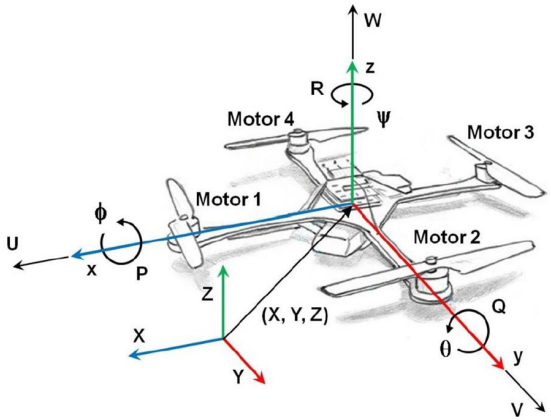
VR Model – Unity 3D

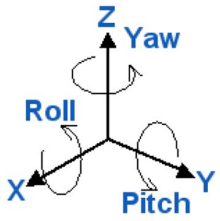


Developed by Modelicon and BMSCE in Bangalore, India

# Digital Twin OpenModelica Applications by Modelicon (Bangalore)
# Model-based Control of UAVs and Walking Robots

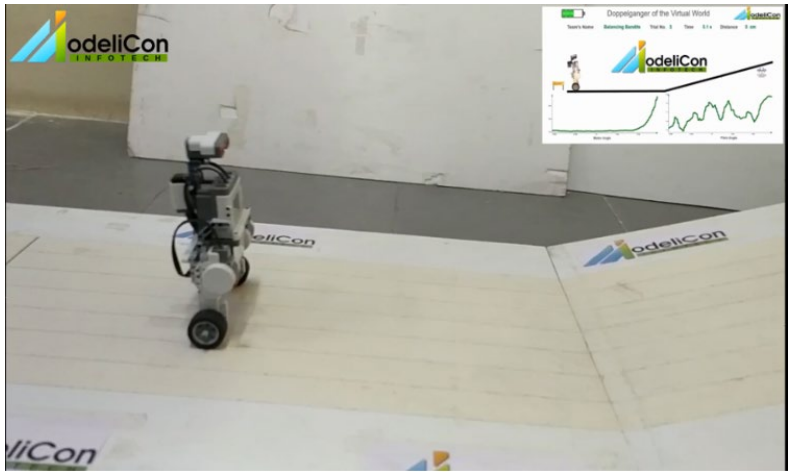
- UAV control and simulation
- Walking 2-wheel robot




**UAV**
Movie demo

**All models and control software done using OpenModelica!**




**Walking 2-wheel Robot**,

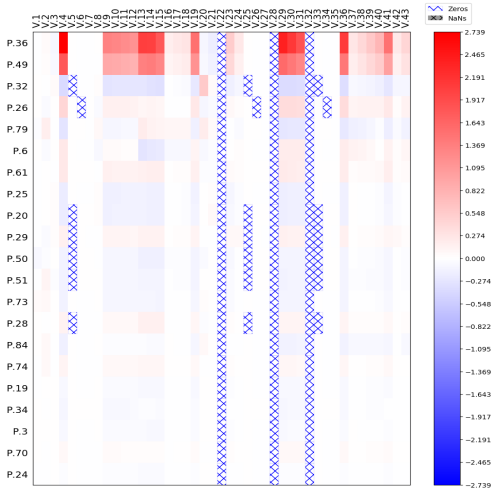
Movie demo

# Simultaneous Param-based Sensitivity Analysis and Robust Optimization (collaboration with Univ. Buenos Aires)

- To define a sensitivity experiment:
  - The state variable to analyze
  - The set of parameters to perturb
  - The allowed perturbation intervals for each parameter
- Main goal: pinpoint a small number of parameters that produce the largest deviations when perturbed within narrow ranges around their default values
- To select parameters and their intervals is not a trivial task
  - Responsibility relies completely on the expertise of the user
  - Enabling all parameters can lead to very costly experiments
- Use a top-N subset of parameters from a ranked list
  - obtained using individual parameter-based analysis
- Using CURVIF robust derivative-free model building method for few function evaluations
- Heat-map visualization of parameter influence

Paper published at EOOLT 2017 (prototype)

Part of OpenModelica since 1.17






MODELICA pelab

# Data reconciliation in OpenModelica

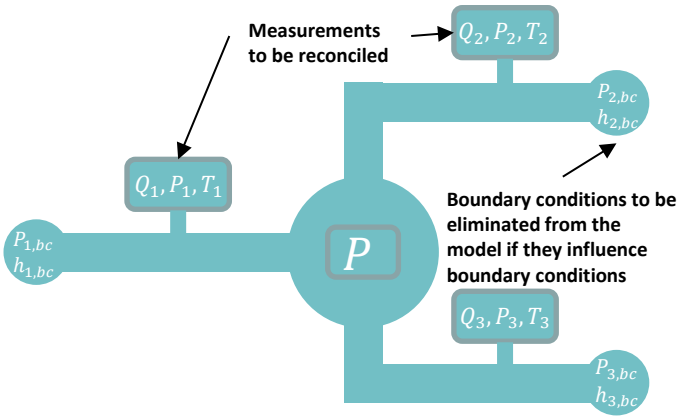
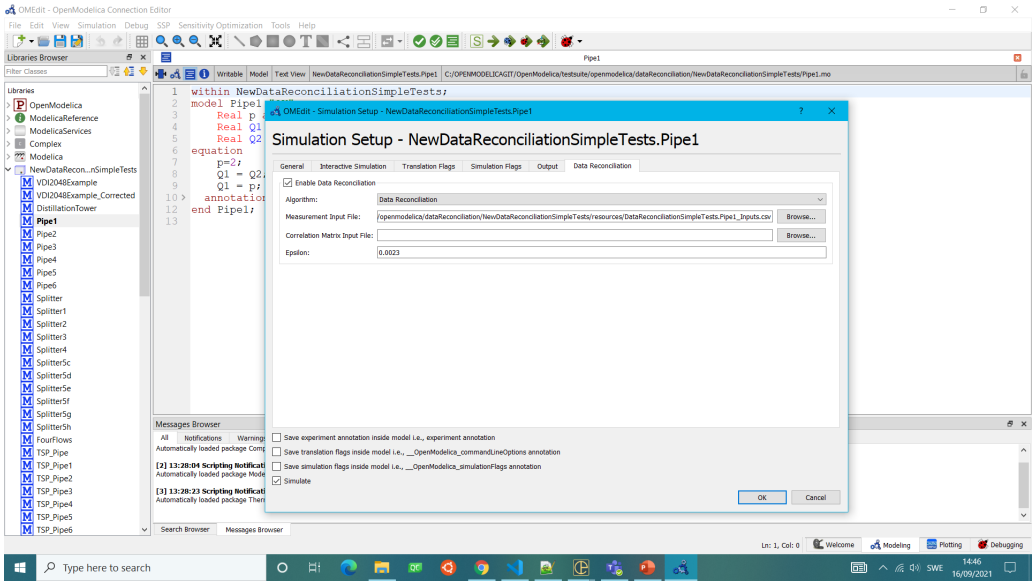



**Figure 1. Splitter**



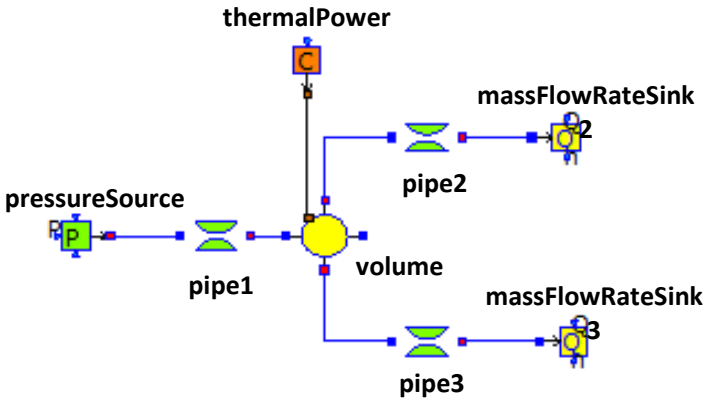




**Figure 2. ThermoSysPro model of the splitter in OMEdit**

- **Uncertainty reduction in measurements**
- **Available in OpenModelica**

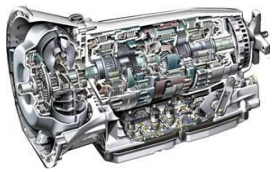

Daniel Bouskela
Audrey Jardin
Arunkumar Palanisamy
Lennart Ochel
Adrian Pop

# Co-simulation, FMI, Model Composition

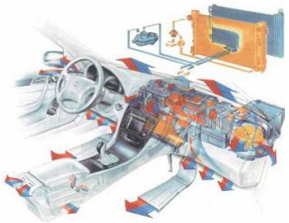
Lennart Ochel, Robert Braun, Adeel Asghar, Adrian Pop,
Arunkumar Palanisamy, Amin Bajand,
Peter Fritzson

# General Tool Interoperability & Model Exchange Functional Mock-up Interface (FMI)

Engine with ECU

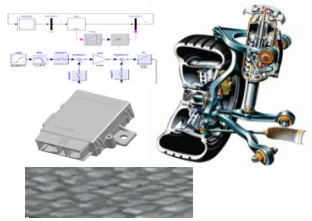


Gearbox with ECU



Thermal systems

Automated cargo door



Chassis components, roadway, ECU (e.g. ESP)

etc.

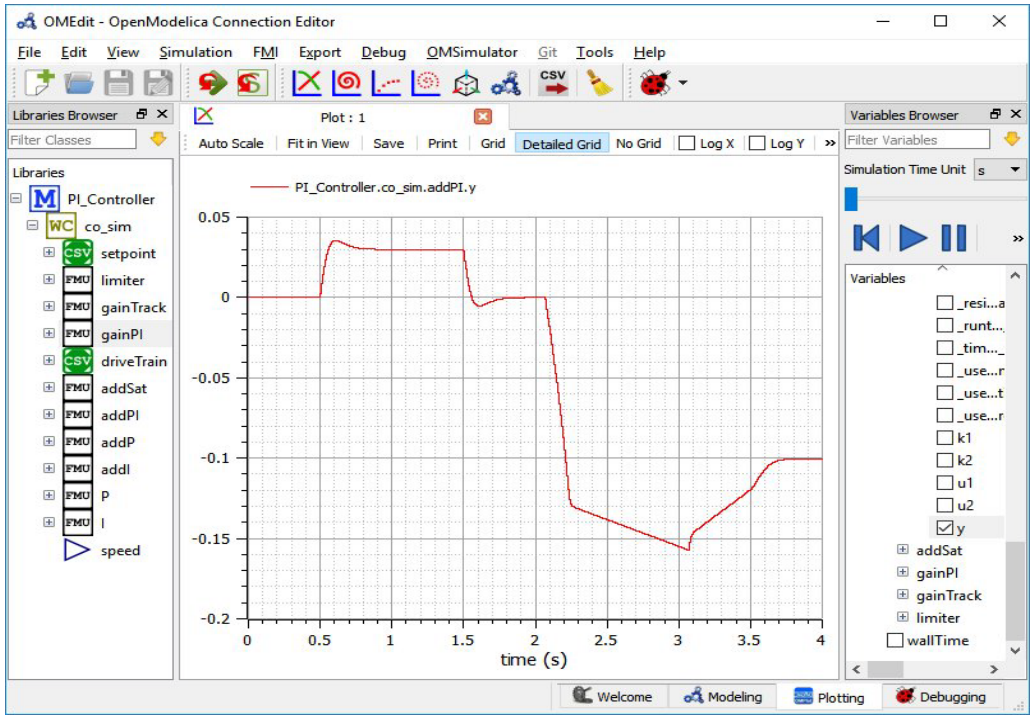
**functional mockup interface for model exchange and tool coupling**

courtesy Daimler

- FMI development was started by ITEA2 MODELISAR project.
  FMI is a Modelica Association Project now
- **Version 1.0**
- FMI for Model Exchange (released Jan 26,2010)
- FMI for Co-Simulation (released Oct 12,2010)
- **Version 2.0**
- FMI for Model Exchange and Co-Simulation (released July 25,2014)
- **> 100 tools** supporting it (https://www.fmi-standard.org/tools)
- **Version 3.0**
- Work in progress

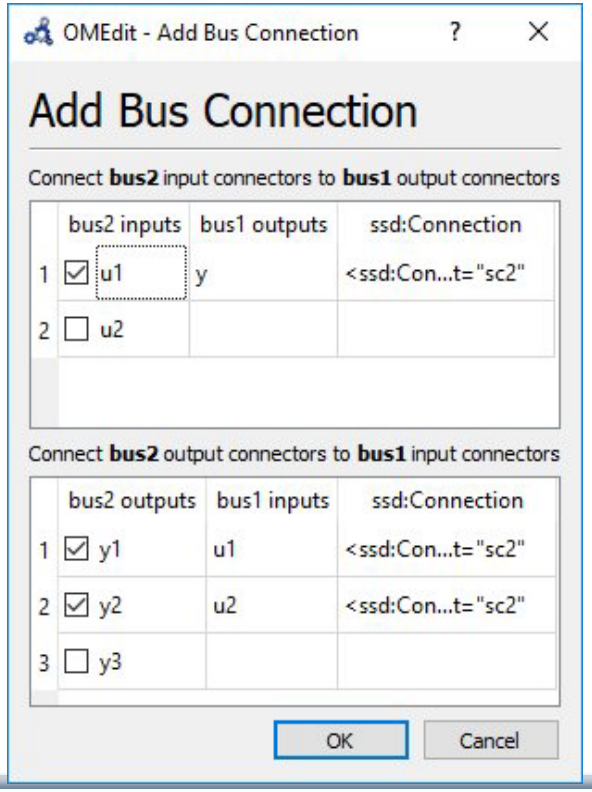
# Enhanced FMI Co-simulation, Run-time, and Master Simulation Tool

- General **Master** simulation tool OMSimulator part of OpenModelica
- 2024-2025 support for FMI3 and SSP2 in OpenSCALING



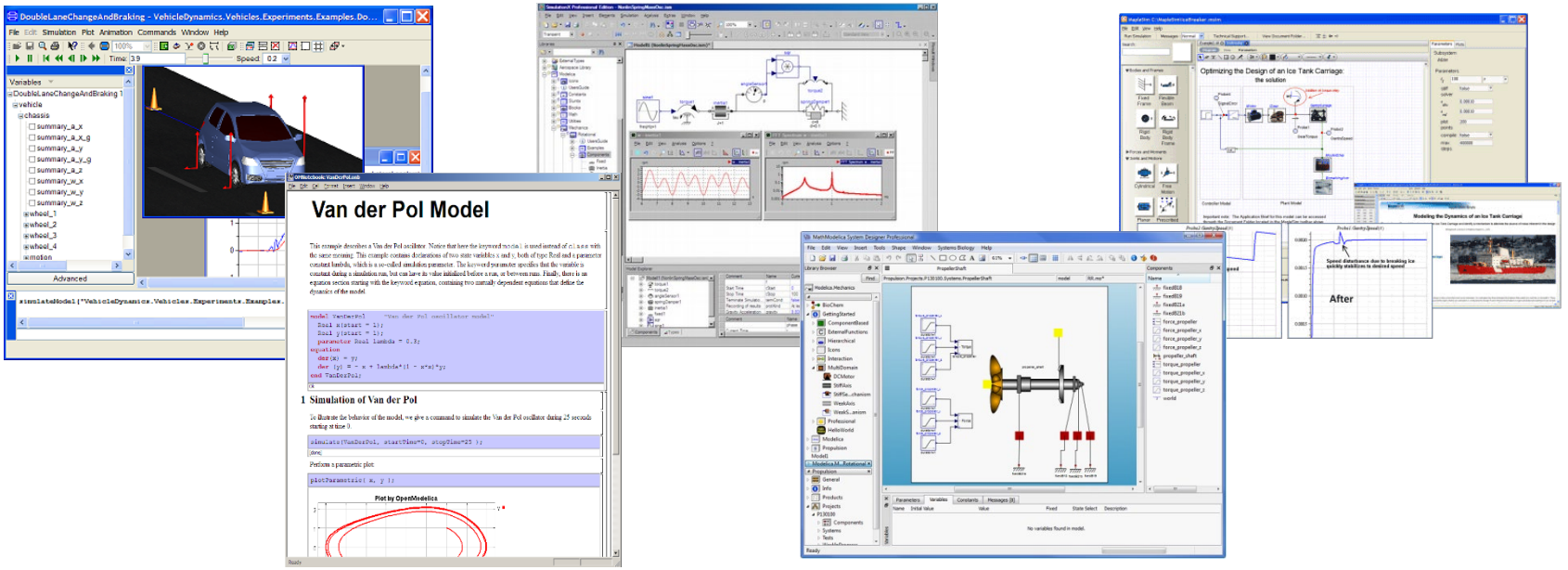

**FMI Simulation results in OMEdit**

**Adding SSP bus connections**



# Dynamic Verification/Testing of Requirements vs Usage Scenario Models EMBRACE (2020-2023), future projects

Lena Buffoni, Adrian Pop, et. al





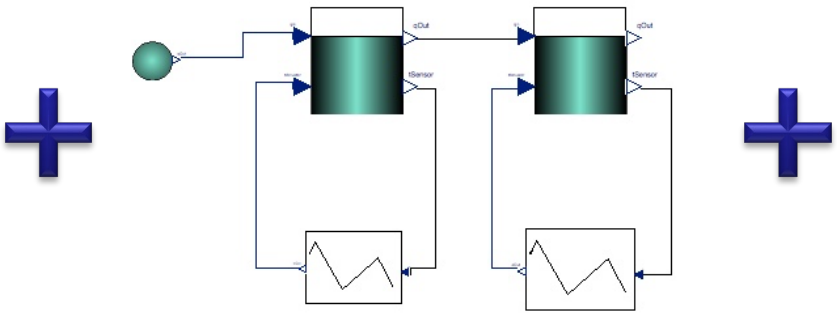

MODELICA pelab

# Testing a single verification model in Modelica

## In EMBRACE project – develop CRML standardized Requirement language

- Req. 001: The volume of each tank shall be at least 2 m3.
- Req. 002: The level of liquid in a tank shall never exceed 80% of the tank height.
- Req. 003: After each change of the tank input flow, the controller shall, within 20 seconds, ensure that the level of liquid in each tank is equal to the reference level with a tolerance of ± 0.05 m.
- …

Design alternative:
two tank model



Design alternative:
two tank model

Start with constant flow and increase at t=150

One possible test scenario

# CRML to Modelica compiler

STATUS:

- A prototype available
- Ongoing project with EDF on supporting CMRL libraries
- Ongoing work on supporting CMRL modeling in OMedit
- Ongoing work on probabilistic aspect modeling in CRML
- A new application in the works
- A thesis on the visualisation of simulation results planned this spring

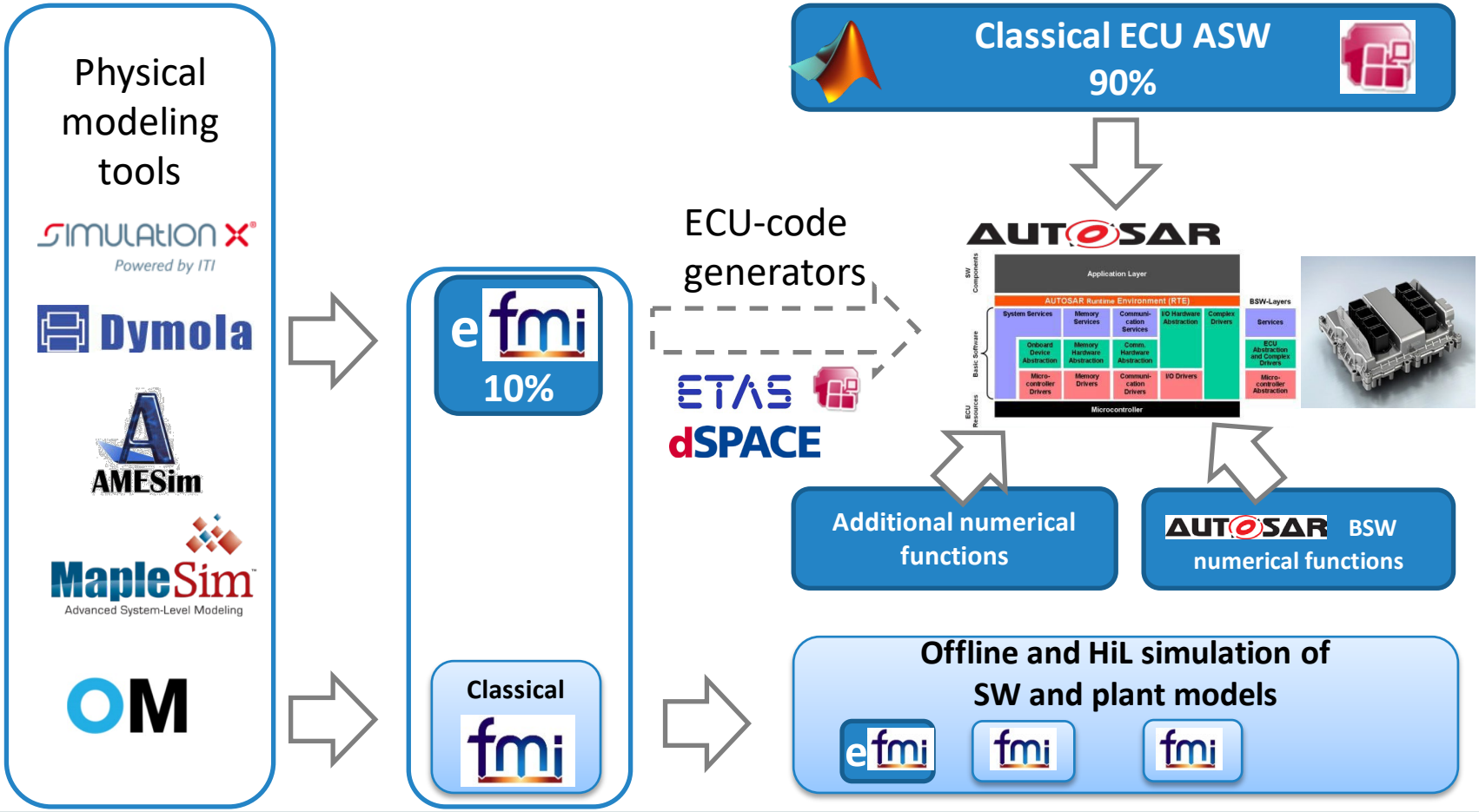
# Model-based Development Tooling for Embedded Systems

## Project EMPHYSIS, EMISYS

**EMbedded systems with PHYSIcal models In production code Software**

**Lennart Ochel, Martin Sjölund, Adrian Pop, et al**
**Dept Computer and Information Science**
**Linköping University**

# Bridging the gap between modelling and simulation tools and embedded systems through a new interface definition (eFMI)




Seamless model-based design of ECU-Software based on physical models.



MODELICA pelab

# Status

- EMPHYSIS – finished 2021, FlatModelica specification draft
- EMISYS – finished 2022, integration with Volvo Trucks tooling: EAST-ADL tooling and Adapt runtime
- New project proposal involving FMI/eFMI, OpenSCALING got funded.

# Embedded Systems Real-time Control Code Generation Using OpenModelica

Martin Sjölund et al
Dept Computer and Information Science
Linköping University

# OpenModelica Code Generators for Embedded Real-time Code

- A **full-fledged** OpenModelica-generated source-code FMU (Functional Mockup Unit) code generator
  - Can be used to **cross-compile FMUs** for platforms with more available memory.
  - These platforms can **map** FMI inputs/outputs to analog/digital I/O in the importing FMI master.
- A very **simple code generator** generating a **small footprint** statically linked executable.
  - Not an FMU because there is no OS, filesystem, or shared objects in microcontrollers.

# Large-Scale Collaborative Cyber-Physical System Development in the Cloud

**Abdelazim Hussien, Adrian Pop,**
**Peter Fritzson, Lena Buffoni, Martin Sjölund**

**Dept Computer and Information Science**
**Linköping University**

**ELLIIT project - in cooperation with Lund University**

# Large-Scale Collaborative CPS Development in the Cloud

- Collaborative **cloud-based tools and methods** for integration and testing in complex large-scale cyber-physical systems development

2024

- published new metaheuristic algorithm for cloud scheduling to improve efficiency and sustainability of cloud-based simulations

2025

- Licentiate defence
- Julia metaheuristics package with 100+ algorithms
- Improvements to automatic parallelization of Modelica simulation code via ParModAuto (Mahder) – better optimization using metaheuristics
- More cooperation with Lund on CodeProber – OpenModelica integration

# Energy Mega Game

## (Att vända strömmen – ett megaspel för ökad förståelse av energisystemet)

**Björn Johansson*, Ola Leifler, Lena Buffoni,
Ola Uhrqvist, Magnus Persson, Emily Hofstetter**

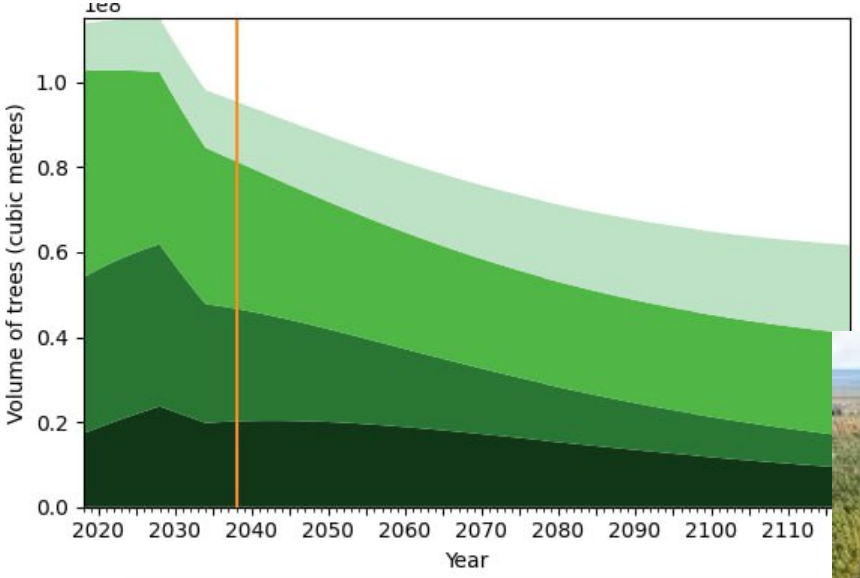
**Dept Computer and Information Science
Linköping University in collaboration with
Jönkoping University and Skövde Högskolan**

**Project supported by Energimyndigheten**

# Project specifics

- Goal - give a better understanding to the participants of the different perspectives on energy, climate, sustainability through a workshop in a mega-game format
- 3 year project 2021-2024
- LiU is investigating the use of simulation in the mega-game format and how it can provide a better understanding of the impact of decisions made in the game
- 04/2025 – OpenModelica predictive simulations on Energy and Transportations are used in a game with 170 students
- Research work on sustainable food cycle modeling ongoing

# The future of the forest




- The rush to replace fossil fuel with biomass to reduce emissions has led to overcutting of the forest
- The simulation shows that we are on track for an irreversible deforestation scenario

# Open standards for SCALable virtual engineerING and operation

## OpenScaling project (started 2024)

Adrian Pop, Lena Buffoni, Martin Sjölund, Amin Bajand, Arunkumar Palanisamy

# Project specifics

- Energy efficient solutions require that the use and optimization of large-scale facilities be considered for full day cycles over a full year to account for seasonal weather conditions.
- OpenScaling aims to extend the open standards Modelica, FMI, eFMI, SSP, and related toolchains with support for:
  - LSS models, which require array and loop preserving code generation, and aspect-oriented object diagrams.
  - Standardized uncertainty quantification of model parameters across these standards and tool support for UQ in V&V workflows
  - Physics enhanced neural ODEs integration with Modelica
  - Connection causality, array sizes and neural network parameterization of FMI to enable models being changed without regeneration, enabling life-time improvements of (runtime adaptable) digital twins of LSS.
- A PhD thesis started in the context of OpenScaling

# Thanks for Listening!