TUTORIAL ABSTRACT BOOKLET

Asian Modelica and FMI Conference 2026

2026-8-20

CONFERENCE ORGANIZATION COMMITTEE
166 Shuanghongqiao ST, Pingyao, Yuhang District, Hangzhou, China

Overview

SMArtInt+ - Hands-on AI: Generation and Integration of Neural Networks and Surrogate Models into Modelica

Duration: 3 hours

Tim Hanke, XRG Simulation GmbH

Abstract

There is an increasing need for hybrid models consisting of physical and machine-learning or surrogate model parts. XRG has developed the **SMArtInt+** Library, which enables the easy generation and integration of neural networks from different sources and types. This tutorial provides a hands-on introduction to the use of this library, which also includes a Python toolbox. The main use case is the creation and integration of a transient surrogate model for a thermodynamic application.

The tutorial focuses on the generation and utilization of training data, as well as the definition and training of neural network architectures, with the objective of developing high-performance surrogate models. Participants will be introduced to different concepts and approaches for creating data-driven surrogate models and will gain an understanding of the underlying methodologies and best practices.

A central part of the tutorial is the integration of the trained neural networks and surrogate models into Modelica models using the **SMArtInt+ Interface-Blocks**, allowing seamless substitution of physical components with learned surrogates. Additionally, several concepts of hybrid modeling that combine physical and data-driven approaches are introduced, with a strong focus on the practical integration of trained neural networks and surrogate models within the Modelica environment.

Finally, model performance is evaluated through comparisons with the original physical model.

Expected experience of participants:

- Basic knowledge of neural networks/AI and Python required
- Basic Modelica knowledge

Software requirements:

- Windows 10 or 11
- Dymola 2026 Refresh 1 or OpenModelica 1.26 (backup)
- Python IDE (PyCharm preferred)

Link to further information

https://xrg-simulation.de/en/seiten/smartint

FMI Beginners' tutorial - Exporting, Simulating, and Co-Simulating FMUs

Duration: 2.5-3 hours

Christian Bertsch, Robert Bosch GmbH; FMI Project Leader; Modelica Association Board Member

Abstract:

The Functional Mock-up Interface (FMI) standard plays a central role in enabling interoperability between simulation tools, particularly in cyber-physical and embedded system development. This tutorial introduces the core concepts of FMI and provides hands-on experience in exporting, simulating, and co-simulating Functional Mock-up Units (FMUs). Designed for both beginners and intermediate users, the tutorial is divided into three main sessions. The first session presents the motivation and history of FMI, key technical concepts, and an overview of FMI 3.0, followed by live demonstrations of FMU export using Dymola and Simulink. The second session dives into working with FMUs using Python and the FMPy library, including validation, simulation, and integration with Jupyter notebooks. Finally, we will set up co-simulations involving multiple FMUs, highlighting both graphical and command-line workflows, and demonstrating the impact of co-simulation parameters on accuracy and stability.
The tutorial concludes with a joint session summarizing the role of FMI in digital twins and highlighting real-world applications. Throughout the tutorial, attendees will have opportunities to ask questions and follow along with guided examples and shared materials.

Expected experience of participants
Basic familiarity with simulation concepts
Some programming experience (e.g., in Python)
Optional: Experience with Simulink, Modelica Tools, or co-simulation tools is beneficial but not required

Software requirements:

Python with fmpy[complete] (installation guide)
Optional: Matlab/Simulink
Optional: a Modelica tool such as OpenModelica or Dymola to export FMUs.
Optional: A Google account to use Google Colab for co-simulation notebooks

Link to further information: https://github.com/modelica/fmi-beginners-tutorial-2026/

OpenModelica Introduction and Hands-on Practice

Duration: 2.5 hours

Cuiting Peng, State Key Laboratory of Nuclear Power Safety Technology and Equipment, State Key Laboratory of Nuclear Power Safety Technology and Equipment, China General Nuclear Power Engineering Co.,Ltd.

Abstract:

This tutorial provides a comprehensive overview of the OpenModelica architecture and development environment. It covers the basics of the Modelica language, OMEdit GUI operations, and model compilation principles. The hands-on session will guide beginners through component library usage and drag-and-drop modeling to complete a full workflow from environment setup to simulation execution.

Software Requirements: OpenModelica 1.26.3

More Information Link: https://product.dangdang.com/12449354467.html (The introductory books we published）

Hands-on agentic modeling and best practices, with Wolfram System Modeler

Duration: 3 hours

Dr. Clément Coïc, partnering with Wolfram MathCore AB

Abstract

(Note: You can get a free trial version of Wolfram System Modeler for this tutorial.)

Agentic workflows developed for software development have changed our software development methods. Most of the trivial coding tasks are solved by AI in a fraction of time of what the software engineer used to take. Instead of coding, the focus of a software engineer is now on specifying the work to be done, choosing the right architecture, orchestrating the agents doing the work, and verifying the results.

Modelica – from its earlier definition – has always included a duality between the language and the user interface artifacts. And the language is no different for an LLM than a programming language like C, C++, Python, Julia, Rust, Wolfram, etc. Therefore, it took only a few months before a similar change happened for modeling and simulation as for software engineering. We do not model the same way as we did a year ago.

In this tutorial, we will first discuss the recent and upcoming evolution of the role of the simulation engineer towards a rather System Engineer type of work, assisted by agents. The skills of today's simulation engineer however remain critical for successful simulations. We will demonstrate this by hands-on examples. In a second step, we will leverage Wolfram System Modeler and their newest agent skills to develop a detailed Modelica in an agentic manner, pointing at best practices and pitfalls.

At the end of this workshop, you will be equipped to develop your own model in a much faster and efficient way, as AI-embracing simulation engineer.

Expected experience of participants:

- Basic Modelica knowledge
- Very basic understanding of using LLMs

Software requirements:

- Windows / Linux / MacOs
- Wolfram System Modeler 15 (get free trial before the workshop if needed)
- Download or check out the Wolfram System Modeler skills
- Access to your preferred LLM

Link to further information

- https://www.wolfram.com/system-modeler/
- https://www.linkedin.com/in/clementcoic/

EV thermal management modeling practice with TFS and YSLAB

Duration: 2 hours

Mr. Junjie Chen, Nanjing Yuansi SimTek Co., Ltd

Abstract

EV Thermal Management Modeling Practice with TFS and YSLAB

Duration: 2 hours

Instructor: Mr. Junjie Chen, Nanjing Yuansi SimTek Co., Ltd.

Abstract

(Note: Participants will receive free access to YSLAB from SimTek for the duration of this tutorial.)

Thermal management modeling and simulation have become critical to electric vehicle (EV) development. Modelica is well-suited for this domain thanks to its multi-disciplinary, acausal modeling capabilities, which naturally capture the physical behavior of batteries, electric motors, power electronics, and their associated cooling systems.

The **ThermofluidStream (TFS)** library, originally developed at DLR, introduces a novel modeling paradigm in Modelica that enables efficient simulation of large-scale thermal-fluid network models. **YSLAB**, developed by Nanjing Yuansi SimTek, is a browser-based Modelica simulation environment that lowers the barrier to entry for system-level modeling and interactive exploration.

This tutorial is structured in two parts. We begin with an introduction to thermodynamics modeling at the component level using YSLAB. Participants will then engage in a hands-on session to build an EV thermal management system model using the TFS library, from component instantiation to system assembly and simulation.

Prerequisites:

- Basic familiarity with Modelica
- Fundamental understanding of vehicle thermal management principles

Software Requirements:

- Operating System: Windows, Linux
- A YSLAB account with the ThermalFluidStreams library installed (access details will be provided at the start of the tutorial)

Further Information:

To be provided at the start of the session.