Welcome to Eclipse OpenMCx

EclipseCon 2021, Community Day - 25.10.2021

J. Balic,   josko.balic@avl.com
K. Schuch,   klaus.schuch@avl.com
S. Terres,   simon.terres@avl.com

Agenda

- Intro   J. Balic (2+ min)

- OpenMCx   K. Schuch (8+ min)

- Demo   S. Terres (10+ min)

- Q&A   All (10- min)

At **AVL** we believe that:…

A. <u>ADAS/AD virtual validation</u> is one of the hottest topics out there (automotive-SW-wise☺)

B. Setting up a virtual system is a **<u>complex task</u>** (environment, sensors, controls, vehicle, analytics,…)

C. Validation efficiency can be increased by a standard-based **open co-simulation** middleware (mix-and-match)

D. <u>There is no one-size-fits-all</u> toolchain (use-case specific: perception and fusion, planning and controls, system validation, driver experience, security and safety,…)

E. The add-ons to the middleware should be **<u>application driven</u>** (democratize plug-ins development: environment, HiL, cloud, sensor integration,…)
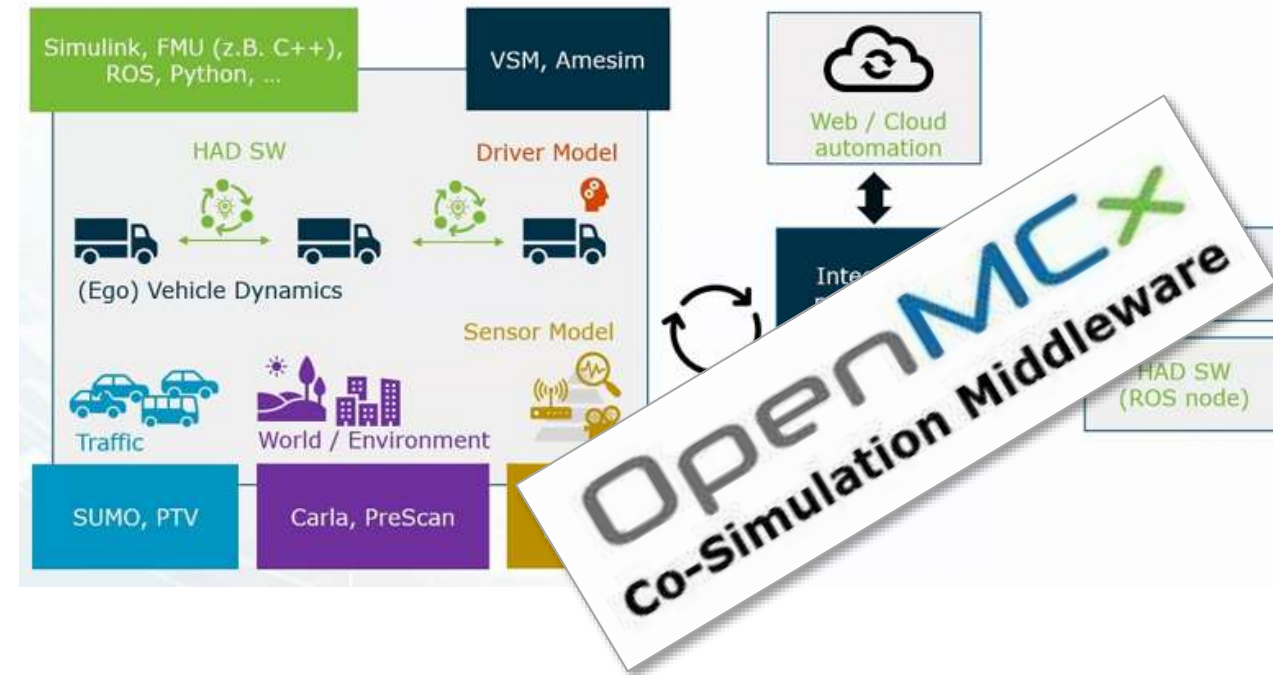
Intro

01/2018

·

06/2019

·
·
·

09/2021

·
·

The OpenADx Working Group wants to serve in the field of **software tools for** the realization of **autonomous driving** by defining **open interfacing standards** for software for use **in-vehicle based systems and in testing environments**, under the governance of the **Eclipse Foundation**.

"Eclipse OpenMCx" is an open, tool-neutral **co-simulation middleware** based upon simulation standards and formats, such as **FMI, SSP, DCP, OSc, OSI, etc.** aiming to support **advanced simulation applications** with a heterogenous toolchain in a distributed **collaborative development process**.

https://github.com/eclipse/openmcx
https://projects.eclipse.org/proposals/eclipse-openmcx

Our goal:

- Providing a referent implementation for promoting standards-based co-simulation methodology and enable the end-users to concentrate on their innovation process using use-case specific, best-in-class models and tool-chains.

- **Making the world a better place! At least a bit.☺**

- Co-simulation Framework based on ([Modelica Association](#)) standards: FMI, SSP, DCP, etc.

- Open for interfacing with other (quasi) standards and tools (python, OSI, ROS2, CARLA, SUMO, …)
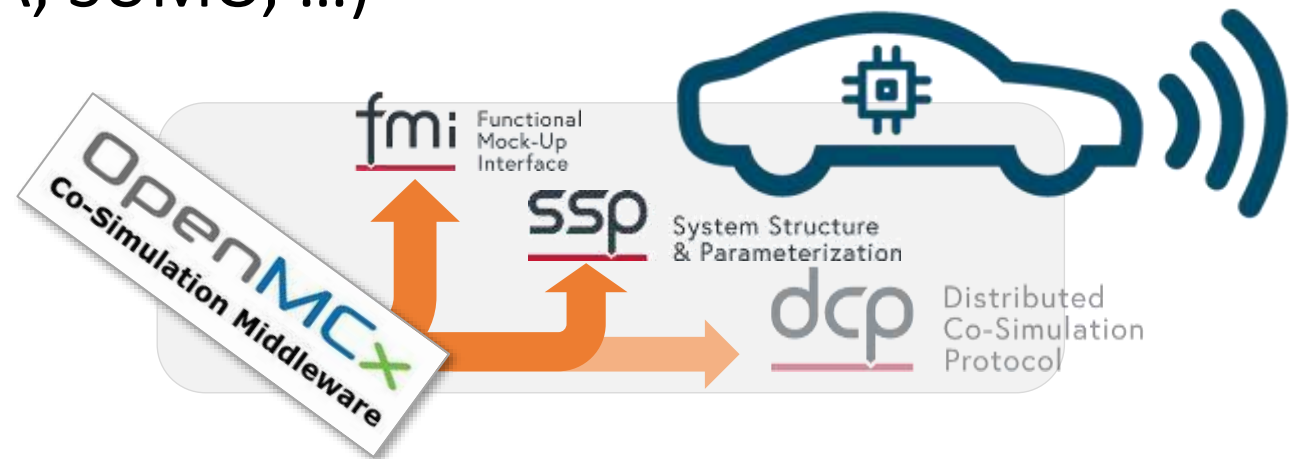
OpenMCx
Co-Simulation Middleware

- System Structure Definition (*.ssd) input file (www.ssp-standard.org)
  - annotations for run-time config (default values if undefined)

- Features:
  - Parallel (Multi-Threading) or sequential execution
  - Unit-conversion
  - Parameter support
  - Result writing
  - binary port support (FMI2.0 with [OSI Sensor Model Packaging](#))
  - …

- How to use (build, run, debug) OpenMCx?

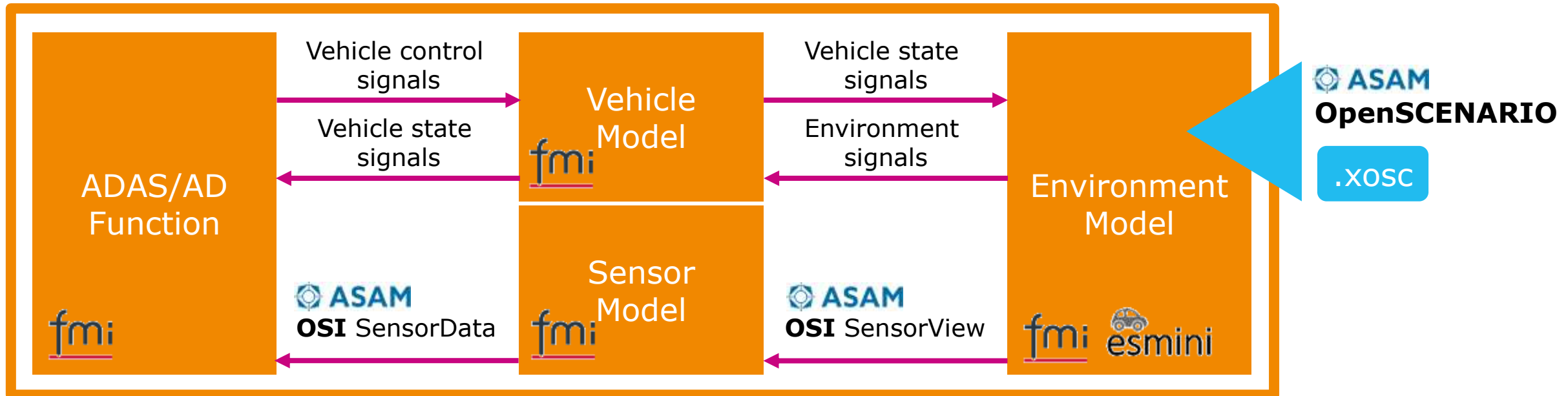https://github.com/eclipse/openmcx

LIVE SET-UP DEMO.

Klaus Schuch

Scenario-based testing of cyber-physical systems
powered by OpenMCx

Modelica Association · SSP

System Structure Definition (SSD)

- ADAS/AD Function
- Vehicle Model (fmi)
- Sensor Model (fmi)
- Environment Model (fmi · esmini)
- Vehicle control signals →
- Vehicle state signals ←
- Vehicle state signals →
- Environment signals ←
- **OSI** SensorData (ASAM)
- **OSI** SensorView (ASAM)

ASAM OpenSCENARIO
.XOSC

.ssd

Simulation
OpenMCx
Co-Simulation Middleware

Results
.csv
OSI trace

Analysis
Calculate KPIs
- Time to collision
- Perceived safety
- …
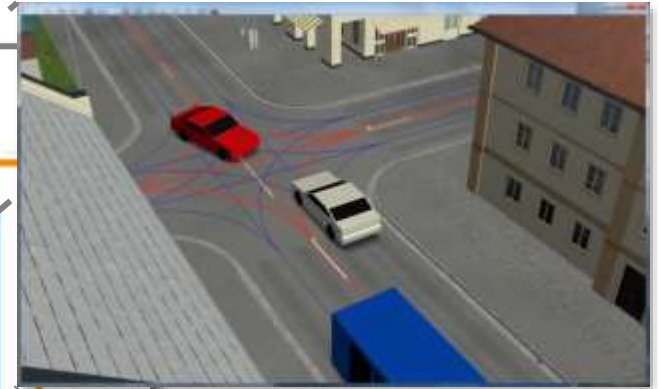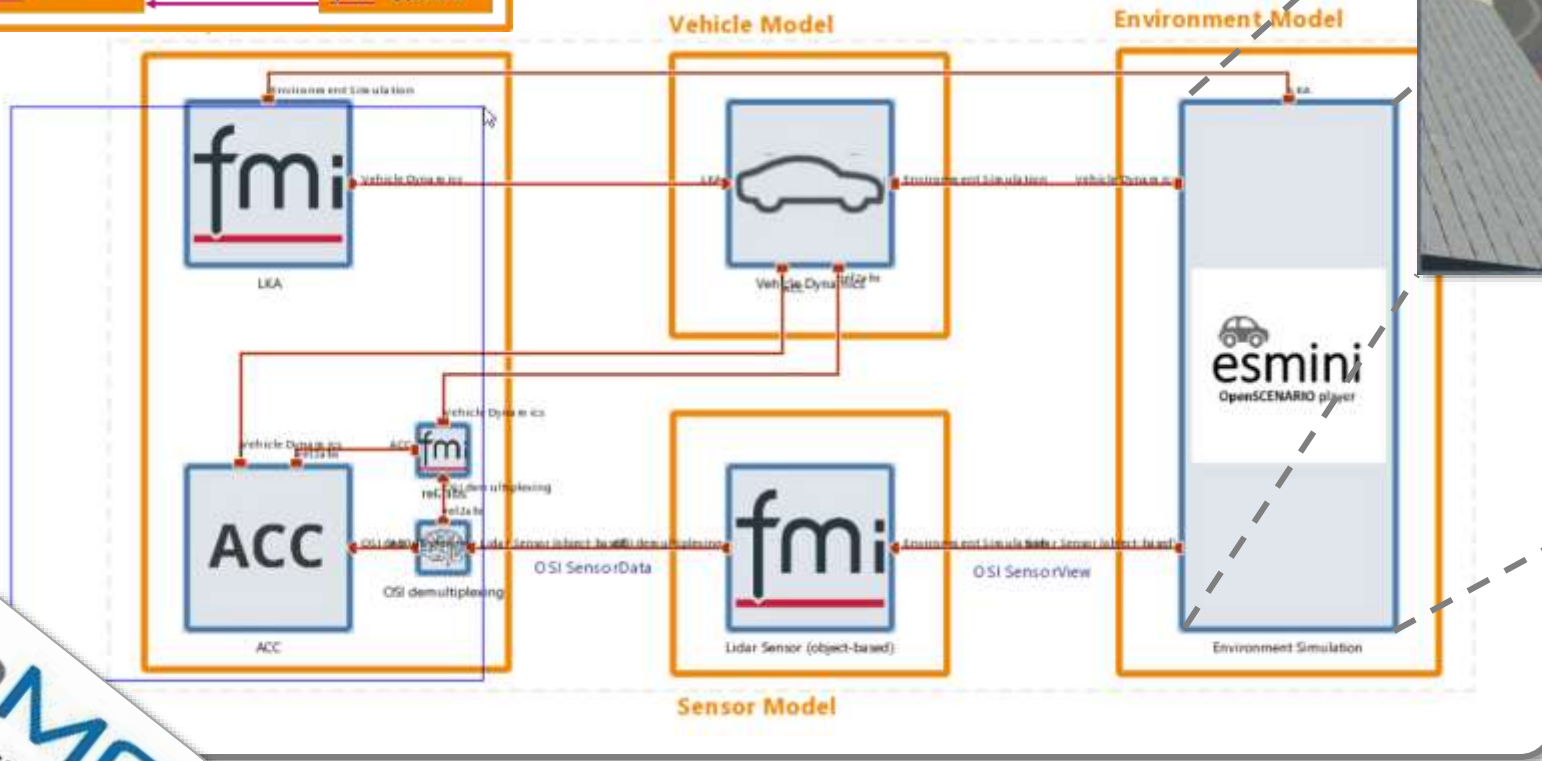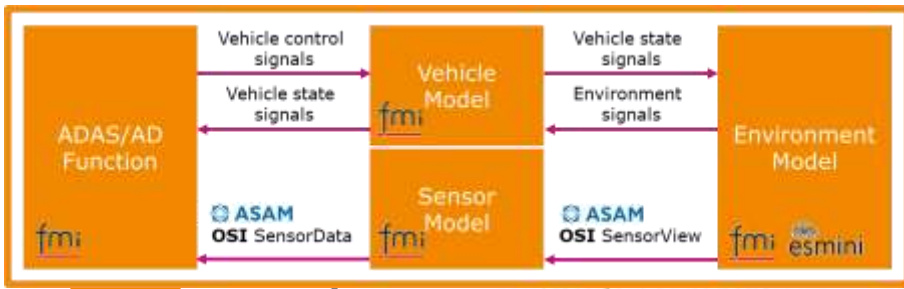
LIVE SIMULATION DEMO.

Simon Terres

https://github.com/eclipse/openmcx

Thank you

AVL

www.avl.com

Contact us:

klaus.schuch@avl.com

Come and contribute!

https://github.com/eclipse/openmcx