MA-Project “System Structure and Parameterization” – Current Status and Plans

presented by Jochen Köhler (ZF), Pierre R. Mai (PMSF)

T. Sommer

M. Najafi

M. Deppe

J. Köhler

J. Krasser

P. R. Mai

M. Nagasawa

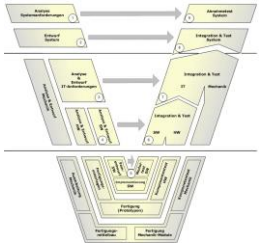
M. Henningsson

J.N. Jäschke

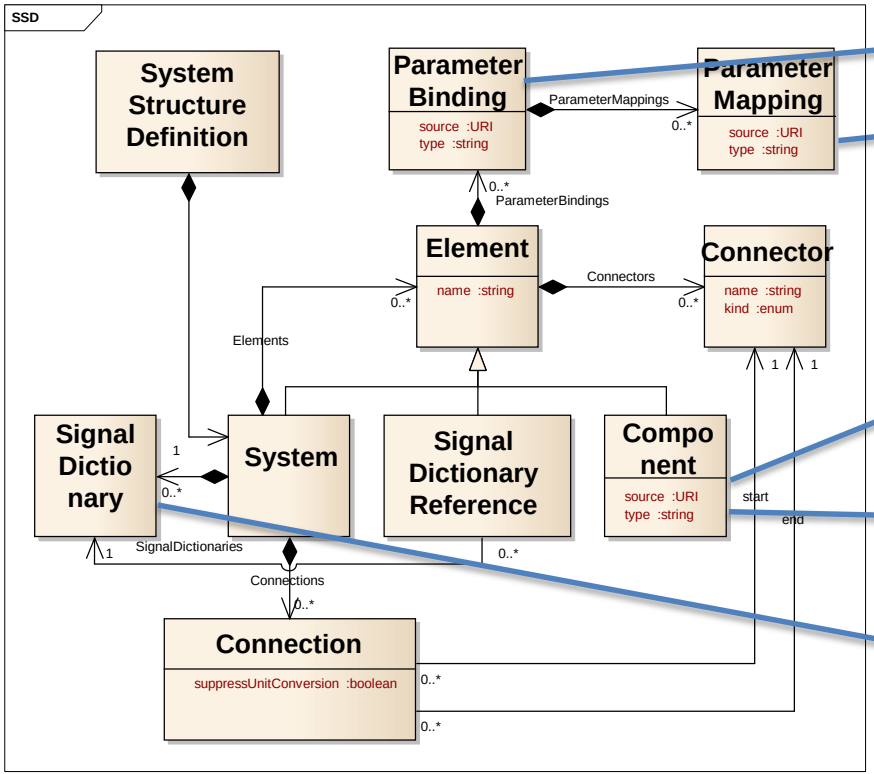
FMI User Meeting 2017-05-15
Prague / Czech Republic

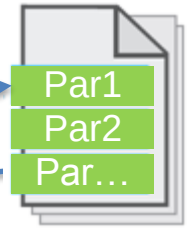
Main Purposes of SSP – Based on FMI standard

- Define a standardized format for the connection structure of a network of components (FMUs in particular).
- Define a standardized way to store and apply parameters to these components.
- The developed standard / APIs should be usable in all stages of development process (architecture definition, integration, simulation, test in MiL, SiL, HiL).
- The work in this project shall be coordinated with other standards and organizations (FMI, ASAM, OMG).



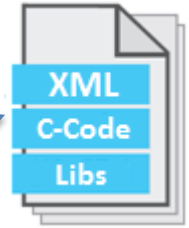
Overview of File Definitions





*.SSV

*.SSM



*.FMU

*.SSP

*.SSB

List of features

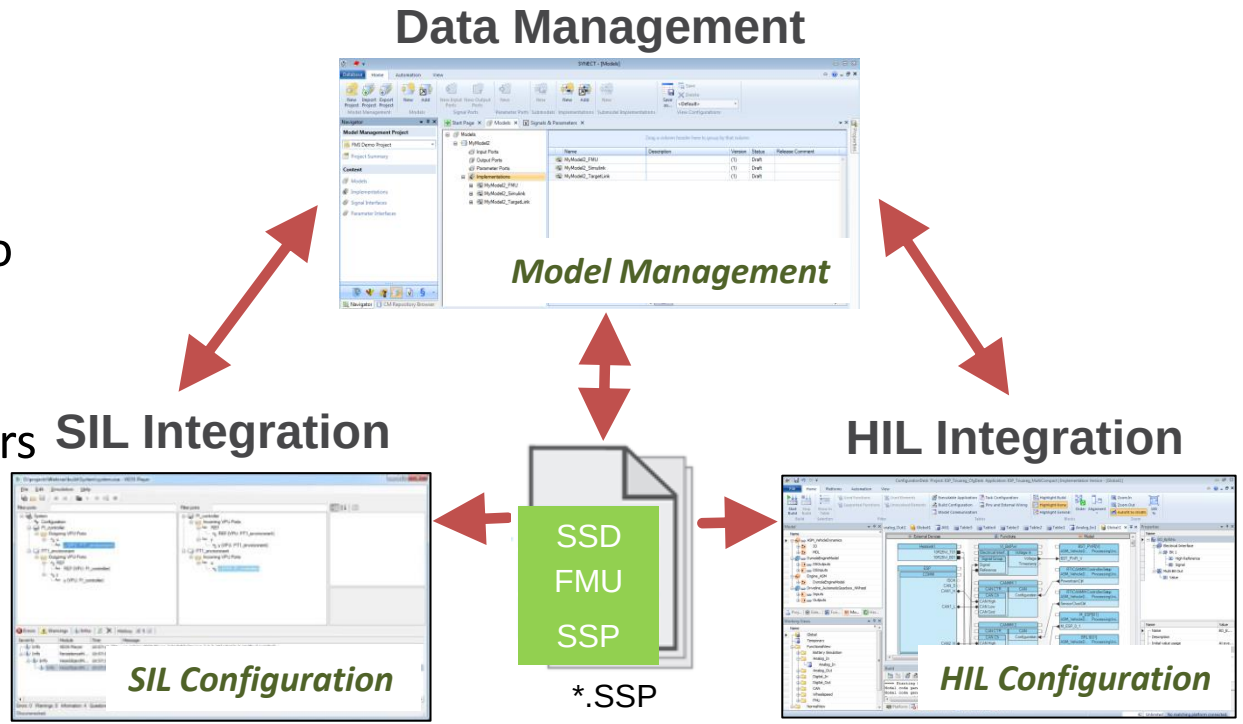
- Hierarchical description of systems of connected components
- Components: FMUs and external SSPs/SSDs, extensible to models, …
- Parameter bindings both at component and system-level, including transformations and name/unit-mapping
- Signal dictionaries support cross-hierarchical data pools (e.g for busses)
- Packaging of SSDs, FMUs, Parameters, … into one bundle (SSP)
- Light-weight support for variant handling at SSP level (multiple SSDs sharing components, parameters, resources)
- Optional exchange of graphical information (similar display across tools)
- URI references to all resources: Integration with other systems via URIs

Parameter Handling with Simulation Data Management Systems and Authoring Tools

- Results from workshop with SDM vendors:
  - Direct connection between SDM and authoring tools is not the preferred way
  - Parameter and system structure exchange could be done via file transfer (SSP standard)
  - Further measures will be taken to make a proof of concept

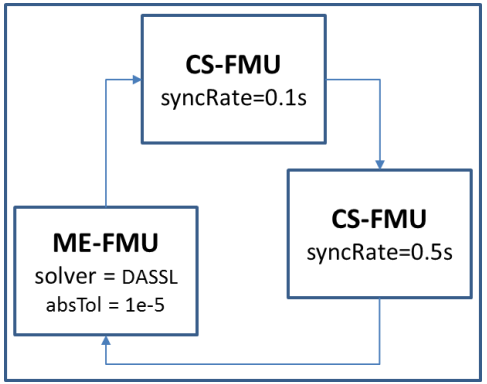
Integration of FMUs for SIL & HIL with SSP: Reuse of the System Structure

- The System Structure defined for SIL can be reused for HIL testing
- It becomes possible to reuse more models, configurations, tests, layouts and parameters
- A Data Management tool controls the lifecycle of the SSP



Potential further work

- Components can support more than FMUs/SSPs:
  - New ADAS Sensor Interfaces currently in development (Object Lists)
- Open Issues in SSP
  - Start values for Inputs
    - Subsystems / t=0
  - Initial state of the system
  - Experiment setup
    - Start time, stop time, selected results
    - Selection of solver, -settings, subsystem handling
  - FMU-instance specific settings
    - e.g. Sync rate, (Co-)Simulation mode, Import Settings
- Reliability check
  - Similar to FMI Crosschecks



Existing prototypes with support of SSP

- Model.CONNECT™ by AVL – Scope
- Integration Tool FMI Bench by PMSF
- Co-Simulation Player” (Cybernet)
- TLK MoBA Lab by TLK
- FMI Composer by Modelon
- FMI Kit – FMI Library for Simulink by Dassault
- SSP support by solidThinking Activate® by Altair

Requested Features from FMI

- Nice support of vectors / arrays for parameters and signals

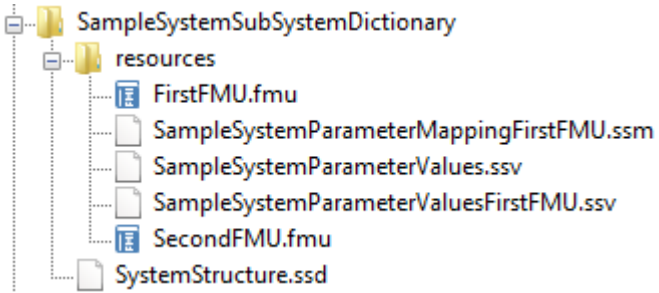
Current Roadmap for First Release

Topic	Due Date
Presenting on Modelica Conference	2017-05-15/17
End of Beta test phase	2017-09-30
Initiate website	2017-05-31
Review of existing documents	2017-05-31
Release Beta version	2017-06-01
Review of existing documents	2017-10-14
Release version 1.0	2017-11-30

BACKUP

File Definitions - System Structure Package (*.SSP)

*.SSP



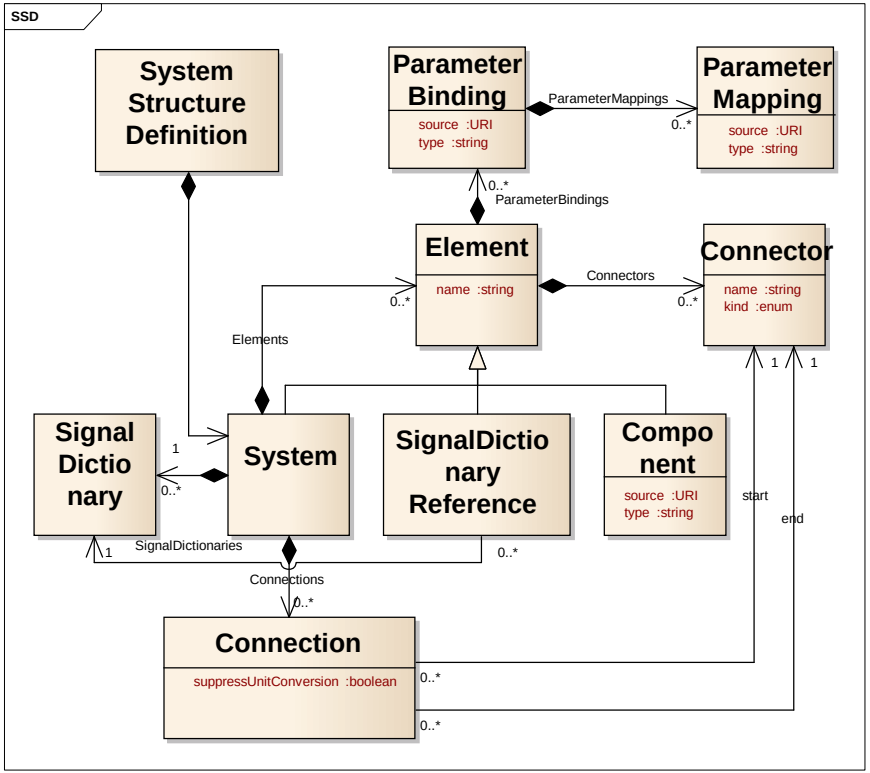
Use case

- Exchange of Complete Systems with Variants

Features

- All information (FMUs, system structure definition, parameters) can be stored in one archive (zip-file)
- Multiple SSDs in one SSP allows for variant modeling

File Definitions - System Structure Definition (*.SSD)



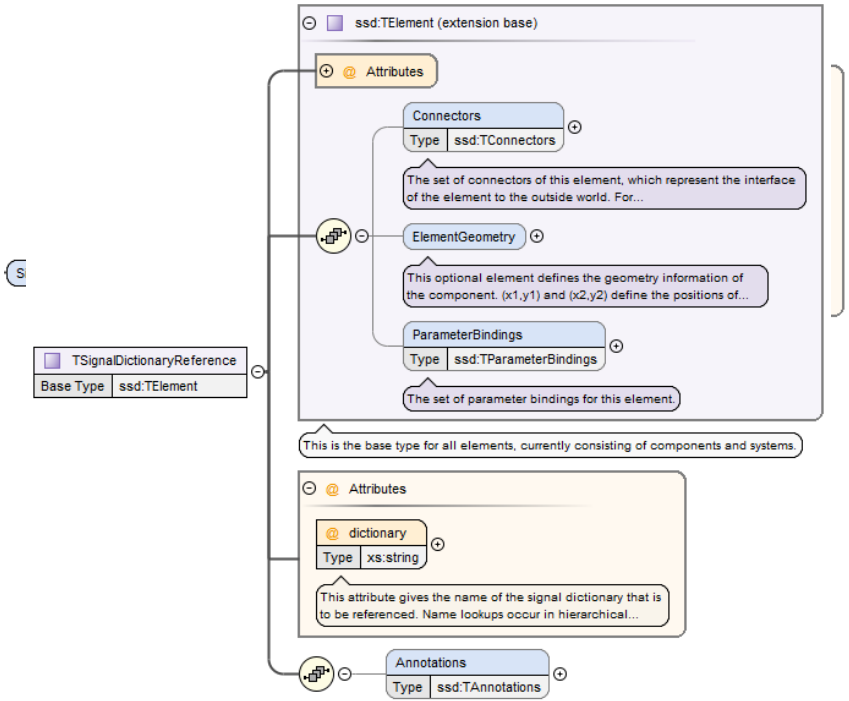
Use case

- Defining a Network of FMUs

Features

- Hierarchical sub-systems
- Empty components/FMUs as interface templates
- External resources via URIs: Both relative to SSD/SSP or absolute, e.g. via HTTP(S).
- Connections with unit conversions and optional linear/map transformations
- Optional: Diagram geometry

File Definitions - Signal Dictionaries (*.SSB)



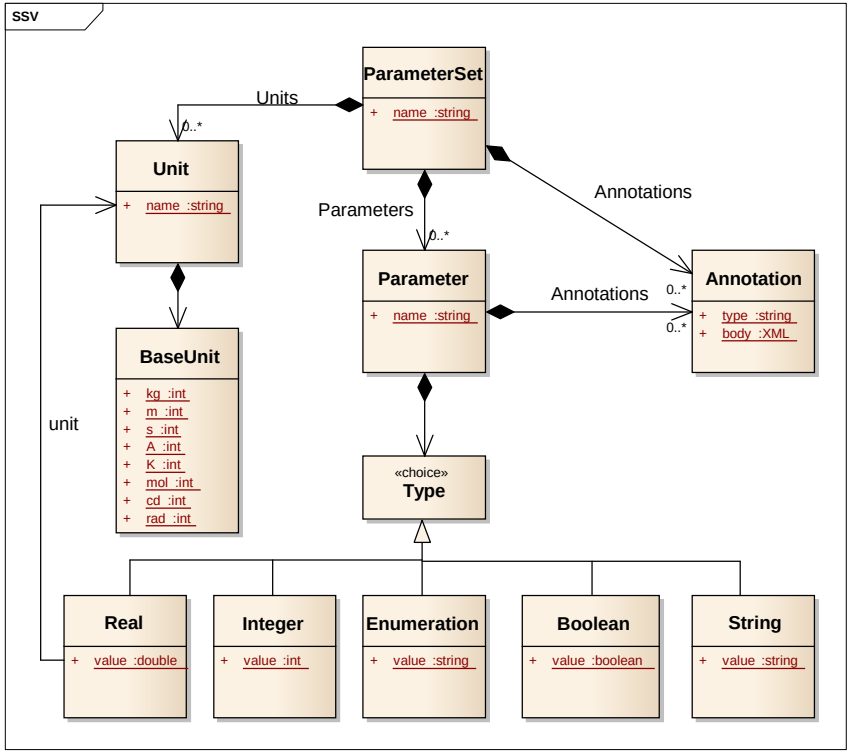
Use cases

- Collecting Control Signals in a Central Location

Features

- Causality is checked by tool automatically
- Crosses hierarchies without need for downward passing
- Well-suited for e.g. ECU control busses
- Can be inline or in external file

File Definitions – Parameter Values Data (*.SSV)



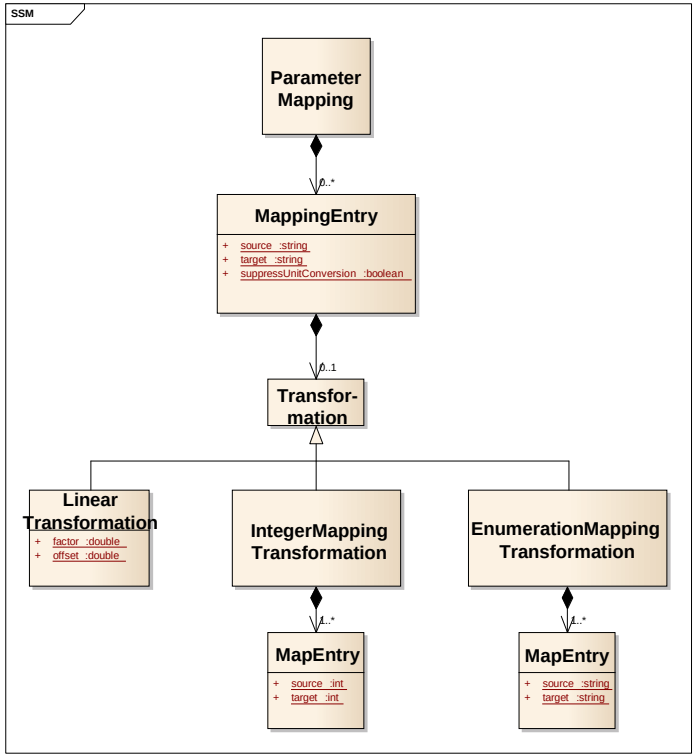
Use case

- Tool-independent Exchange of Parameter Data

Features

- Neutral exchange format between parameter sources
- Compatible to FMI standard
- Provides some meta data
- Access to param DBs via HTTP (-> Parameter API)

File Definitions - Parameter Mapping (*.SSM)



Use case

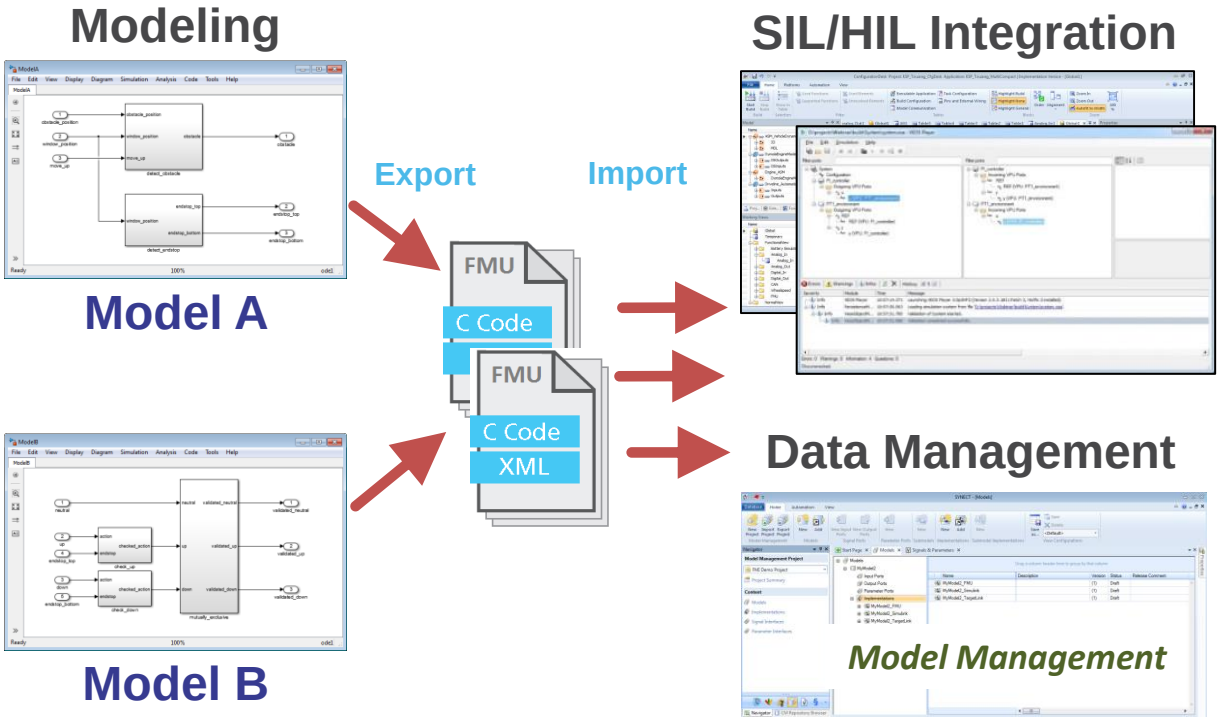
- Mapping Parameters to FMUs when the Parameter Names differ or Parameter Values require Transformations

Features

- Can be stored separately from System Structure and Parameter Data
- Can be inlined into SSD
- Optional manual linear and mapping transformations

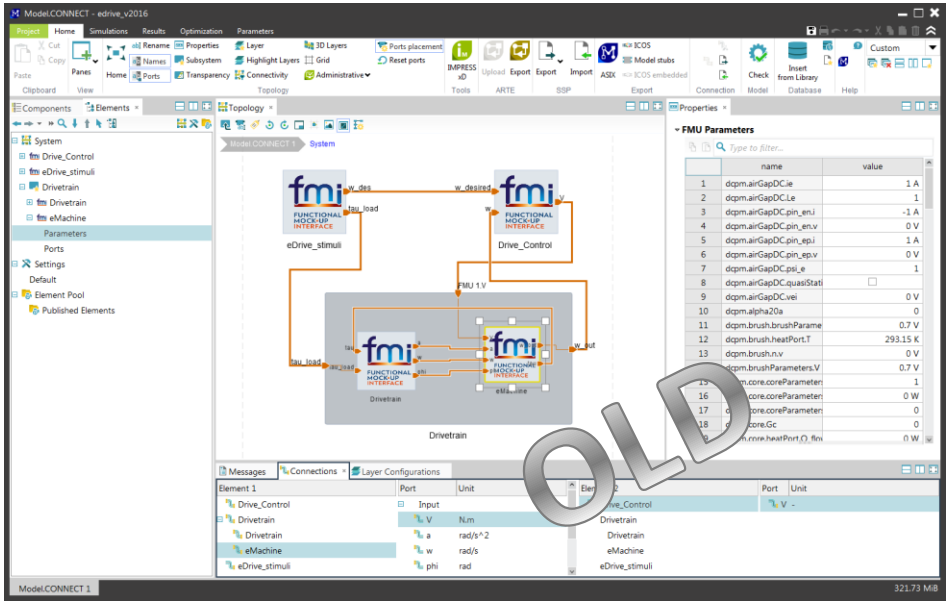
State of the Art: Integration of FMUs for SIL & HIL

- Integration tools are importing FMUs, Simulink-based models and ECU code
- A Data Management tool controls the lifecycle of all models
- The SIL/HIL integration results (System Descriptions) can not be exchanged easily among the tools



Prototypes – Integration Tool

- Model.CONNECT™ by AVL – Scope:
  - Simulation architecture set-up
  - Model integration
    (FMI and dedicated interfaces)
  - Execution
    (office and lab)
  - Model management
  - Handling system structure and parameter variants

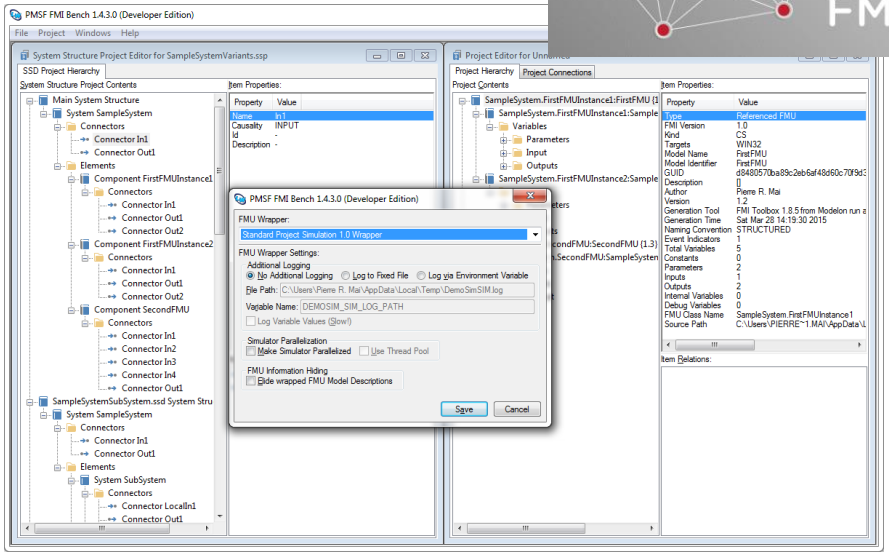


Prototypes – Integration Tool

- Model.CONNECT™ by AVL – SSP prototype:
  - Import and export of system structure (SSP packages)
    - Prototype supports multiple structure variants in the package
    - Mapping between the SSP variant handling and the tool-specific variant handling had to be implemented
    - Import-export roundtrip does not re-produce original ssp content
    - This is a consequence of the deliberately simple SSP variant handling concept
  - Import and export of graphical information
    - Overall layout information can be transferred via SSP.

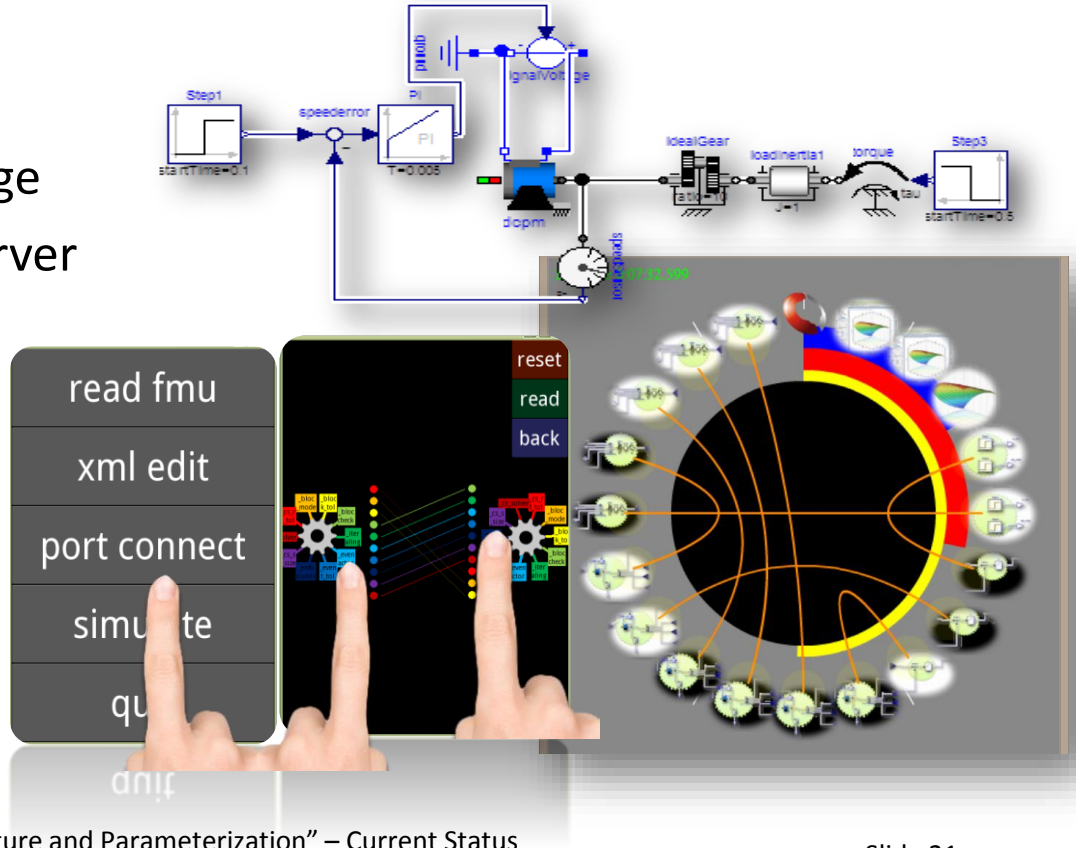
Prototypes: Integration Tool FMI Bench

- FMI Bench by PMSF: Workbench for FMUs
  - FMU Inspector & Editor, Integration, Automation
  - Export Integrated FMUs, Simulators, Remote FMU Execution, FMU-internal Parallelization, IP-Protection
- FMI Bench SSP Prototype
  - Direct Editing of SSDs, SSPs, incl. Variants
  - Generation of Native FMI Bench Projects from SSP Projects
  - Generation of FMU or Stand-alone Sim. from SSP
  - Parallelization
  - IP Protection



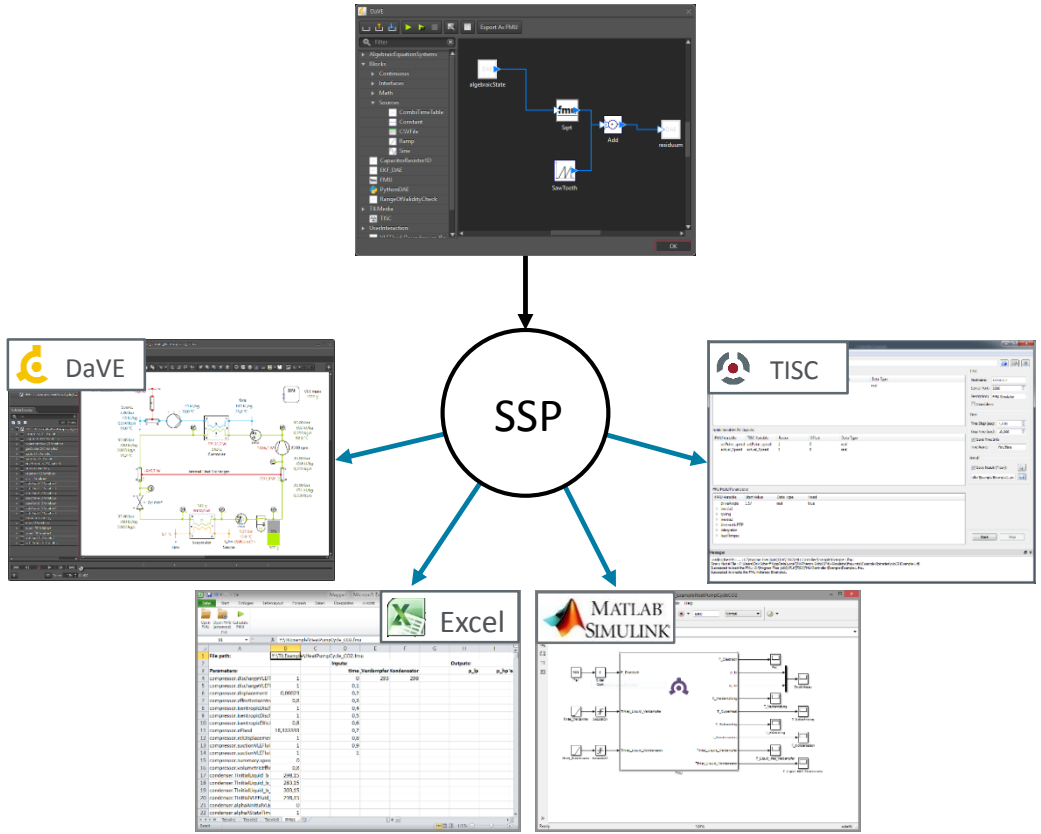
Prototypes – “Co-Simulation Player” (Cybernet)

- For Mobile Co-Simulation
  - SSP as an Online AP Package
  - Flash client and Python server
- Natural Interface
  - Ring connection geometry
  - Flexible 2D/3D Layout
- Parameter repository
  - Load sqlite.DB as a FMU
  - URI online access



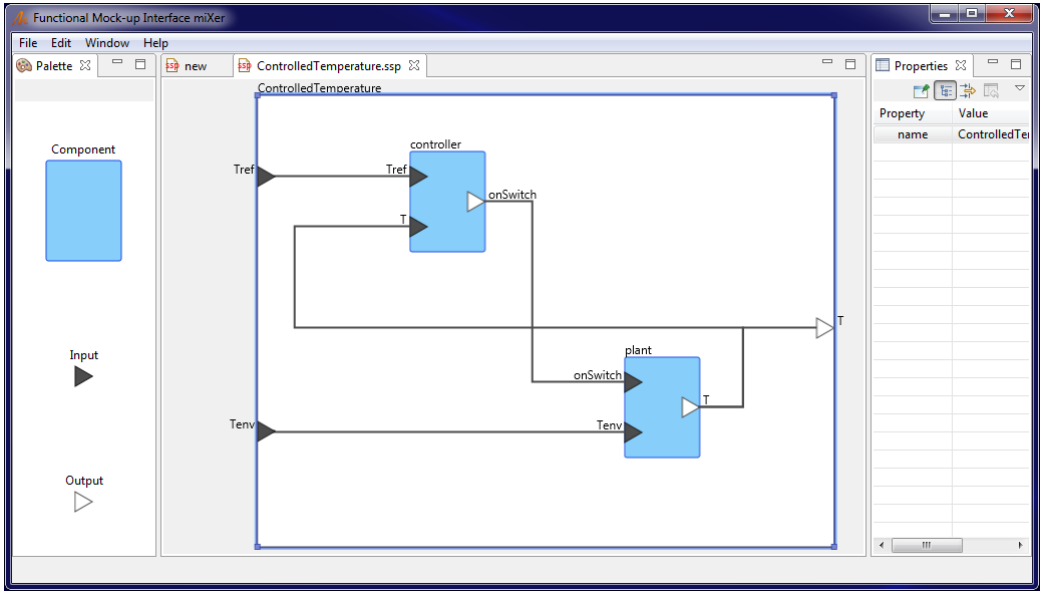
Prototypes – TLK MoBA Lab

- Focus:
  - FMI-based models
  - physical connections
  - thermodynamic environment
- Features
  - System Design & Model management
  - Simulation & virtual test bench
  - Optimization & parameter fitting
- Prototype for im-/export support of SSP in development



fMIXer by Modelon

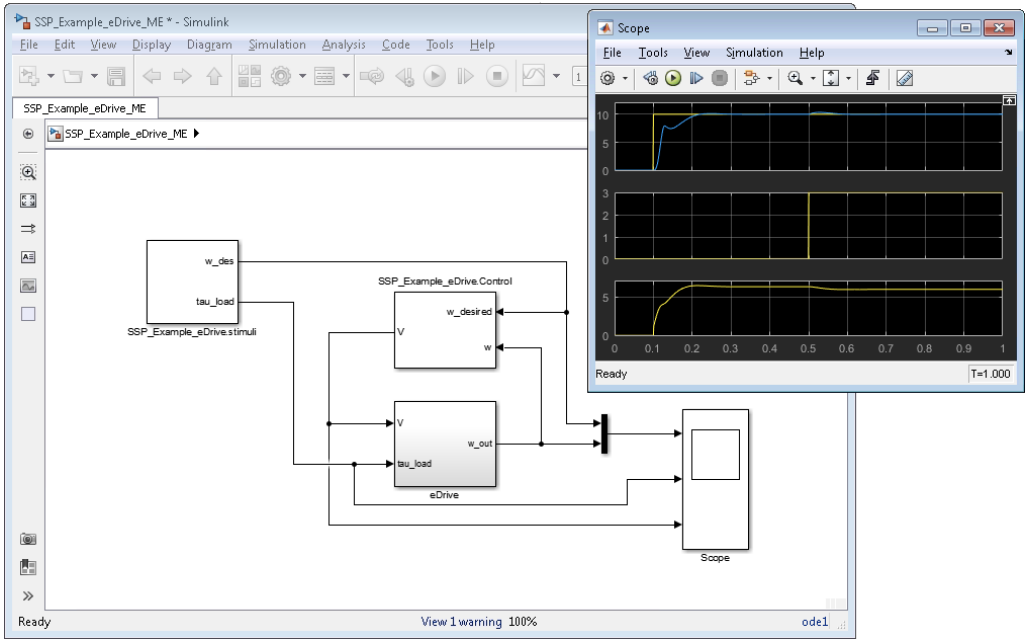
- Connect FMUs graphically
- Save as .ssp or export to FMU for simulation in any FMI-compliant tool
- View FMU information
- Manage parameters
- Convert ME to CS FMUs



FMI Kit – FMI Library for Simulink

FMI Kit by Dassault Systèmes

- Export models as FMUs
- Import of FMUs into Models
- Initial support of SSP



SSP support by solidThinking Activate®

- Leverage complete FMI 2.0 support (import/export, ME and CS)
- Simulation of connected FMUs, with handling of parameters
- Support of SSP format for both import and export (WIP)