

Exploitable Results by Third Parties

11004 MODRIO

Project details

| | |
|-----------------|---|
| Project leader: | Daniel Bouskela |
| Email: | daniel.bouskela@edf.fr |
| Website: | https://www.modelica.org/external-projects/modrio |

| Name: O3PRM editor | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ PRM (Probabilistic Relational Model) written in the O3PRM modeling language ▪ Observations and requests on some variables of the PRM | <ul style="list-style-type: none"> ▪ Syntactic editor for O3PRM language ▪ Bayesian inference engine | <ul style="list-style-type: none"> ▪ Probability distributions of the requested variables |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Supports <i>object oriented</i> PRM ▪ Will soon be connected to Modelica models ▪ Performance of inference algorithms ▪ Free, open source ▪ Web site including documentation, ready to use executable, source code: http://o3prm.lip6.fr | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Uses the Agrum open source library for inference | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ In a first step: researchers interested in creating diagnosis applications. Then the users of such applications in the industry. | |
| Provider: | <ul style="list-style-type: none"> ▪ Lip6 (Laboratoire d'informatique de Paris 6) and EDF | |
| Contact point: | <ul style="list-style-type: none"> ▪ Marc Bouissou (EDF R&D) | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ This software is currently under a GPL license | |

Latest update: 19/04/2016

| Name: SKELBO Figaro library | | |
|---|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Thermohydraulic system architecture | <ul style="list-style-type: none"> This library can be exploited by the Figaro processor in order to generate a fault tree describing the causes of a thermohydraulic system failure Describes failure modes (on demand and in function) of the most common thermohydraulic components, with the way they can propagate in a system Includes 29 classes of objects that can be used to describe a system This is a “static” library in the sense that the only possible processing is the generation of (static) fault trees When loaded in the KB3 tool, offers a user friendly graphical user interface | <ul style="list-style-type: none"> Fault tree(s) |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Probably the only library of that kind available worldwide Free | |
| Integration constraint(s): | <ul style="list-style-type: none"> Must be used with the Figaro processor The input must be a list of interconnected objects described in Figaro syntax | |
| Intended user(s): | <ul style="list-style-type: none"> Designers of thermohydraulic systems with high dependability stakes | |
| Provider: | <ul style="list-style-type: none"> EDF | |
| Contact point: | <ul style="list-style-type: none"> Marc Bouissou (EDF R&D) | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> The license is not yet determined precisely, but will be of type LGPL | |

Latest update: 18/04/2016

| Name: Figaro export in Dymola | | |
|--|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> A Modelica model, “enriched” with a few string parameters containing bits of Figaro code | <ul style="list-style-type: none"> Establishes a bridge between detailed simulation models built in Modelica and simplified discrete state (sometimes even Boolean) models suitable for dependability analysis | <ul style="list-style-type: none"> A list of Figaro objects describing the input system A Figaro behavioural model of the system (optional) A fault tree (optional) |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Associates two mature tools: Dymola for simulation, the Figaro tools for dependability analysis | |
| Integration constraint(s): | <ul style="list-style-type: none"> The use of this export function requires Dymola itself, the model management library for Dymola, the Figaro tools and a Figaro library suitable for the kind of system to be processed (see for example component SKELBO) | |
| Intended user(s): | <ul style="list-style-type: none"> Designers of systems with high dependability stakes | |
| Provider: | <ul style="list-style-type: none"> EDF | |
| Contact point: | <ul style="list-style-type: none"> Marc Bouissou (EDF R&D) | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Purchase of licenses for Dymola and the model management library Figaro tools are in a process towards open source; the license is not yet determined precisely, but will probably be of type LGPL | |
| <i>Latest update: 18/04/2016</i> | | |

| Name: Triphase | | |
|---|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Connection to a physical point of measurement | <ul style="list-style-type: none"> High-speed, synchronized, multi-node current and voltage sensor clusters. Supports several hundreds of measurement channels sampled at 100 kHz+. All channels are synchronized down to +/-10 nanoseconds. Measurement nodes are connected to and synchronized via a real-time optical network. Network cards and drivers for integration in Linux-based data servers. Support for continuous and event-based measurement processing | <ul style="list-style-type: none"> Synchronized current and voltage measurement streams |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Open-data architecture; users have access to raw data streams; full integration in to Linux and real-time Linux extensions (Xenomai). System has the ability to analyze incoming data streams in real-time, thereby enabling sensor use for smart fuses and protections. Closed-loop reaction times down to 50µs. | |
| Integration constraint(s): | <ul style="list-style-type: none"> Hardware installation required (sensors; field hubs for data acquisition; network card) Support for PCIe required Linux-based OS required | |
| Intended user(s): | <ul style="list-style-type: none"> Companies interested in smart monitoring and protections for electrical machines and power systems. This includes IoT and data-mining for automotive, aerospace and industrial applications | |
| Provider: | <ul style="list-style-type: none"> Triphase N.V. | |
| Contact point: | <ul style="list-style-type: none"> piet.vanassche@triphase.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Single cost for hardware. Yearly cost for support and software updates | |

Latest update: 20/04/2016

| Name: Simpack FMI 2.0 Import | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> FMI 2.0 FMUs | <ul style="list-style-type: none"> FMI 2.0 for Co-Simulation import FMI 2.0 for Model-Exchange import Full event handling support including discontinuous (multi-mode) systems | <ul style="list-style-type: none"> Coupled simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Available for all major platforms Full integration of the FMI 2.0 standard Seamless coupling between different FMUs and between the Simpack MBS solver | |
| Integration constraint(s): | <ul style="list-style-type: none"> Simpack FMI import license required External licenses may also be required dependent on the FMU | |
| Intended user(s): | <ul style="list-style-type: none"> Users interested in coupling MBS models with other domain (e.g. controllers) | |
| Provider: | <ul style="list-style-type: none"> Dassault Systems (Tool: Simpack) | |
| Contact point: | <ul style="list-style-type: none"> Remco.mansvelders@3ds.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: Simpack FMI export | | |
|--|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Simpack models | <ul style="list-style-type: none"> ▪ Converts ANY Simpack model into a Co-Simulation FMU ▪ The FMU is fully self-contained ▪ 3rd party sub-models are also exported: e.g. tire models ▪ Imported FMUs are also exported: the generated FMU contains then other FMUs hierarchically. | <ul style="list-style-type: none"> ▪ Co-Simulation FMU |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Available for all major platforms ▪ Export of ANY Simpack model possible ▪ Export of 3rd party sub-model ▪ Export of imported FMUs: hierarchical FMU export | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ On each computer the FMU must be run ONCE with administrator privileges to configure some OS details. ▪ FMU export license required ▪ When running the FMU all normal Simpack licenses are required. | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Users interested in coupling MBS models with other domain (e.g. controllers). FMI import is recommended. If this is not possible or undesired the FMU export approach is an alternative option. | |
| Provider: | <ul style="list-style-type: none"> ▪ Dassault Systems (Tool: Simpack) | |
| Contact point: | <ul style="list-style-type: none"> ▪ Remco.mansvelders@3ds.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Software license required | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: xMOD FMI 2.0 Import | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> FMI 2.0 FMUs | <ul style="list-style-type: none"> FMI 2.0 for Model-Exchange import FMI 2.0 for Co-Simulation import Multi-core, multi-rate, multi-solver co-simulation of FMUs with other models from major modeling and simulation tools | <ul style="list-style-type: none"> Intuitive and interactive co-simulation, with rich dashboards Co-simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> xMOD is a tool neutral integration environment Unique features to enable FMUs use by non-experts Hard-real time co-simulation under the RTX real-time operating system | |
| Integration constraint(s): | <ul style="list-style-type: none"> Windows 7 or later RTX if hard-real time execution is needed Other third-party licenses may also be required dependent on the FMU provider | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> IFP Energies nouvelles – D2T | |
| Contact point: | <ul style="list-style-type: none"> contact@xmodsoftware.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> xMOD Workshop + xMOD FMI licenses required | |

Latest update: 05/05/2016

| Name: xMOD advanced multi-core co-simulation methods | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ FMI 1.0 FMUs ▪ FMI 2.0 FMUs | <ul style="list-style-type: none"> ▪ Refined Scheduling Co-simulation method enabling the parallelization of FMUs execution, taking into account structural dependency information ▪ Context-based polynomial extrapolation improving the accuracy of data exchange ▪ Multi-core, multi-rate, multi-solver co-simulation of FMUs with other models from major modeling and simulation tools | <ul style="list-style-type: none"> ▪ Speed-up co-simulation execution (up to a factor of 10 in 16 cores observed) ▪ Intuitive and interactive co-simulation, with rich dashboards ▪ Co-simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ xMOD is a tool neutral integration environment ▪ Unique features to enable models use by non-experts ▪ Hard-real time co-simulation under the RTX real-time operating system | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Windows 7 or later ▪ RTX if hard-real time execution is needed ▪ Other third-party licenses may also be required dependent on the FMU provider | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ IFP Energies nouvelles – D2T | |
| Contact point: | <ul style="list-style-type: none"> ▪ contact@xmodsoftware.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ xMOD workshop license | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: IDA ICE for building operations | | |
|---|--|---|
| Input(s): | Main feature(s): | Output(s): |
| <ul style="list-style-type: none"> ▪ Measured signals from a building management system ▪ An IDA ICE simulation model of the building (normally a result from the design process) | <ul style="list-style-type: none"> ▪ The software module is designed to be permanently connected to a building in real time ▪ The simulation model is continuously adjusted by state estimation techniques to track the evolution of the building ▪ Automatic re-calibration will be regularly performed and the results of this calibration will provide early fault-detection and diagnostics | <ul style="list-style-type: none"> ▪ Parameter evolution as a diagnostic tool ▪ Virtual sensing from the simulator for enhanced control ▪ Enhanced manual fault diagnosis by a 3D presentation of the building |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ The only building monitoring tool that relies on a detailed simulation model ▪ Leverages the information collected by the numerous sensors that are installed in modern buildings ▪ Provides unique diagnostic capabilities | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Runs on a windows platform ▪ Reads and writes BMS signals from and to an OPC server | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Building continuous commissioning experts | |
| Provider: | <ul style="list-style-type: none"> ▪ EQUA Simulation AB | |
| Contact point: | <ul style="list-style-type: none"> ▪ Per Sahlin, EQUA Simulation AB, per.sahlin@equa.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Commercial licensing, rental | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: JModelica.org for on-line dynamic optimization | | |
|---|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ A Modelica model of the process to be controlled or monitored ▪ Objective function and optimization constraints ▪ Measurement data or equivalent, at all sampling times | <ul style="list-style-type: none"> ▪ Classes for Model Predictive Control and Moving Horizon Estimation ▪ Improved performance for real-time applications such as warm start capabilities and softening of optimization constraints ▪ Debugging features such as back tracking (from discretized problem to original, continuous) | <ul style="list-style-type: none"> ▪ For Model Predictive Control: optimal process inputs satisfying constraints, at sampling instants ▪ For Moving Horizon: estimate of the monitored variables, at sampling instants |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ JModelica.org with on-line optimizing controller (NMPC) features to solve a finite horizon optimal control problem. | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ JModelica.org is supported on Windows (7, Vista) and later and on Linux with 32-bit or 64-bit architecture | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ Modelon AB | |
| Contact point: | <ul style="list-style-type: none"> ▪ Johan Åkesson Modelon AB, johan.akesson@modelon.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Distributed under the GPL v.3 license approved by the Open Source Initiative. | |

Latest update: 05/05/2016

| Name: JModelica.org for off-line dynamic optimization | | |
|---|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ A Modelica model of the process to be calibrated ▪ Measurement data to be matched by the process model | <ul style="list-style-type: none"> ▪ A tool-box for grey box identification, including the ability to identify measurement noise intensity and statistical tests for overfitting. | <ul style="list-style-type: none"> ▪ Calibrated process model ▪ Characteristics of the measurement noise |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ JModelica.org for offline estimation of grey-box model parameters | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ JModelica.org is supported on Windows (7, Vista) and later and on Linux with 32-bit or 64-bit architecture | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ Modelon AB | |
| Contact point: | <ul style="list-style-type: none"> ▪ Johan Åkesson Modelon AB, johan.akesson@modelon.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Distributed under the GPL v.3 license approved by the Open Source Initiative. | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: DYMOLA FMI 2.0 Import | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ FMI 2.0 FMUs ▪ Modelica models | <ul style="list-style-type: none"> ▪ FMI 2.0 for Co-Simulation import ▪ FMI 2.0 for Model-Exchange import ▪ Full event handling support including support for multi-mode DAE systems | <ul style="list-style-type: none"> ▪ Coupled simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Available for all major platforms (win32, win64, linux32, linux64) ▪ Full integration of the FMI 2.0 standard ▪ First tool on market with FMI 2.0 support ▪ Seamless coupling between different FMUs and Modelica models | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Supported with the standard DYMOLA license ▪ External licenses may be required if required by the imported FMU | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ Dassault Systèmes (Tool: DYMOLA) | |
| Contact point: | <ul style="list-style-type: none"> ▪ Dan.HENRIKSSON@3ds.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Software license required | |
| <i>Latest update: 22/04/2016</i> | | |

| Name: DYMOLA FMI 2.0 Export | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica models | <ul style="list-style-type: none"> FMI 2.0 for Co-Simulation export FMI 2.0 for Model-Exchange export Full event handling support Export of Modelica models including resources | <ul style="list-style-type: none"> FMI 2.0 FMUs |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Available for all major platforms (win32, win64, linux32, linux64) Full support of the FMI 2.0 standard First tool on market with FMI 2.0 support Seamless coupling between different FMUs and Modelica models | |
| Integration constraint(s): | <ul style="list-style-type: none"> Supported with the standard DYMOLA license License-free FMUs generated with the Binary Model Export option, to simplify integration in other FMI environments | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> Dassault Systèmes (Tool: DYMOLA) | |
| Contact point: | <ul style="list-style-type: none"> Dan.HENRIKSSON@3ds.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required | |
| <i>Latest update: 22/04/2016</i> | | |

| Name: < Energy performance simulation tool for buildings > | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Building specification | <ul style="list-style-type: none"> ▪ Building library ▪ Improved building design ▪ Supervision Assistance ▪ Evaluation of the impact of sensors default on comfort and energy | <ul style="list-style-type: none"> ▪ Building performance analysis ▪ Improve optimization algorithm ▪ Building supervisor |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Energy performance simulation tool for buildings <ul style="list-style-type: none"> ○ Standard simulator (editable) ○ Analysis tools ○ Optimization algorithm multi-source multi consumers | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ need the platform modelica or Dymola | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ TRL 4 : need some work/collaboration for end users | |
| Provider: | <ul style="list-style-type: none"> ▪ Sherpa Engineering | |
| Contact point: | <ul style="list-style-type: none"> ▪ Philippe FIANI - p.fiani@sherpa-eng.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Licencing | |
| <i>Latest update: 05/05/2016</i> | | |

| Name: LMS Imagine.Lab FMI 2.0 Import & Export | | |
|--|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Amesim models ▪ Modelica models | <ul style="list-style-type: none"> ▪ FMI 2.0 for Co-Simulation export ▪ FMI 2.0 for Model-Exchange export | <ul style="list-style-type: none"> ▪ FMI 2.0 FMU |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Full support of the FMI 2.0 standard ▪ Validated on real cross-tools use cases ▪ Available on win32, win64, linux32, linux64 ▪ Seamless integration in Amesim, ease of use | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Supported with the standard LMS Imagine.Lab Amesim/Amerun licensing | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Plant modellers ▪ Integrators ▪ System engineers ▪ Control engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ Siemens PLM (Tool: LMS Imagine.Lab Amesim) | |
| Contact point: | <ul style="list-style-type: none"> • Pacome.magnin@siemens.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Software license required | |
| <i>Latest update: 22/04/2016</i> | | |

| Name: LMS Imagine.Lab MODELICA compiler | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica models | <ul style="list-style-type: none"> Authoring of MODELICA models MODELICA compilers | <ul style="list-style-type: none"> Simulators |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Fully integrated in LMS Imagine.Lab Amesim Combine MODELICA model and with legacy bond-graph models Improved behavior and performances Available on win32, win64, linux32, linux64 | |
| Integration constraint(s): | <ul style="list-style-type: none"> Supported with the standard LMS Imagine.Lab Amesim/Amerun licensing | |
| Intended user(s): | <ul style="list-style-type: none"> Plant modellers System engineers | |
| Provider: | <ul style="list-style-type: none"> Siemens PLM (Tool: LMS Imagine.Lab Amesim) | |
| Contact point: | <ul style="list-style-type: none"> Pacome.magnin@siemens.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required | |
| <i>Latest update: 22/04/2016</i> | | |

| Name: OPCClassic | | |
|--|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ OPC DA data ▪ Modelica models | <ul style="list-style-type: none"> ▪ Allows you to link your simulation models to the real world by connecting to OPC DA servers. | <ul style="list-style-type: none"> ▪ Simulation results ▪ OPC DA data |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Modelica library for accessing OPC DA servers. ▪ The library acts as an OPC client, enabling you to incorporate real, live data into your simulations. ▪ Suitable for real-time simulations. ▪ Supports reading OPC DA data. ▪ Supports writing OPCA DA data. | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Supported by Wolfram SystemModeler | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ Process industry and other heavy users of OPC | |
| Provider: | <ul style="list-style-type: none"> ▪ Wolfram MathCore | |
| Contact point: | <ul style="list-style-type: none"> ▪ Otto Tronarp ottot@wolfram.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Commercial license | |
| <i>Latest update: 25/04/2016</i> | | |

| Name: System Reliability in Wolfram SystemModeler | | |
|--|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica models with extra annotations for reliability | <ul style="list-style-type: none"> Use your Modelica model (annotated with reliability properties) for system reliability analysis. | <ul style="list-style-type: none"> Reliability results Reliability analysis |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Annotate your Modelica model with reliability properties directly in the GUI. Apply Wolfram Mathematica's rich set of powerful reliability analysis tools for full system reliability analysis. | |
| Integration constraint(s): | <ul style="list-style-type: none"> A Wolfram SystemModeler license is required. A Wolfram Mathematica license is required. | |
| Intended user(s): | <ul style="list-style-type: none"> Reliability engineers | |
| Provider: | <ul style="list-style-type: none"> Wolfram MathCore | |
| Contact point: | <ul style="list-style-type: none"> Otto Tronarp ottot@wolfram.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Commercial license | |
| <i>Latest update: 25/04/2016</i> | | |

| Name: SimulationX FMI 2.0 Import | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ FMI 2.0 FMUs | <ul style="list-style-type: none"> ▪ FMI 2.0 for Co-Simulation import ▪ FMI 2.0 for Model-Exchange import | <ul style="list-style-type: none"> ▪ Coupled simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Full support of the FMI 2.0 standard ▪ Seamless coupling of model exchange and co-simulation FMUs with SimulationX models ▪ Re-export of imported connected FMUs is possible | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ SimulationX license ▪ Runtime licenses for the imported FMU might be required (depends on the FMU) | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ ESI ITI GmbH, www.simulationx.com | |
| Contact point: | <ul style="list-style-type: none"> ▪ info@itisim.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Commercial software license required | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: SimulationX FMI 2.0 Export | | |
|---|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ SimulationX models ▪ Modelica models | <ul style="list-style-type: none"> ▪ Export of simulation models as FMU for Co-Simulation or Model Exchange | <ul style="list-style-type: none"> ▪ FMUs for Co-Simulation ▪ FMUs for Model Exchange |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Full support of the FMI 2.0 standard ▪ The generated FMUs are stand-alone FMUs. Neither a SimulationX installation nor a special run-time license is required for execution of generated FMUs. ▪ Co-simulation FMUs can be exported with a variable or a fixed step solver. FMUs with fixed step solver can be used for real-time applications like Hardware-in-the-Loop simulation. ▪ Source code of FMU can be included. This allows for compilation on different target platforms. ▪ User can select which inputs, outputs and parameters are provided by the FMU. Internal (local) variables can be hidden. ▪ Re-export of imported connected FMUs is possible. ▪ External data files are automatically included within the FMU. | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ SimulationX license required for FMU generation | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ Test engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ ESI ITI GmbH, www.simulationx.com | |
| Contact point: | <ul style="list-style-type: none"> ▪ info@itisim.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Commercial software license required | |

Latest update: 27/04/2016

| Name: SimulationX Fault Tree Analysis Module | | |
|---|---|---|
| Input(s): | Main feature(s): | Output(s): |
| <ul style="list-style-type: none"> ▪ SimulationX models ▪ Modelica models | <ul style="list-style-type: none"> ▪ Reliability analysis using Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis (FMEA) based on Modelica models ▪ System simulation and reliability analysis on the same platform | <ul style="list-style-type: none"> ▪ Fault Trees ▪ Minimal Cut Sets ▪ FMEA Table |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ System simulation and reliability analysis can be performed using the same models and modeling platform ▪ HiP-HOPS developed at the University of Hull is used as the analysis backend for FTA and FMEA ▪ Ability to develop reusable model components which are augmented with failure events and specific failure behavior ▪ Failure propagation between model components can be defined on the connection level ▪ Scripting support for reliability analysis allows the implementation of user-defined workflows | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ SimulationX license with FTA/FMEA module required | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Reliability engineers ▪ System engineers in safety-critical domains ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ ESI ITI GmbH, www.simulationx.com | |
| Contact point: | <ul style="list-style-type: none"> ▪ info@itisim.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Commercial software license required | |

Latest update: 27/04/2016

| Name: Functional Mock-up Interface (FMI) Version 2.0 | | |
|---|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Simulation models | <ul style="list-style-type: none"> Standard for exchange of simulation models and co-simulation between different modelling, simulation and co-simulation tools | <ul style="list-style-type: none"> Functional Mock-up Units (FMUs) |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Tool independent standard Standard was developed and is maintained as a joint effort by the Modelica Association Project FMI Supported by more than 60 simulation, integration or test tools Further development is organized in FMI Working Groups Development process follows certain process rules A free FMU Compliance Checker is available | |
| Integration constraint(s): | <ul style="list-style-type: none"> The Modelica Association FMI provides the FMI specification only For generation of FMUs, software tools are necessary | |
| Intended user(s): | <ul style="list-style-type: none"> Simulation tool developers Software developers | |
| Provider: | <ul style="list-style-type: none"> Modelica Association Project FMI, www.fmi-standard.org | |
| Contact point: | <ul style="list-style-type: none"> contact@fmi-standard.org | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Specification is available free of charge under the CC-BY-SA (CreativeCommons Attribution-Sharealike 4.0 International) license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: Sundials/ML Library | | |
|--|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> OCaml programs | <ul style="list-style-type: none"> Comprehensive and type-safe OCaml interface to the Sundials suite of numeric solvers. | <ul style="list-style-type: none"> Executables that link with the Sundials binaries. |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> A well-documented interface for the high-level OCaml programming language to the widely used Sundials suite of numeric solvers. Ideal for implementing algorithms that mix symbolic manipulation and numeric code. Static and dynamic checks clarify library use and prevent programming errors. Thorough benchmarking against all of the standard Sundials examples demonstrates that the library is robust and efficient. Free, open source (BSD license). Web site including documentation: http://inria-parkas.github.io/sundialsml/. | |
| Integration constraint(s): | <ul style="list-style-type: none"> Uses the Sundials suite of numeric solvers. | |
| Intended user(s): | <ul style="list-style-type: none"> Researchers and engineers prototyping algorithms or applications that combine symbolic manipulation and numeric simulation (e.g., compilers and interpreters for hybrid modeling languages). | |
| Provider: | <ul style="list-style-type: none"> Inria | |
| Contact point: | <ul style="list-style-type: none"> Timothy Bourke (Inria Paris) | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> BSD License | |

Latest update: 19/04/2016

| Name: PySimulator FMI 2.0 Import | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ FMI 2.0 for Model Exchange FMU ▪ FMI 2.0 for Co-Simulation FMU | <ul style="list-style-type: none"> ▪ Displaying all XML-meta information of the imported FMU (FMI version 2.0) ▪ Numerical integration of FMI 2.0 for Model Exchange FMU ▪ Master algorithm for a single FMI 2.0 for Co-Simulation FMU | <ul style="list-style-type: none"> ▪ Simulation results and plots |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Innovative plugin concept to include own features in Python ▪ Open Source Simulator without costs | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Python installation suited for PySimulator ▪ Windows platform, on Linux limited | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ Research engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ DLR Institute of System Dynamics and Control (download: https://github.com/PySimulator) | |
| Contact point: | <ul style="list-style-type: none"> ▪ Andreas Pfeiffer Andreas.Pfeiffer@dlr.de | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ LGPL version 3 | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: Nonlinear Kalman Filter Modelica Library | | |
|--|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica continuous-time plant model | <ul style="list-style-type: none"> GUI supported fully automatic generation of nonlinear Modelica Kalman filter models Several filter algorithms: EKF, UKF, moving horizon | <ul style="list-style-type: none"> Tailored Kalman filter models in Modelica for the given plant model |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Support of fully nonlinear plant models to be used in Kalman filters Extension available for Kalman filter models on real time systems | |
| Integration constraint(s): | <ul style="list-style-type: none"> Dymola license Modelica model for the plant | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers Research engineers | |
| Provider: | <ul style="list-style-type: none"> DLR Institute of System Dynamics and Control | |
| Contact point: | <ul style="list-style-type: none"> Jonathan Brembeck Jonathan.Brembeck@dlr.de | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Commercial license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: Modelica_Requirements Library | | |
|--|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Requirements | <ul style="list-style-type: none"> ▪ Defining requirements in a convenient way by “Drag & Drop” with a GUI. ▪ Features <ul style="list-style-type: none"> ○ time locators (when to check) ○ check in fixed/sliding time windows ○ check based on FFTs ○ check based on 2D domain | <ul style="list-style-type: none"> ▪ Requirements are automatically checked in every simulation run of a Modelica model. |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Modelica implementation of the FORM-L basics to formally define requirements ▪ Automatic checking of formally defined requirements whenever a Modelica model is simulated | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Modelica environment (checked with Dymola, OpenModelica, SimulationX) | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ Research engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ DLR Institute of System Dynamics and Control, EDF, Dassault Aviation UNICAL (Version 0.6 in MODRIO deliverable D2.1.1. Version 1.0 planned to be available at https://github.com/modelica) | |
| Contact point: | <ul style="list-style-type: none"> ▪ Martin Otter Martin.Otter@dlr.de | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Open Source License (Modelica License 2) | |

Latest update: 27/04/2016

| Name: OpenModelica for reliability verification | | |
|--|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Modelica models with safety information ▪ Modelica models with bindings | <ul style="list-style-type: none"> ▪ Support for requirement binding and verification scenario generation ▪ Support for export of Figaro models | <ul style="list-style-type: none"> ▪ Reliability analysis ▪ Requirement verification |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Supports reliability verification ▪ Supports requirement verification | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Depends on Figaro processor for reliability analysis | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ System engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ Linköping University | |
| Contact point: | <ul style="list-style-type: none"> ▪ Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ OSMC-PL license ▪ Figaro tools are in a process towards open source; the license is not yet determined precisely, but will probably be of type LGPL | |
| <i>Latest update: 19/04/2016</i> | | |

| Name: OpenModelica for simulation | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica models | <ul style="list-style-type: none"> Support for Modelica language features, particularly clocked synchronous language elements and built-in state machines Support for Modelica Standard Library 3.2.2 | <ul style="list-style-type: none"> Simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Comprehensive modeling, simulation and systems engineering environment Open-source for both industrial and academic usage | |
| Integration constraint(s): | <ul style="list-style-type: none"> ... | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> Linköping University | |
| Contact point: | <ul style="list-style-type: none"> Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> OSMC-PL license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: OpenModelica FMI 2.0 Import | | |
|--|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> FMI 2.0 FMUs | <ul style="list-style-type: none"> Full support for FMI 2.0 for Model-Exchange import | <ul style="list-style-type: none"> Simulation results |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Imports the FMU as wrapper Modelica model so can be used with difference FMUs and Modelica models Available for all major platforms (win32, win64, linux32, linux64) Full integration of the FMI 2.0 standard | |
| Integration constraint(s): | <ul style="list-style-type: none"> External licenses may be required if required by the imported FMU | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> Linköping University | |
| Contact point: | <ul style="list-style-type: none"> Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> OSMC-PL license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: OpenModelica FMI 2.0 Export | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Modelica models | <ul style="list-style-type: none"> Full support for FMI 2.0 for Co-Simulation export Full support for FMI 2.0 for Model-Exchange export | <ul style="list-style-type: none"> FMI 2.0 FMUs |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Available for all major platforms (win32, win64, linux32, linux64) Cross-compilation for different platforms Full support of the FMI 2.0 standard | |
| Integration constraint(s): | <ul style="list-style-type: none"> License-free FMUs generated with the binaries and source code, to simplify integration in other FMI environments | |
| Intended user(s): | <ul style="list-style-type: none"> Control engineers System engineers System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> Linköping University | |
| Contact point: | <ul style="list-style-type: none"> Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> OSMC-PL license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: OpenModelica for dynamic optimization | | |
|---|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Modelica process models ▪ Objective function and optimization constraints specified using Modelica annotations | <ul style="list-style-type: none"> ▪ Solve classes of nonlinear optimal control problems (NOCP) with fixed time horizons | <ul style="list-style-type: none"> ▪ Optimal process inputs and simulation result for optimal inputs |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Built-in optimization support leveraging OpenModelica's symbolic processing engine for preprocessing the NOCP problem ▪ Smoothly integrated in the OpenModelica modeling, simulation and systems engineering environment ▪ Open-source for both industrial and academic usage | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ ... | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ Linköping University | |
| Contact point: | <ul style="list-style-type: none"> ▪ Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ OSMC-PL license | |
| <i>Latest update: 27/04/2016</i> | | |

| Name: OpenModelica Performance Analyzer | | |
|---|---|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Modelica models with performance issues ▪ Modelica models with real-time constraints | <ul style="list-style-type: none"> ▪ Tells the user which equations cause slowness in model simulation ▪ Allows the user to compare how different design choices influence performance ▪ The ability to choose if you want to profile linear/non-linear systems, functions, or all equations | <ul style="list-style-type: none"> ▪ A sortable and user-friendly overview of the equations in the model showing their absolute and relative cost of the simulation time |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Supports the engineer in understanding the performance impact of equations in models ▪ Supports the engineer in designing models that simulate quickly | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ The graphical user interface depends on simulation with OMEdit ▪ The full data is stored in a simple binary format as well as a JSON file containing summary information | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Control engineers ▪ System engineers ▪ System simulation end-users | |
| Provider: | <ul style="list-style-type: none"> ▪ Linköping University | |
| Contact point: | <ul style="list-style-type: none"> ▪ Peter Fritzson peter.fritzson@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ The output of the simulation executable is free to use without conditions, as is running OMEdit ▪ The run-time of the simulation executable generating the profiling information is the OSMC-PL run-time license (BSD 3-clause) ▪ The code generator and OMEdit are licensed under the OSMC-PL license (GPLv3 with a commercial license option for OSMC members). | |

Latest update: 27/04/2016

| Name: OpenModelica Debugger | | |
|---|--|---|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Modelica models showing some error during run-time (solvers not converging, constraints violated, and more) ▪ Modelica models with unexpected result for algorithmic code or functions ▪ Any Modelica model | <ul style="list-style-type: none"> ▪ Tells the user what the compiler did to each equation in order to end up with the simulated system ▪ Shows which equations are part of any block of equations ▪ Shows the relation between variables and equations, including dependencies | <ul style="list-style-type: none"> ▪ A view in the OMEdit user interface showing the debugging information to the user |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Supports the engineer in understanding the simulated model ▪ Helps finding the source(s) of bugs in the model | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ The graphical user interface depends on simulation with OMEdit ▪ The full data is stored in a simple binary format as well as a JSON file containing summary information | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ System engineers | |
| Provider: | <ul style="list-style-type: none"> ▪ Linköping University | |
| Contact point: | <ul style="list-style-type: none"> ▪ Peter Fritzson peter.fritzson@liu.se ▪ Martin Sjölund martin.sjolund@liu.se | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ The output of the simulation executable is free to use without conditions, as is running OMEdit ▪ The run-time of the simulation executable generating the profiling information is the OSMC-PL run-time license (BSD 3-clause) ▪ The code generator and OMEdit are licensed under the OSMC-PL license (GPLv3 with a commercial license option for OSMC members) | |

Latest update: 27/04/2016

| Name: LMS Imagine.Lab MPC Prototype | | |
|--|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> Imagine.Lab model to be controlled | <ul style="list-style-type: none"> Quick prototyping of a model-based, constrained MIMO controller | <ul style="list-style-type: none"> Closed-loop simulation of controller+model |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Fully integrated in LMS Imagine.Lab Amesim State-of the art optimization routines Automated control model generation through linearization of provided non-linear Imagine.Lab model | |
| Integration constraint(s): | <ul style="list-style-type: none"> Standard LMS Imagine.Lab Amesim/Amerun licensing Prototype blocks not included in standard distribution | |
| Intended user(s): | <ul style="list-style-type: none"> Plant modellers faced with evaluating closed-loop system performance System engineers Control engineers | |
| Provider: | <ul style="list-style-type: none"> Siemens PLM (Tool: LMS Imagine.Lab Amesim) | |
| Contact point: | <ul style="list-style-type: none"> bert.pluymers@siemens.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required Prototype software not included in standard distribution | |
| <i>Latest update: 04/05/2016</i> | | |

| Name: Model Based State-Estimation Algorithms | | |
|--|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> ▪ Lumped parameter model: support for bond-graph models ▪ Structural finite element models - system matrices ▪ Sensor output(s) for the modeled system | <ul style="list-style-type: none"> ▪ Compares the model prediction with the measurements for estimation of states/inputs/parameters ▪ Different estimator: Linear and nonlinear Kalman filters, Moving Horizon Estimation | <ul style="list-style-type: none"> ▪ Estimated model states ▪ Estimated model inputs ▪ Estimated model parameters |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> ▪ Consistent approach to exploit design models in operational setting ▪ Applicable for detailed component models and high-level system models | |
| Integration constraint(s): | <ul style="list-style-type: none"> ▪ Finite elements models should be exportable to Matlab ▪ Integration with lumped parameter models is currently focused on Amesim | |
| Intended user(s): | <ul style="list-style-type: none"> ▪ Measurement engineers ▪ Condition monitoring | |
| Provider: | <ul style="list-style-type: none"> ▪ KU Leuven | |
| Contact point: | <ul style="list-style-type: none"> ▪ Wim Desmet wim.desmet@kuleuven.be ▪ Bert Pluymers bert.pluymers@kuleuven.be | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> ▪ Collaborative or bilateral research project | |

Latest update: 03/05/2016

| Name: LMS Virtual.Lab Motion – FMI 2.0 Co-Simulation export | | |
|---|--|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> LMS Virtual.Lab Motion models | <ul style="list-style-type: none"> Export LMS Virtual.Lab Motion model into a Co-Simulation FMU The FMU is compiled for win32 and win64 and dynamically links to the Virtual.Lab Motion solver. | <ul style="list-style-type: none"> FMI 2.0 Co-Simulation export |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Allows the export of flexible multibody systems models in standard FMU for co-simulation Re-initialization of the equations of motion Linearization and extraction of a reduced model (ODEs from DAEs) | |
| Integration constraint(s): | <ul style="list-style-type: none"> Supported on an R&D version of LMS Virtual.Lab Motion Functionality activation key required | |
| Intended user(s): | <ul style="list-style-type: none"> Users interested in making use of MBS models in mechatronic context System designers and engineers Control engineers | |
| Provider: | <ul style="list-style-type: none"> Siemens PLM (Tool: LMS Virtual.Lab Motion) | |
| Contact point: | <ul style="list-style-type: none"> Christophe.liefoghe@siemens.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required Functionality activation key required | |
| <i>Latest update: 02/05/2016</i> | | |

| Name: LMS Virtual.Lab Motion – Real time parallel solver | | |
|---|---|--|
| Input(s): | Main feature(s) | Output(s): |
| <ul style="list-style-type: none"> LMS Virtual.Lab Motion models | <ul style="list-style-type: none"> Deterministic computation of the multibody systems models. | <ul style="list-style-type: none"> Binary / ASCII Results file readable into the LMS Virtual.Lab Motion GUI |
| Unique Selling Proposition(s): | <ul style="list-style-type: none"> Portability to several Real-Time Operating Systems for online implementation of systems validation and state estimation with complex models Efficient implementation of a parallel solution of the multibody system Capability to handle complex mechanism models (e.g. > 150 DOFs). | |
| Integration constraint(s): | <ul style="list-style-type: none"> Export of RT solver input file from standard solver License required | |
| Intended user(s): | <ul style="list-style-type: none"> System designers and engineers Control and mechatronic engineers Sil, HIL and HiTL engineers | |
| Provider: | <ul style="list-style-type: none"> Siemens PLM (Tool: LMS Virtual.Lab Motion) | |
| Contact point: | <ul style="list-style-type: none"> Christophe.liefooghe@siemens.com | |
| Condition(s) for reuse: | <ul style="list-style-type: none"> Software license required | |
| <i>Latest update: 02/05/2016</i> | | |