ITEA 3

ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

Exploitable Results by Third Parties

15016 EMPHYSIS

Project details

Project leader:	Oliver Lenord
Email:	Oliver.lenord@de.bosch.com
Website:	https://emphysis.github.io/

Name: eFMI Support in DYMOLA and 3DEXPERIENCE / DBM App

Input(s):	Main feature(s)	Output(s):
- Model - EqCode eFMU (after project end)	- Supporting eFMI workflow - Advanced solvers with real-time capabilities - Integration with other DS solutions for certified production code generation	- AlgCode eFMU - ProdCode eFMU (via CATIA ESP app) - EqCode eFMU (after project end)

Unique Selling Proposition(s):	- Complete support for the Modelica language and Modelica Standard Library - Fast time-to-market of EMPHYSIS project results - Access to big customer installed base through integration in CATIA on 3DEXPERIENCE Platform
Integration constraint(s):	- Currently only supporting Modelica models as input - EqCode eFMUs will also be supported when that specification is available (after end of EMPHYSIS)
Intended user(s):	- Customers of the Dassault Systèmes Modelica solutions (3DEXPERIENCE and DYMOLA)
Provider:	- Dassault Systèmes
Contact point:	- Dan Henriksson, dan.henriksson@3ds.com
Condition(s) for reuse:	- License for export needed for eFMU generation - No license required for later reuse of generated eFMUs

Latest update: 2021-01-15

Name: eFMI Support in SimulationX

Input(s):	Main feature(s)	Output(s):
▪ Model	▪ Generation of an eFMU containing GALEC code (AlgCode) from clocked partition of the model. ▪ Providing an user interface to parametrize the code generation process	▪ eFMU (AlgCode) ▪ eFMU(EqCode) -> after the project

Unique Selling Proposition(s):	▪ Complete support of the Modelica Language, Modelica Standard Library, SimulationX specific and third-party libraries ▪ Fast time-to-market of EMPHYSIS project results ▪ Big customer base of users of SimulationX and other ESI-tools ▪ Enables code generation directly from Modelica model to ECU.
Integration constraint(s):	▪ The clocked model partitions of a Modelica model are transformed into GALEC (AlgCode). The modeler has to use the synchronous extension of Modelica, which are the right means for controller modelling. ▪ SimulationX as a modelling tool is the entry point of the tool chain, other tools for the back end are necessary.
Intended user(s):	▪ SimulationX users who need to generate code for ECUs
Provider:	▪ ESI ITI GmbH
Contact point:	▪ Gerd Kurzbach, Gerd.Kurzbach@esi-group.com
Condition(s) for reuse:	▪ License of this code export option required ▪ No license required for further use of the generated eFMU

Latest update: 2021-01-26

<table>
<tr><th colspan="3">Name: eFMI Compliance Checker</th></tr>
<tr><td>Input(s):</td><td>Main feature(s)</td><td>Output(s):</td></tr>
<tr><td>▪ eFMU archive file</td><td>▪ Validating the archive architecture according to the specified container architecture
▪ Consistency checking of included model representations
▪ Checking the Compliance of the GALEC code (AlgCode) against the rules of the eFMI specification</td><td>▪ Compliance report which lists all inconsistencies and non-conformities with rules of the eFMI specification</td></tr>
<tr><td>Unique Selling Proposition(s):</td><td colspan="2">▪ Fast reporting of any anomalies in the container architecture or GALEC code files
▪ Can be easily updated and extended
▪ This tool has the potential to become the official eFMI compliance checker</td></tr>
<tr><td>Integration constraint(s):</td><td colspan="2">▪ It is written in Python 3.8 so it requires the proper python installation
▪ Required Python modules: lxml.etree, hashlib, lark, colorama, collections.namedtuple and shutil
▪ Can be found in the repository: https://gitlab.ida.liu.se/emphysis/eFMU_ComplianceChecker</td></tr>
<tr><td>Intended user(s):</td><td colspan="2">▪ eFMI users and tool vendors</td></tr>
<tr><td>Provider:</td><td colspan="2">▪ ESI ITI GmbH</td></tr>
<tr><td>Contact point:</td><td colspan="2">▪ Khaled Alekeish, khaled.alekeish@esi-group.com</td></tr>
<tr><td>Condition(s) for reuse:</td><td colspan="2">▪ Open source
▪ License: GPLv3 license</td></tr>
</table>

Latest update: 2021-01-26

<table>
<tr><th colspan="3">Name: eFMU Front End for Astrée</th></tr>
<tr><td>Input(s):</td><td>Main feature(s)</td><td>Output(s):</td></tr>
<tr><td>ProductionCode eFMU</td><td>▪ The eFMU front end reads an eFMU with C production code, sets up an Astrée project and starts Astrée
▪ Astrée automatically checks for runtime errors and violations of coding rules in C applications</td><td>List of runtime errors and rule violations, or statement that no such problems exist</td></tr>
<tr><td>Unique Selling Proposition(s):</td><td colspan="2">▪ The eFMU front end allows for the automatic generation of an Astrée project from an eFMU.
▪ Astrée is a static code analyzer that finds runtime errors and rule violations in safety-critical software written or generated in C.
▪ Astrée is sound - that is, if no errors are signaled, the absence of errors has been proved.
▪ Astrée offers powerful annotation mechanisms for supplying external knowledge and fine-tuning the analysis precision for individual loops or data structures.
▪ The integrated RuleChecker can be configured to check for compliance with MISRA, CWE, ISO/IEC, and SEI CERT C coding rules.</td></tr>
<tr><td>Integration constraint(s):</td><td colspan="2">The eFMU front end requires Python 3.
Astrée requirements:
▪ Windows: 64-bit Windows 7 SP1 or newer
▪ Linux: 64-bit CentOS/RHEL 7 or compatible
▪ 4 GB of RAM (16 GB recommended)
▪ 4 GB of disk space</td></tr>
<tr><td>Intended user(s):</td><td colspan="2">▪ Developers using the eFMI workflow</td></tr>
<tr><td>Provider:</td><td colspan="2">▪ AbsInt Angewandte Informatik GmbH</td></tr>
<tr><td>Contact point:</td><td colspan="2">▪ support@absint.com</td></tr>
<tr><td>Condition(s) for reuse:</td><td colspan="2">▪ AbsInt offers commercial licenses, including training, support, and maintenance.</td></tr>
<tr><td></td><td colspan="2"><i>Latest update: 2021-01-19</i></td></tr>
</table>

Name: eFMI support in CATIA - ESP App		
Input(s):	Main feature(s)	Output(s):
▪ AlgCode eFMU ▪	▪ Supporting eFMI workflow ▪ Transformation of model for embedded	▪ ProdCode eFMU ▪ BinaryCode eFMU

Unique Selling Proposition(s):	▪ Multiple sources & multiple targets embedded C code generation and compilation toolset ▪ Access to big customer installed base through integration in CATIA on 3DEXPERIENCE Platform ▪ Generated code is OS neutral
Integration constraint(s):	▪ App called by other CATIA apps, eg. 3DEXP DBM, CATIA Magic ▪
Intended user(s):	▪ Customers of the Dassault Systèmes solutions for embedded systems (3DEXPERIENCE)
Provider:	▪ Dassault Systèmes
Contact point:	▪ Eric Mével, eric.mevel@3ds.com
Condition(s) for reuse:	▪ License for export needed for eFMU generation ▪ No license required for later reuse of generated eFMUs
	Latest update: 2021-01-21

Name: eFMI support in CATIA - AUTOSAR Builder App

Input(s):	Main feature(s)	Output(s):
▪ ProdCode eFMU ▪ BinaryCode eFMU	▪ Supporting eFMI workflow ▪ Transformation of model for AUTOSAR integration	▪ ProdCode eFMU ▪ BinaryCode eFMU ▪ AUTOSAR SW components ▪ AUTOSAR adaptive application

Unique Selling Proposition(s):	▪ Support of both classic and adaptive AUTOSAR standards ▪ Access to big customer installed base through integration in CATIA on 3DEXPERIENCE Platform
Integration constraint(s):	▪ ▪
Intended user(s):	▪ Customers of the Dassault Systèmes solutions for automotive embedded systems
Provider:	▪ Dassault Systèmes
Contact point:	▪ Eric Mével, eric.mevel@3ds.com
Condition(s) for reuse:	▪ License for export needed for eFMU generation & AUTOSAR support ▪ No license required for later reuse of generated eFMUs
	Latest update: 2021-01-21

Name: eFMI support for Capital Software Designer		
Input(s):	Main feature(s)	Output(s):
- ProdCode eFMU -	- Allow to import an eFMU into a larger software architecture - Validate with code check and open and closed loop testing of the code.	- Errors in the code, - Errors in the interface of the component - Behavior within the full software stack.

Unique Selling Proposition(s):	- Support of AUTOSAR and non-AUTOSAR workflows. - Integrated with ALM solution to trace the full lifecycle from model to code.
Integration constraint(s):	- -
Intended user(s):	- Software integrators and testers.
Provider:	- Siemens NV
Contact point:	- Jan Richter. janrichter@siemens.com
Condition(s) for reuse:	- Siemens offers commercial licenses, including training, support, and maintenance.
	Latest update: 2021-01-22

Name: dSPACE eFMU Container Manager		
Input(s):	Main feature(s)	Output(s):
▪ eFMU Container ▪ model representations	▪ create new container ▪ add, delete, modify model representations in container ▪ checksum calculation and handling	▪ modified eFMU container ▪
Unique Selling Proposition(s):	▪ consistent handling of eFMI containers ▪ command line tool	
Integration constraint(s):	▪ tool based on .Net Framework 4.8	
Intended user(s):	▪ model developers ▪ software developers ▪ software integrators	
Provider:	▪ dSPACE GmbH, Paderborn	
Contact point:	▪ Jörg Niere, jniere@dspace.de	
Condition(s) for reuse:	▪ container manager usage requires license ▪ no license required for later reuse of generated/modified eFMUs	

Latest update: 2021-01-22

Name: dSPACE eFMI Toolset for TargetLink		
Input(s):	Main feature(s)	Output(s):
▪ eFMU container with algorithm code model representation	▪ consistency checks code <-> manifest ▪ syntactical and semantical GALEC code checks ▪ strict data type analysis	▪ eFMU container from input enhanced with production code model representation

Unique Selling Proposition(s):	▪ seamless integration of Modelica models or Amesim models based on GALEC code ▪ full range of TargetLink features usable ▪ generate legacy production code or AUTOSAR code ▪ seamless integration in existing dSPACE tool chains
Integration constraint(s):	▪ TargetLink version 5.2 ▪ .Net Framework 4.8
Intended user(s):	▪ software developer ▪ software integrator
Provider:	▪ dSPACE GmbH, Paderborn
Contact point:	▪ Jörg Niere, jniere@dspace.de
Condition(s) for reuse:	▪ TargetLink Base license ▪ separate license required
	Latest update: 2021-01-22

Name: Tire Model Converter from MF-Tire to TMeasy		
Input(s):	Main feature(s)	Output(s):
▪ MF-Tire parameter set based on tire measurement fitting	▪ Transfer of the parameter sets from a phenomenological tire modeling (MFTire) to a physically motivated modeling of the tire behavior (TMeasy).	▪ parameter set for TMeasy tire model ▪ validation plots

Unique Selling Proposition(s):	▪ Direct transfer between two independent modeling approaches. ▪ The extensive and very complex tire measurements do not have to be available.
Integration constraint(s):	▪ Matlab is required for use
Intended user(s):	▪ Cooperation partner DLR in work package Validation and Demonstrator
Provider:	▪ EFS
Contact point:	▪ Paul Spannaus, paul.spannaus@efs-auto.com
Condition(s) for reuse:	▪ MF-Tire data record must be available and a cooperative agreement with EFS
	Latest update: 22.02.2021

Name: Test automation on SiL and HiL CarMaker-based Simulation		
Input(s):	Main feature(s)	Output(s):
▪ Maneuver description for test implementation in CarMaker	▪ Automated test catalog for the validation of the Modelica vehicle dynamics implementation for referencing the reference simulation (Ground Truth)	▪ vehicle movement ▪ validation Plots ▪ test results

Unique Selling Proposition(s):	▪ Test catalog for SiL and HiL environment in the CarMaker and EXAM tool chain for the validation of the Modellica driving dynamics implementation with consideration of active chassis components (active damper control on the HiL test bench)
Integration constraint(s):	▪ SiL test catalog requires CarMaker environment ▪ HiL environment requires EFS test benches with active chassis components and EXAM test automation
Intended user(s):	▪ EFS-GV
Provider:	▪ EFS
Contact point:	▪ Paul Spannaus, paul.spannaus@efs-auto.com
Condition(s) for reuse:	▪ Supplementary cooperation with EFS, as specific hardware has to be set up and maintained in order to be able to use active chassis systems
	Latest update: 22.02.2021

Name: eFMI Support in OpenModelica		
Input(s):	Main feature(s)	Output(s):
▪ Modelica code	▪ Flat Modelica export	▪ Flat Modelica

Unique Selling Proposition(s):	▪ Ability to use OpenModelica as frontend only, outputting intermediate code suitable for integration in other Modelica or non-Modelica tools.
Integration constraint(s):	▪ Flat Modelica not yet standardized. Output may change.
Intended user(s):	▪ Modelica users ▪ People who want to interface with Modelica code
Provider:	▪ Open Source Modelica Consortium
Contact point:	▪ https://openmodelica.org
Condition(s) for reuse:	▪ Open source
	Latest update: 2021-01-26

Name: eFMI Plugin for TPT		
Input(s):	Main feature(s)	Output(s):
▪ ProdCode eFMU ▪ eFMI Behavioral Model	▪ Automatic import, generation and configuration of a test environment ▪ Automatic generation of test cases and test oracles from Behavioral Model.	▪ Test Report ▪ Test Data

Unique Selling Proposition(s):	▪ Automatic import of the eFMI Component interface into the test development environment. ▪ Automatic generation and compilation of a test harness for general purpose code to enable SiL testing. ▪ Automatic back-to-back testing of use cases provided reference data w.r.t. tolerance, value bounds and Error signals. ▪ Reuse of imported scenarios for Back-2-Back testing against other execution platforms in TPT (e.g. FMI vs. eFMI Production code, eFMI Production vs. PiL/HiL)
Integration constraint(s):	▪ Basic TPT installation ▪ Windows Operation System
Intended user(s):	▪ Developers and Test Engineers in the eFMI Workflow
Provider:	▪ PikeTec GmbH
Contact point:	▪ support@piketec.com
Condition(s) for reuse:	▪ Commercial License needed for import and testing
	Latest update: 26.01.2021

Name: eFMI support in QuaRTOS-DSE – AlgCode2quartos traductor		
Input(s):	Main feature(s)	Output(s):
▪ AlgCode eFMU	▪ Supporting eFMI workflow ▪ Transformation of model for embedded software architecture design space exploration	▪ quartos model ▪ software architecture of AlgCode eFMU model

Unique Selling Proposition(s):	▪ Access to QuaRTOS-DSE: a modular framework for embedded SW architecture design space exploration and evaluation ▪ Based on AlgCode eBNF facilitates the integration of new AlgCode update
Integration constraint(s):	▪ ▪
Intended user(s):	▪ Tool provider solutions for embedded systems ▪ SW developer /SW integrator
Provider:	▪ CEA
Contact point:	▪ Belgacem BEN HEDIA: belagcem.ben-hedia@cea.fr
Condition(s) for reuse:	▪ CEA license needed for test and use
	Latest update: 2021-01-26

Name: eFMI support in QuaRTOS-DSE		
Input(s):	Main feature(s)	Output(s):
▪ AlgCode eFMU Transformed using AlgCode2quartos	▪ Supporting eFMI workflow ▪ modular and expandable according domain	▪ software architecture of AlgCode eFMU model ▪ AlgCode eFMU (wip)
Unique Selling Proposition(s):	▪ Access to QuaRTOS-DSE: a modular framework for embedded SW architecture design space exploration and evaluation. ▪ Expandable with a provided API: easily integration of domain constraint (adding new embedded SW architecture strategies, evaluation criteria, connection with another evaluation and analysis tool)	
Integration constraint(s):	▪ ▪	
Intended user(s):	▪ Tool provider solutions for embedded systems ▪ SW developer /SW integrator	
Provider:	▪ CEA	
Contact point:	▪ Belgacem BEN HEDIA: belagcem.ben-hedia@cea.fr	
Condition(s) for reuse:	▪ CEA license needed for test and use	

Latest update: 2021-01-26

Name: ETAS SCODE-CONGRA eFMI Toolbox		
Input(s):	Main feature(s)	Output(s):
▪ eFMU container with algorithm code model representation	▪ IDE for GALEC language including editor, syntax highlighting and editing support ▪ Syntax and semantic checking (e.g. dimension, type checking) of Algorithm code container, also against the manifest ▪ Generation of production code for various SW architectures as a production code container into the eFMU ▪ Generation of GALEC code from SCODE-CONGRA models	▪ eFMU container from input enhanced with production code model representation for selected software architectures

Unique Selling Proposition(s):	▪ Integration with other eFMU producing tools (e.g. Modelica tools, Amesim) ▪ Support of programming directly in the GALEC language. ▪ Range of target software architectures (OS neutral, AUTOSAR, …)
Integration constraint(s):	▪ ETAS SCODE-CONGRA tool needed ▪ Currently no command-line interface
Intended user(s):	▪ model developers ▪ software developers ▪ software integrators
Provider:	▪ ETAS GmbH, Stuttgart
Contact point:	▪ Christoph Malz, christoph.malz@etas.com
Condition(s) for reuse:	▪ ETAS SCODE-CONGRA license
	Latest update: 26.01.2021

Name: ETAS SCODE-CONGRA Flat-Modelica converter		
Input(s):	Main feature(s)	Output(s):
▪ Model in Flat Modelica syntax	▪ Conversion into SCODE-CONGRA systems ▪ Representation of SCODE-CONGRA models in Flat Modelica	▪ SCODE-CONGRA model

Unique Selling Proposition(s):	▪ Wide range of existing models in Modelica tools become easily usable within SCODE-CONGRA, allowing the SCODE-CONGRA analysis, model definition and generation capabilities to be applied ▪ System simplification allows easier system understanding and generation of embedded code with optimized code size and runtime.
Integration constraint(s):	▪ ETAS SCODE-CONGRA needed ▪ Currently no command-line interface
Intended user(s):	▪ model developers ▪ software developers ▪ software integrators
Provider:	▪ ETAS GmbH, Stuttgart
Contact point:	▪ Christoph Malz, christoph.malz@etas.com
Condition(s) for reuse:	▪ ETAS SCODE-CONGRA license ▪
	Latest update: 26.01.2021

Name: eFMI Support in Modelon Impact / OPTIMICA Compiler Toolkit		
Input(s):	Main feature(s)	Output(s):
▪ Modelica Model	▪ Support for eFMI AlgCode generation coupled with support for inlining integration algorithms suitable for real-time simulations.	▪ AlgCode eFMU

Unique Selling Proposition(s):	▪ Support for Modelica Standard Library ▪ Support for state-of-the-art Modelon's Modelica libraries covering a wide range of domains
Integration constraint(s):	▪ Increased support for a wider range of Modelica models will be supported after end of EMPHYSIS
Intended user(s):	▪ Customers of Modelon Impact / OPTIMICA Compiler Toolkit
Provider:	▪ Modelon AB
Contact point:	▪ Johan Windahl, <johan.windahl@modelon.com>
Condition(s) for reuse:	▪ Undecided
	Latest update: 2021-01-27