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

ITEA2 P13017 AMALTHEA4public

Project details

Project leader:	Joerg Tessmer
Email:	Joerg.tessmer@de.bosch.com
Website:	http://www.amalthea-project.org/

Name: APP4MC		
Input(s):	Main feature(s)	Output(s):
▪ SW-description ▪ HW-description ▪ Constraints ▪ Decisions ▪ Costs	▪ Multi- and Many-Core development process support ▪ Common Data exchange and simulation format ▪ Event tracing ▪ Customizable workflow	▪ SW distribution for embedded multicore systems ▪ Common data exchange and simulation format

Unique Selling Proposition(s):	▪ Consistent continuous tooling ▪ Development efficiency increase ▪ De-facto standard for data exchange ▪ New services and functions ▪ Traceability for systems engineering artifacts
Integration constraint(s):	▪ designed for Laptop or desktop machines ▪ JAVA 8 ▪ Supported platforms: Win32, Win 64, Linux (64 bit), OSX (64 bit)
Intended user(s):	▪ SW-architects, SW-developer, HW designer,
Provider:	▪ Eclipse APP4MC community ▪ https://www.eclipse.org/app4mc/community/
Contact point:	▪ https://www.eclipse.org/app4mc/community/
Condition(s) for reuse:	▪ EPL licensed (Eclipse public license) ▪
	Latest update: 2017-04-30

Name: Eclipse Capra		
Input(s):	Main feature(s)	Output(s):
▪ Design Artifacts ▪ Requirements ▪ Code ▪ Test Cases ▪ Bugs and Tickets	▪ End-to-End traceability ▪ Visualization of traceability links ▪ Consistency checks with semi-automated consistency fixes ▪ Supports change impact analysis ▪ Highly configurable and extensible ▪ Support for many common DSLs	▪ Traceability Matrix ▪ Traceability Graph

Unique Selling Proposition(s):	▪ Open source solution addressing scientifically validated traceability needs ▪ Highly customizable w.r.t. traceability link semantics, supported artifact types, visualization, etc., thus allowing adaptation for specific project environment ▪ Seamless integration with the Eclipse IDE ▪ Under active development by an open community
Integration constraint(s):	▪ Designed for the Eclipse Platform ▪ Concrete artifact types need to be supported through handler (can be implemented by each user)
Intended user(s):	▪ Software development organizations with traceability needs
Provider:	▪ Eclipse Capra community
Contact point:	▪ https://eclipse.org/capra
Condition(s) for reuse:	▪ Licensed under EPL (Eclipse Public License)

Latest update: 2017-04-30

<table>
<tr><th colspan="4">Name: MechatronicUML</th></tr>
<tr><td>Input(s):</td><td colspan="2">Main feature(s)</td><td>Output(s):</td></tr>
<tr><td>Software requirements specification</td><td colspan="2">Software modeling for cooperating, technical systems
APP4MC Export
Domain-specific model checking
Export for Model-in-the-Loop simulation with COTS-Tools
Hardware modeling and deployment
Software reconfiguration
Based on open-source Eclipse tooling
Source code generation
Holistic open source example for an advanced driver assistance system</td><td>Platform-independent software model
Hardware and deployment models
Simulation models (Matlab/Modelica)
ANSI C99 source code</td></tr>
<tr><td>Unique Selling Proposition(s):</td><td colspan="3">Correctness by construction
Seamless tool support
Integrated formal analysis
Extensible
Platform-independent development
Faster development</td></tr>
<tr><td>Integration constraint(s):</td><td colspan="3">Designed for laptop or desktop machines
Eclipse Neon, SR2
Java 8
Supported platforms: Win32, Win 64, Linux (32 bit), Linux (64 bit)</td></tr>
<tr><td>Intended user(s):</td><td colspan="3">SW Architect, SW Developer, Deployment Engineer</td></tr>
<tr><td>Provider:</td><td colspan="3">Heinz Nixdorf Institute at Paderborn University
Fraunhofer IEM</td></tr>
<tr><td>Contact point:</td><td colspan="3">http://www.mechatronicuml.org/en/index.html</td></tr>
<tr><td>Condition(s) for reuse:</td><td colspan="3">EPL licensed (Eclipse public license)</td></tr>
<tr><td></td><td colspan="3"><i>Latest update: 2017-04-30</i></td></tr>
</table>

Name: ScenarioTools MSD		
Input(s):	Main feature(s)	Output(s):
▪ System Design Model	▪ Scenario-based, formal requirements specification for cooperating, technical systems ▪ Modal Sequence Diagrams (MSDs) ▪ Based on open-source Eclipse tooling ▪ Holistic open source example for an advanced driver assistance system	▪ Software requirements specification

Unique Selling Proposition(s):	▪ UML-compliant ▪ Executable requirements specification ▪ Reproducible system behavior by means of simulation (Play-Out) ▪ Formal verification for requirements consistency and implementability
Integration constraint(s):	▪ Designed for laptop or desktop machines ▪ Java 8 ▪ Supported platforms: Win32, Win 64, Linux (32 bit), Linux (64 bit)
Intended user(s):	▪ SW Requirements Engineer
Provider:	▪ Leibniz Universität Hannover ▪ Heinz Nixdorf Institute at Paderborn University ▪ Fraunhofer IEM
Contact point:	▪ http://scenariotools.org/projects/msd/ ▪ http://www.mechatronicuml.org/en/index.html
Condition(s) for reuse:	▪ EPL licensed (Eclipse public license)

Latest update: 2017-04-30