

# Exploitable Results by Third Parties

ITEA2 P13017 AMALTHEA4public

---

## Project details

Project leader:	Joerg Tessmer
Email:	<a href="mailto:Joerg.tessmer@de.bosch.com">Joerg.tessmer@de.bosch.com</a>
Website:	<a href="http://www.amalthea-project.org/">http://www.amalthea-project.org/</a>

Name: APP4MC		
Input(s):	Main feature(s)	Output(s):
<ul style="list-style-type: none"> <li>▪ SW-description</li> <li>▪ HW-description</li> <li>▪ Constraints</li> <li>▪ Decisions</li> <li>▪ Costs</li> </ul>	<ul style="list-style-type: none"> <li>▪ Multi- and Many-Core development process support</li> <li>▪ Common Data exchange and simulation format</li> <li>▪ Event tracing</li> <li>▪ Customizable workflow</li> </ul>	<ul style="list-style-type: none"> <li>▪ SW distribution for embedded multicore systems</li> <li>▪ Common data exchange and simulation format</li> </ul>
Unique Selling Proposition(s):	<ul style="list-style-type: none"> <li>▪ Consistent continuous tooling</li> <li>▪ Development efficiency increase</li> <li>▪ De-facto standard for data exchange</li> <li>▪ New services and functions</li> <li>▪ Traceability for systems engineering artifacts</li> </ul>	
Integration constraint(s):	<ul style="list-style-type: none"> <li>▪ designed for Laptop or desktop machines</li> <li>▪ JAVA 8</li> <li>▪ Supported platforms: Win32, Win 64, Linux (64 bit), OSX (64 bit)</li> </ul>	
Intended user(s):	<ul style="list-style-type: none"> <li>▪ SW-architects, SW-developer, HW designer,</li> </ul>	
Provider:	<ul style="list-style-type: none"> <li>▪ Eclipse APP4MC community</li> <li>▪ <a href="https://www.eclipse.org/app4mc/community/">https://www.eclipse.org/app4mc/community/</a></li> </ul>	
Contact point:	<ul style="list-style-type: none"> <li>▪ <a href="https://www.eclipse.org/app4mc/community/">https://www.eclipse.org/app4mc/community/</a></li> </ul>	
Condition(s) for reuse:	<ul style="list-style-type: none"> <li>▪ EPL licensed (Eclipse public license)</li> <li>▪</li> </ul>	
<i>Latest update: 2017-04-30</i>		

Name: Eclipse Capra		
Input(s):	Main feature(s)	Output(s):
<ul style="list-style-type: none"> <li>▪ Design Artifacts</li> <li>▪ Requirements</li> <li>▪ Code</li> <li>▪ Test Cases</li> <li>▪ Bugs and Tickets</li> </ul>	<ul style="list-style-type: none"> <li>▪ End-to-End traceability</li> <li>▪ Visualization of traceability links</li> <li>▪ Consistency checks with semi-automated consistency fixes</li> <li>▪ Supports change impact analysis</li> <li>▪ Highly configurable and extensible</li> <li>▪ Support for many common DSLs</li> </ul>	<ul style="list-style-type: none"> <li>▪ Traceability Matrix</li> <li>▪ Traceability Graph</li> </ul>
Unique Selling Proposition(s):	<ul style="list-style-type: none"> <li>▪ Open source solution addressing scientifically validated traceability needs</li> <li>▪ Highly customizable w.r.t. traceability link semantics, supported artifact types, visualization, etc., thus allowing adaptation for specific project environment</li> <li>▪ Seamless integration with the Eclipse IDE</li> <li>▪ Under active development by an open community</li> </ul>	
Integration constraint(s):	<ul style="list-style-type: none"> <li>▪ Designed for the Eclipse Platform</li> <li>▪ Concrete artifact types need to be supported through handler (can be implemented by each user)</li> </ul>	
Intended user(s):	<ul style="list-style-type: none"> <li>▪ Software development organizations with traceability needs</li> </ul>	
Provider:	<ul style="list-style-type: none"> <li>▪ Eclipse Capra community</li> </ul>	
Contact point:	<ul style="list-style-type: none"> <li>▪ <a href="https://eclipse.org/capra">https://eclipse.org/capra</a></li> </ul>	
Condition(s) for reuse:	<ul style="list-style-type: none"> <li>▪ Licensed under EPL (Eclipse Public License)</li> </ul>	

*Latest update: 2017-04-30*

Name: MechatronicUML		
Input(s):	Main feature(s)	Output(s):
<ul style="list-style-type: none"> <li>▪ Software requirements specification</li> </ul>	<ul style="list-style-type: none"> <li>▪ Software modeling for cooperating, technical systems</li> <li>▪ APP4MC Export</li> <li>▪ Domain-specific model checking</li> <li>▪ Export for Model-in-the-Loop simulation with COTS-Tools</li> <li>▪ Hardware modeling and deployment</li> <li>▪ Software reconfiguration</li> <li>▪ Based on open-source Eclipse tooling</li> <li>▪ Source code generation</li> <li>▪ Holistic open source example for an advanced driver assistance system</li> </ul>	<ul style="list-style-type: none"> <li>▪ Platform-independent software model</li> <li>▪ Hardware and deployment models</li> <li>▪ Simulation models (Matlab/Modelica)</li> <li>▪ ANSI C99 source code</li> </ul>
Unique Selling Proposition(s):	<ul style="list-style-type: none"> <li>▪ Correctness by construction</li> <li>▪ Seamless tool support</li> <li>▪ Integrated formal analysis</li> <li>▪ Extensible</li> <li>▪ Platform-independent development</li> <li>▪ Faster development</li> </ul>	
Integration constraint(s):	<ul style="list-style-type: none"> <li>▪ Designed for laptop or desktop machines</li> <li>▪ Eclipse Neon, SR2</li> <li>▪ Java 8</li> <li>▪ Supported platforms: Win32, Win 64, Linux (32 bit), Linux (64 bit)</li> </ul>	
Intended user(s):	<ul style="list-style-type: none"> <li>▪ SW Architect, SW Developer, Deployment Engineer</li> </ul>	
Provider:	<ul style="list-style-type: none"> <li>▪ Heinz Nixdorf Institute at Paderborn University</li> <li>▪ Fraunhofer IEM</li> </ul>	
Contact point:	<ul style="list-style-type: none"> <li>▪ <a href="http://www.mechatronicuml.org/en/index.html">http://www.mechatronicuml.org/en/index.html</a></li> </ul>	
Condition(s) for reuse:	<ul style="list-style-type: none"> <li>▪ EPL licensed (Eclipse public license)</li> </ul>	
<i>Latest update: 2017-04-30</i>		

Name: ScenarioTools MSD		
Input(s):	Main feature(s)	Output(s):
<ul style="list-style-type: none"> <li>System Design Model</li> </ul>	<ul style="list-style-type: none"> <li>Scenario-based, formal requirements specification for cooperating, technical systems</li> <li>Modal Sequence Diagrams (MSDs)</li> <li>Based on open-source Eclipse tooling</li> <li>Holistic open source example for an advanced driver assistance system</li> </ul>	<ul style="list-style-type: none"> <li>Software requirements specification</li> </ul>
Unique Selling Proposition(s):	<ul style="list-style-type: none"> <li>UML-compliant</li> <li>Executable requirements specification</li> <li>Reproducible system behavior by means of simulation (Play-Out)</li> <li>Formal verification for requirements consistency and implementability</li> </ul>	
Integration constraint(s):	<ul style="list-style-type: none"> <li>Designed for laptop or desktop machines</li> <li>Java 8</li> <li>Supported platforms: Win32, Win 64, Linux (32 bit), Linux (64 bit)</li> </ul>	
Intended user(s):	<ul style="list-style-type: none"> <li>SW Requirements Engineer</li> </ul>	
Provider:	<ul style="list-style-type: none"> <li>Leibniz Universität Hannover</li> <li>Heinz Nixdorf Institute at Paderborn University</li> <li>Fraunhofer IEM</li> </ul>	
Contact point:	<ul style="list-style-type: none"> <li><a href="http://scenariotools.org/projects/msd/">http://scenariotools.org/projects/msd/</a></li> <li><a href="http://www.mechatronicuml.org/en/index.html">http://www.mechatronicuml.org/en/index.html</a></li> </ul>	
Condition(s) for reuse:	<ul style="list-style-type: none"> <li>EPL licensed (Eclipse public license)</li> </ul>	

*Latest update: 2017-04-30*