Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles
## Authors

<table>
<thead>
<tr>
<th>Editor</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carl-Johan Sjöstedt</td>
<td><a href="mailto:carlj@md.kth.se">carlj@md.kth.se</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authors</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carl-Johan Sjöstedt</td>
<td><a href="mailto:carlj@md.kth.se">carlj@md.kth.se</a></td>
</tr>
<tr>
<td>Matthias Biehl</td>
<td><a href="mailto:biehl@md.kth.se">biehl@md.kth.se</a></td>
</tr>
<tr>
<td>David Servat</td>
<td><a href="mailto:david.servat@cea.fr">david.servat@cea.fr</a></td>
</tr>
<tr>
<td>Martin Walker</td>
<td><a href="mailto:martin.walker@hull.ac.uk">martin.walker@hull.ac.uk</a></td>
</tr>
<tr>
<td>Frang Hagl</td>
<td><a href="mailto:frank.hagl@continental-corporation.com">frank.hagl@continental-corporation.com</a></td>
</tr>
</tbody>
</table>

## The Consortium

Volvo Technology Corporation (S)  
Continental Automotive (D)  
MetaCase (Fi)  
Kungliga Tekniska Högskolan (S)  
Centro Ricerche Fiat (I)  
Delphi/Mecel (S)  
Systemite (SE)  
4S Group (I)  
Pulse-AR (Fr)  
Technische Universität Berlin (D)  
CEA LIST (F)  
University of Hull (GB)
## Revision chart and history log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2011-05-31</td>
<td>First release</td>
</tr>
</tbody>
</table>
Table of contents

Authors ............................................................................................................................ 2
Revision chart and history log ......................................................................................... 3
Table of contents ........................................................................................................... 4
1 Introduction .................................................................................................................. 5
2 The components of the MAENAD Analysis Platform .................................................. 6
  2.1 AUTOSAR Gateway ............................................................................................... 7
    2.1.1 Current status................................................................................................... 7
    2.1.2 Future plans ..................................................................................................... 7
    2.1.3 Requirements from WT2.1: Identifications of needs ....................................... 7
  2.2 Timing Analysis ...................................................................................................... 8
    2.2.1 Current status................................................................................................... 8
    2.2.2 Future plans .................................................................................................... 8
    2.2.3 Requirements from WT2.1: Identifications of needs ....................................... 8
  2.3 Simulink Gateway .................................................................................................. 9
    2.3.1 Current status................................................................................................... 9
    2.3.2 Future plans .................................................................................................... 9
    2.3.3 Requirements from WT2.1: Identifications of needs ....................................... 10
  2.4 HiP-HOPS Gateway ............................................................................................... 11
    2.4.1 Current status................................................................................................... 11
    2.4.2 Future plans .................................................................................................... 11
    2.4.3 Requirements from WT2.1: Identifications of needs ....................................... 11
  2.5 Architecture optimization and configuration ............................................................ 13
    2.5.1 Current status................................................................................................... 13
    2.5.2 Future plans .................................................................................................... 13
    2.5.3 Requirements from WT2.1: Identifications of needs ....................................... 13
  2.6 Plugin for EAST-ADL exchange format EAXML ................................................... 14
    2.6.1 Current status................................................................................................... 14
    2.6.2 Future plans .................................................................................................... 14
    2.6.3 Requirements from WT2.1: Identifications of needs ....................................... 14
  2.7 Modelica Exchange ............................................................................................... 15
    2.7.1 Current status................................................................................................... 15
    2.7.2 Future plans .................................................................................................... 15
    2.7.3 Requirements from WT2.1: Identifications of needs ....................................... 15
  2.8 Modelisar FMU Import ........................................................................................... 16
    2.8.1 Current status................................................................................................... 16
    2.8.2 Future plans .................................................................................................... 16
    2.8.3 Requirements from WT2.1: Identifications of needs ....................................... 16
3 References .................................................................................................................. 17
1 Introduction

Deliverable D5.2.1 consists of:

- The various plugins and tools that make up the MAENAD analysis workbench (MAW)
- This document providing references and short descriptions of the tools in the MAW.

The MAW consists of software developed to provide support for modelling and analysis, based on specifications of WT3.x. One objective is to validate the analysis concepts. The MAW will be made public in order to make the analysis concepts more accessible and understandable.

This intermediate release consists of the sections Current status, Future plans and Requirements, where the latter describes what project and language requirements that the plugin/tool fulfills. The final deliverable will also include installation/usage instructions, not available for this release.

The target of MAW is the MAENAD Modeling Workbench, D5.1.1, but with the model exchange tool, as described in section 2.6. They will also work with the tool adaptations (MetaEdit+ and System Weaver), which are developed in Work Task 5.3.
2 The components of the MAENAD Analysis Platform

Some of the plugins were developed throughout the ATESST [1], ATEST2 [1] and EDONA [2] projects, and will be updated during the MAENAD project, because of new releases of the MAENAD modeling workbench, and new releases of the profile. There will also be new plugins, for the interchange of EAST-ADL models using an exchange format, and for exchange with ModelicaML and Modelisar FMU:s.

<table>
<thead>
<tr>
<th>Main developer</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOSAR Gateway</td>
<td>CEA Provide an enhanced transformation from EAST-ADL2 design architecture to AUTOSAR compliant software architecture, based on the ARTOP framework.</td>
</tr>
<tr>
<td>Timing Analysis</td>
<td>CEA Provides link to schedulability analysis tool (to be chosen during the project) and help for modelling of timing related information.</td>
</tr>
<tr>
<td>Simulink Gateway</td>
<td>KTH Provides input/output facilities with Simulink to enable simulation.</td>
</tr>
<tr>
<td>HiP-HOPS Gateway</td>
<td>KTH Builds on previous results, expanding analysis capability and optimization engine of HiP-HOPS, and enhancing the feedback of FMEA/FTA in the design process.</td>
</tr>
<tr>
<td>Architecture optimization and configuration</td>
<td>University of Hull Builds on CVM, the variability management plugin from ATESTST2, which will be extended and interfaced with an optimization engine.</td>
</tr>
<tr>
<td>Plugin for EAST-ADL exchange format EAXML</td>
<td>CEA Provides exchange between EAST-ADL2 models in a UML2 tool and an AUTOSAR-compliant XML exchange format</td>
</tr>
<tr>
<td>Modelica Exchange</td>
<td>KTH Using ModelicaML together with EAST-ADL for analysis.</td>
</tr>
<tr>
<td>Modelisar FMU Import</td>
<td>VTEC Generate EAST-ADL FAA models from Modelisar Functional Mockup Units.</td>
</tr>
</tbody>
</table>
2.1 AUTOSAR Gateway

The AUTOSAR gateway builds on results from EDONA and ATESST2 projects to provide an enhanced transformation from EAST-ADL design architecture to AUTOSAR compliant software architecture, based on the ARTOP framework, which is an implementation of common base functionality for AUTOSAR development tools, available free of charge for AUTOSAR members [5]. The transformation also takes into account allocations and hardware description architecture from the EAST-ADL model to generate a tentative topology for the AUTOSAR architecture.

2.1.1 Current status

The plugin based on the one developed in the EDONA project is being ported to the new Papyrus MDT platform and current EAST-ADL profile version. This is needed as in EDONA development was performed on an earlier version of the profile. Release of this update is planned for M12.

2.1.2 Future plans

Beyond M12 the next version of the plugin will rely on the definition of an AUTOSAR UML profile (subset of AUTOSAR centered on relevant templates: software component, system and ECU resource namely). As a result, the transformation from an EAST-ADL design architecture to an AUTOSAR implementation architecture will produce UML profiled models. An export functionality from AUTOSAR-UML profiled models to AUTOSAR XML will also be developed. Planned releases for this plugin are M24 and M30.

2.1.3 Requirements from WT2.1: Identifications of needs

The AUTOSAR gateway is mentioned in the following requirement:

DOW#0111: The SWC Synthesis (FDA-IL) shall be modeled in MAW-AR Gateway plugin [4].
2.2 Timing Analysis

The Timing Analysis plugin builds on results from EDONA and ATESSST2 projects to provide schedulability means on EAST-ADL models. The schedulability analysis engine itself (or link to such an engine, provided as a third-party tool) will not be developed in MAENAD, rather the adaptation of a bridge to such an engine will be provided to enable EAST-ADL models analysis. As such, the gap between what EAST-ADL models provide for in terms of timing information and what is needed to perform such an analysis will be assessed and means for manual edition or automated processing will be documented/developed in the course of the project.

2.2.1 Current status

The plugin based on the one developed in the EDONA project is being ported to the new Papyrus MDT platform and current EAST-ADL profile version. Essentially the first version to be released for M12 will not allow a direct analysis of EAST-ADL models. Rather such models will have to be enriched with MARTE stereotypes to be analyzed by the tool, and some additional diagrammatic information provided. Some assumptions will be made for instance the execution flow that is implicitly defined in the FDA will be explicitly defined as an execution flow of tasks and frames in an UML activity diagram, which is required by the tool at this stage.

2.2.2 Future plans

The MAENAD project will release in M12 a new version of the EAST-ADL profile, which will map EAST-ADL concepts to MARTE concepts (i.e. deliverable D4.2.1). Based on this new version of the profile, developments will be performed on the Timing Analysis plugin to provide a more direct analysis of EAST-ADL models. However, the extent of extra information a user will have to manually add to the conventional EAST-ADL model or that will be automated, will be assessed in the course of the development. This assessment will be done in link with the needs and goals set for the optimization experiments and tool architecture. Planned releases for these updates are M24 and M30.

2.2.3 Requirements from WT2.1: Identifications of needs

The Timing Analysis plugin is mentioned in the following requirement:

DOW#0110: The Timing analysis (DL) shall be modelled in MAW-Timing plugin [4].
2.3 Simulink Gateway

The Simulink gateway builds on results from the ATESST2 project, and provides input/output facilities of models with Simulink to enable simulation. The plugin is divided in two parts (Figure 1):

- A GUI plugin to the MATLAB/Simulink environment, which aids the user in creating models that conform to the format that is needed to being able to convert it into an EAST-ADL model.
- An Eclipse plugin, which can convert between the intermediate format of Simulink models and EAST-ADL models

The MATLAB plugin exports the MATLAB/Simulink models into a custom Ecore-based format. A subset of Simulink functions is used; only library blocks of subsystems are considered. However, any Simulink model could be converted into a structure of system reference blocks, without affecting the model's simulation behavior. The GUI plugin for Simulink mentioned above converts standard Simulink subsystems to system reference blocks, and tags them for conversion to EAST-ADL by putting them in a "FunctionTypes"-library, and assigning a unique ID, to allow bi-directional exchange and updates. To include the internal structure of a subsystem, the same pattern is repeated.

Import works the other way around, FAA FunctionTypes and FunctionPrototypes are imported to empty library blocks in the "FunctionTypes"-library, and instances of them respectively.

2.3.1 Current status

The plugin is based on the one developed in the ATESST2 project, ported to the new Papyrus MDT version. Core functionality is working, but there are various major and minor issues to deal with, e.g. implementing bi-directional transformation, support environment model, support for different ports, etc.

2.3.2 Future plans

There are no concrete plans for further development of the Simulink plugin. Case studies need to be setup, to show what type of things to focus on. This could include allowing bi-directional exchange of models, allow simulation results to be exchanged, alignment with EAST-ADL behavior model (FunctionBehavior), exchange of other abstraction levels than Functional Analysis Architecture (e.g. Functional Design Architecture), to include a larger meta-model of Simulink models, etc. There are also numerous usability issues, e.g. checking that Simulink models are valid, i.e. conforms to the metamodels defined above, allow graphical information to propagate, and more.
2.3.3 Requirements from WT2.1: Identifications of needs

A2#11: A formalized meta-model of the architecture description language shall be developed. (including structural elements, behavioural description means, models of computations, and transformation rules to prototype tools and Simulink.)

DOW#0112: The Simulink import-export ( FAA, FDA ) shall be modelled in MAW-Simulink plugin.
2.4 HiP-HOPS Gateway

Integrating safety analysis into the development of automotive embedded systems requires translating concepts of the automotive domain to the generic safety and error analysis domain. We assume a model-based development process where automotive concepts are represented by the EAST-ADL2 architecture description language, which supports system design on multiple levels of abstraction. The concepts of the error analysis domain are represented by the safety analysis tool HiP-HOPS.

It is assumed that EAST-ADL2 models are built using the UML profile. The HiP-HOPS plugin extensively uses the concepts defined in the EAST-ADL error model, but also other language constructs from the FDA level.

We automate the translation from EAST-ADL2 to HiP-HOPS by using model transformations. We leverage the advantages of different model transformation techniques by decomposing the translation into two distinct phases, and using an appropriate technique for each phase: A phase for conceptual mapping between the domains followed by a phase for representing the output in the desired concrete syntax.

With the resulting tight integration of the safety analysis tool and the model-based development environment, the automotive safety engineer can perform the safety analysis repeatedly on refined models with minimal effort. This is compliant with the iterative design activities, which require starting the analysis after each change in the system design.

The HiP-HOPS Gateway builds on previously developed Model Transformations and Eclipse Plugin, expanding analysis capability and optimization engine of HiP-HOPS, and enhancing the feedback of FMEA/FTA in the design process.

2.4.1 Current status

The plugin is based on the one developed in the ATESST2 project. It is assumed that EAST-ADL2 models are built using the UML profile version 2.1.5 from the end of ATESST2. The HiP-HOPS plugin extensively uses the concepts defined in the EAST-ADL error-model, but also other language constructs from the FDA level. Core functionality is working, including HW-SW allocation.

2.4.2 Future plans

Maintenance until M12:

- Update of the plugin to work with the current version of Papyrus
- Creation of a test model for the current version of Papyrus

Enhancements until M12: Support for ASIL allocation based on the UML/Profile version of the Plugin:

- Extension of the transformation
- Intermediate metamodel, HiP-HOPS input file

It is currently open and under discussion if the HiP-HOPS optimization engine will use the existing EAST-ADL to HiP-HOPS gateway or if HiP-HOPS will natively support the EAXML format.

2.4.3 Requirements from WT2.1: Identifications of needs

UOH#0003: The HiP-HOPS analysis tool should support any ISO 26262 or related concepts (such as ASIL decomposition) necessary to allow ISO-compatible dependability analysis of EAST-ADL models.
UOH#0004: EAST-ADL and HiP-HOPS should be able to intercommunicate by means of model transformations provided by a dependability plugin in the MAENAD Analysis Workbench (MAW). Furthermore it should be possible to import or store the results from HiP-HOPS in the Workbench and/or the EAST-ADL model, which will require establishing some form of (perhaps XML based) interchange format.
2.5 Architecture optimization and configuration

The architectural optimization & configuration capabilities to be developed in MAENAD build upon several tools and plugins, including CVM (and the corresponding variability management plugin from ATESSST2), HiP-HOPS (and its associated plugin), and timing analysis tools like MAST (and associated plugin(s)). These tools need to be interfaced with an optimization engine for fully multi-objective optimization of EAST-ADL models to be possible.

2.5.1 Current status

Initial work on concepts has taken place and initial steps towards an implementation have been made, although additional investigation is still required. A physical meeting took place in York (UK) in May 2011 where the optimization concepts and ideas about what the tool architecture may look like were discussed. Primary involved partners are TUB (for resolution of variability), UOH (for the optimization engine and genetic algorithms), CEA (for timing analysis and Papyrus integration assistance), and KTH.

2.5.2 Future plans

It was agreed at York that the main implementation work would begin after M12 (before M12, ASIL decomposition work is being prioritized instead) although further conceptual work would continue before then. As a first step, it was proposed that a subset of the overall capability would be developed first involving cost vs reliability optimization using HiP-HOPS itself as the optimization engine and allowing export of EAST-ADL models with variability to be converted into HiP-HOPS models with HiP-HOPS variability (implementations etc). This would enable some experimentation and validation of the concepts involved.

A more substantial tool architecture involving a central optimization engine (independent from the HiP-HOPS analysis tool) and a variability resolution mechanism to dispatch resolved models to different analysis tools for evaluation was also proposed. This ideal would be worked towards throughout the project.

A case study based on a brake-by-wire model was also planned, to be developed by VTEC.

2.5.3 Requirements from WT2.1: Identifications of needs

VTEC#UC006: A model of the validator with timing, dependability and cost annotations as well as design space, variability and take rate annotations is defined and exported to EAXML. An optimization tool computes the optimal design for the defined product line. The resulting optimized model is recorded in the model (design space variability removed) and exported in the EAXML file.

UOH#0002: The EAST-ADL error model should fully support automatic optimisation, e.g. through rules that specify a 1:1 mapping from nominal to error models.

UOH#0005: To support multi-objective optimisation, there must be a standardised way of passing design candidates to analysis tools/plugins and receiving results in a given format.
2.6 Plugin for EAST-ADL exchange format EAXML

The model exchange plugin provides exchange between EAST-ADL2 models in a UML2 tool and an AUTOSAR-compliant XML exchange format: EAXML.

2.6.1 Current status

Development is postponed until a complete workbench based on ATESST2 results update is secured. Decision was made to have a joint collaboration on this track and iterative version.

2.6.2 Future plans

CEA/Pulse-AR proposed to investigate the following restricted scenario: export to EAXML an existing EAST-ADL profile-based model from Papyrus; import an EAXML model as a new EAST-ADL profile-based UML model.

For this, CEA/Pulse-AR suggested to make use of the investigations that will be made for the AR Gateway reengineering (see above 2.1): this work indeed demands a connection between an AR profile UML model and an ARXML model. Adaptation of this work could be made to support the scenario during second half of year 2.

From this simple scenario, other contributors would define extensions as needed.

2.6.3 Requirements from WT2.1: Identifications of needs

EAXML is mentioned in VTEC#UC001 through VTEC#UC008 [4], which describe scenarios where 1) models are exchanged from the modeling workbench and the external tools of WT5.3; 2) the analysis tools of WT5.2 are used in link with the external tools of WT5.3, independently from the modeling workbench. Such scenarios rely on exchange based on EAXML files produced/read by the various tool chain participants.
2.7 Modelica Exchange

Modelica is a language for modeling and simulation of dynamical systems, and could be used for various analyses of EAST-ADL models, e.g. modeling of plant models or timing constraints.

2.7.1 Current status

There is a plugin for Papyrus MDT for ModelicaML, using the UML stereotype mechanism. By assigning both an EAST-ADL FunctionType and a ModelicaML stereotype to a class, Modelica behavior can be assigned to EAST-ADL FunctionTypes.

Another possibility is to use the Functional Mockup Interface to exchange Modelica models with EAST-ADL. A prototype plugin for getting structural information from Functional Mockup Units and transforming it into EAXML has been developed by VTEC.

2.7.2 Future plans

One way of co-use of EAST-ADL and ModelicaML is using the EAST-ADL structural model in combination with the ModelicaML dynamic model. The structural parts could be mapped more or less directly 1:1 into ModelicaML, the dynamic parts can be modeled by ModelicaML as the behavioral parts of a FunctionType. The FunctionType in this case should be seen as an AUTOSAR runnable.

ModelicaML will be used in order to do verification of TADL timing constraints in the context of the TIMMO2USE project by Continental during the fall of 2011 and spring 2012. Based on these efforts, a plug-in for TADL timing constraints within MAENAD might be an option.

A summary on how to co-use EAST-ADL and ModelicaML could be developed, in form of an alignment/mapping of ModelicaML and EAST-ADL stereotypes and possibly an automatic conversion.

2.7.3 Requirements from WT2.1: Identifications of needs

Modelica exchange is mentioned in e.g.:

CON#0009: Annotate SysML/Modelica models with EAST-ADL stereotypes. On base of a defined mapping between SysML and EAST-ADL, the SysML model of the profile and mode selection logic shall be annotated with EAST-ADL stereotypes. Structural as well as behavioral elements shall be annotated with EAST-ADL stereotypes CON#0012: EAST-ADL supports the definition of timing of constraints by the inclusion of the TADL language. A verification of the TADL constraints shall be possible. It is an option to verify TADL constraints either by the use of timing analysis techniques as provided by languages as MARTE or AADL or model simulation techniques as provided by Modelica. Within the scope of the project, it has to be evaluated, if the verification of timing constraints on base of these techniques is possible and samples shall be given.

CON#0014: VV Case Development, including fault injection and verification of model constraints in a Modelica simulation environment.

CON#0021: Virtual integration is an important use case during development within the ID4EV project. It is obvious that the physical demonstrator will not be available for a long time and the SW modules must be integrated in a virtual environment. The Modelica simulation environment is very well suited for integration C-code into the simulation environment.
2.8 Modelisar FMU Import

The Modelisar project [5] has defined Function Mockup Units to exchange and integrate simulation blocks from different modeling tools.

2.8.1 Current status

There is an Eclipse plugin that imports the Function Mockup Unit specification, called Function Mockup Interface (FMI). The Function Mockup Interface defines the input and output variables of each unit and also the data types of these variables. Based on this information, an AnalysisFunctionType with the corresponding interface is defined in EAXML.

2.8.2 Future plans

The current plugin has flaws concerning the formatting of the EAXML file, which needs to be corrected. Further, additional options for the import can be foreseen. Currently, only AnalysisFunction is supported, but any specialization of the FunctionType is a candidate for import. Further, the FunctionBehavior construct is currently not created and populated with FMU information, such as the path to the FMU file.

A more extensive potential addition concerns export. Export in the form of FMU generation is probably not appropriate since EAST-ADL is not primarily a behavioral definition language. However, export of the FMU:s linked to FunctionBehaviors to a simulation engine would be useful. This would concern creating S-functions in Simulink according to the connected FunctionPrototypes or to configure a Simulation manager to run the executables according to execution and connection information defined in the EAST-ADL model.

2.8.3 Requirements from WT2.1: Identifications of needs

Modelisar FMU import is not explicitly mentioned in the requirements, although it relates to behavioral simulation in general.

DOW#0012 O2-2: Behavioural Simulation of EAST-ADL2 models

4SG#0003: Perform behavioural Simulation of EAST-ADL2 models according to performance evaluation standards

4SG#0004: Perform behavioural Simulation of EAST-ADL2 models according to standards covering communication with infrastructures

4SG#0019: The project shall enable to perform behavioural simulation according to ISO 8715: Electric road vehicles - Road operating characteristics

There are more similar requirements on behavior simulations, see e.g.

4SG#0020, 4SG#0021, 4SG#0022, 4SG#0023, 4SG#0024, 4SG#0025, 4SG#0026
3 References

[4] MAENAD: Deliverable D2.1.1, draft version 0.5