Advancing Traffic Efficiency and Safety through Software Technology phase 2 (ATESST2)

Report type: Deliverable D4.3.1
Report name: The EAST-ADL Analysis Platform

Dissemination level: PU
Status: Final release
Version number: 1.0
Date of preparation: 2009-11-04
## Authors

<table>
<thead>
<tr>
<th>Editor</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matthias Biehl</td>
<td><a href="mailto:biehl@md.kth.se">biehl@md.kth.se</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authors</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark-Oliver Reiser</td>
<td><a href="mailto:moreiser@cs.tu-berlin.de">moreiser@cs.tu-berlin.de</a></td>
</tr>
<tr>
<td>Helko Glathe</td>
<td><a href="mailto:helko.glathe@carmeq.com">helko.glathe@carmeq.com</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>Yann Tanguy</td>
<td><a href="mailto:Yann.Tanguy@cea.fr">Yann.Tanguy@cea.fr</a></td>
</tr>
</tbody>
</table>

## The Consortium

<table>
<thead>
<tr>
<th>Volvo Technology Corporation (S)</th>
<th>VW/Carmeq (D)</th>
<th>Centro Ricerche Fiat (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continental Automotive (D)</td>
<td>Delphi/Mecel (S)</td>
<td></td>
</tr>
<tr>
<td>Mentor Graphics Hungary (H)</td>
<td>CEA LIST (F)</td>
<td></td>
</tr>
<tr>
<td>Kungliga Tekniska Högskolan (S)</td>
<td>Technische Universität Berlin (D)</td>
<td>University of Hull (GB)</td>
</tr>
</tbody>
</table>
## 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>2009-11-04</td>
<td>Final release</td>
</tr>
</tbody>
</table>
# Table of contents

Advancing Traffic Efficiency and Safety through Software Technology phase 2 (ATESST2) ............................................. 1  
Authors .............................................................................................................................................................................. 2  
Revision chart and history log ........................................................................................................................................ 3  
Table of contents .............................................................................................................................................................. 4  
1 Introduction ...................................................................................................................................................................... 5  
2 Safety Analysis Plugin ...................................................................................................................................................... 6  
3 Simulink Plugin ................................................................................................................................................................. 7  
4 Variability Plugin .............................................................................................................................................................. 8  
5 RIF Plugin ........................................................................................................................................................................... 9  
6 AUTOSAR Gateway .......................................................................................................................................................... 11  
7 Timing analysis Gateway ............................................................................................................................................... 12  
8 References......................................................................................................................................................................... 13
1 Introduction

This document provides references to the different plugins that make up the ATESST2 Analysis Platform.

The plugins are tailored for the ATESST2 Workbench, a customized version of the Papyrus UML modeling tool for the EAST-ADL language which can be downloaded at www.papyrusuml.org.

Detailed descriptions of the plugins can be found in the internal plugin deliverables [1-5] available on request.
2 Safety Analysis Plugin

The Safety Analysis Plugin performs safety analysis based on an EAST-ADL error model. It transforms the information available in EAST-ADL to a format that can be processed by HiP-HOPS, which automatically creates FMEA tables, fault trees and minimal cutsets.

![Figure 1: Transformation Design](image)

We leverage the advantages of different model transformation techniques by splitting the translation into two distinct phases and using an appropriate model transformation technique for each phase. Each phase has a distinct purpose and tackles a different concern as depicted in Figure 1. More detailed information on the Safety Analysis Plugin is available in I3.2.3.

The Safety Analysis Plugin is available on:

[http://www.md.kth.se/~biehl/east-adl/safetyanalysisplugin](http://www.md.kth.se/~biehl/east-adl/safetyanalysisplugin)
3 Simulink Plugin

The Simulink Plugin exports modeling elements of the EAST-ADL Functional Analysis Architecture (FAA) to Matlab/Simulink.

As illustrated in figure 2, the plugin uses model-to-model transformations and the Simulink API to create modeling elements in Simulink that correspond to EAST-ADL modeling elements in the FAA. Further details on the plugin can be found in I3.4.3.

The plugin is available on:
http://www.md.kth.se/~biehl/east-adl/simulinkplugin
4 Variability Plugin

The tool support for variability in EAST-ADL was extensively refined in Q1 and Q2 of 2009 according to the review of EAST-ADL variability concepts in WT3.3 in Q4 of 2008. At time of writing the current release of this tool support is 0.5.10.

Compared to the status at the end of ATESSST1, the tool was renamed from “IoVM” to “CVM Framework”, which stands for “Compositional Variability Management Framework”.

This tool support for variability is provided in two parts:

1. **CVM Variability Management Framework:**
   - core CVM with Model Editor, textual VSL Editor, etc.

2. **EAST-ADL Bridge for CVM Organizer:**
   - EAST-ADL related functionality

No. 1 is a stand-alone version of CVM covering the fundamental variability management concepts of EAST-ADL but without relying on the rest of EAST-ADL. No. 2 then integrates CVM into the EAST-ADL language and framework, using the EAST-ADL profile from CEA as a basis.

The plugin is available on the ATESSST2 SVN in folder /WP3/3.3/CVM/

Further information on the plugin is available on [http://www.cvm-framework.org/](http://www.cvm-framework.org/)
5 RIF Plugin

This plugin provides support for exchanging requirements into and out of an EAST-ADL system model using the Requirements Interchange Format (RIF). The intended scope of the plugin includes use cases, an architectural overview with information flows and an informal conception of (model to model) transformation rules to convert RIF data into appropriate EAST-ADL data and vice versa.

![Diagram showing transformation design]

Figure 3: Transformation Design of importing Requirements given by a RIF Document into a given EAST-ADL System Model

As depicted in figure 3, during a RIF import, we expect to already have an EAST-ADL system model. At first, the RIF model file and the EAST-ADL system model file will be deserialized into temporary representations where each conforms to its given meta specification. These are the RIF meta model and the EAST-ADL meta model. Because an EAST-ADL system model already exists, the model to model transformation is a so called inplace transformation. It is inplace, because the given EAST-ADL system model will be refined and enhanced by adding or changing requirements data which will be provided by the given RIF model. Finally, after the transformation has been done, a new EAST-ADL system model is created and serialized including new and changed requirement data. Further details on that point can be found in I3.1.3.
Figure 4 shows the other direction of exchange. Here, requirements data will be exported out of a given EAST-ADL system model. You can see that the export will be done in two steps. The first step is also an inplace model to model transformation. Here, all requirements data that should be exported will be copied into an extra section inside the EAST-ADL system model. Then - the second step – a model to model transformation will be done. Here all requirements data of the newly created section will be transformed into a completely new RIF model. Finally the refined EAST-ADL system model and the new RIF model will be serialized into appropriate files. Further details on that point can also be found in I3.1.3.

Moreover, the domain model of the EAST-ADL language has been adjusted to have a target for each information type of RIF. However, the requirement related information types of the EAST-ADL language have no direct 1:1 mapping to elements in the RIF language. Instead there are still language-specific specializations of requirements (e.g. Technical Requirement, Quality Requirement, Timing Requirement etc.) and tightly couplings to other information types of EAST-ADL (e.g. see ADLSatisfy Relationship Type).

The RIF Plugin implementation is not finalized and cannot be downloaded at the time of writing.
6 AUTOSAR Gateway

In the EDONA project (a French System@tic cluster funded project, http://www.edona.fr/), CEA LIST has provided two specific plugins: an AUTOSAR gateway described here and a bridge to two schedulability analysis tools, MAST and RT-Druid (described in the subsequent section). These plugins result from a collaboration of both ATESST2 and EDONA projects, though the use of both plugins is subjected to special agreement between partners.

The AUTOSAR gateway provides an automated projection from an EAST-ADL design architecture to a preliminary AUTOSAR software component architecture. It has been found that this step in the refinement process of an EAST-ADL modeling project is error-prone, tedious and time consuming. Automating this step provides a tighter integration between EAST-ADL models and process with the AUTOSAR implementation in fine. Two mapping strategies have been developed: 1) Each elementary function is mapped as an independent AUTOSAR software component, embedding a single Runnable, while non-elementary functions are mapped to AUTOSAR composition; 2) Each non-elementary functions are mapped to an independent AUTOSAR software component, embedding as many Runnables as there are elementary functions in the EAST-ADL construct.

The plugin is currently evaluated by CONTINENTAL (Toulouse) as part of the EDONA project validation.
7 Timing analysis Gateway

The timing analysis gateway is developed in the context of the EDONA project [http://www.edona.fr](http://www.edona.fr) by CEA LIST. Its use is subjected to special agreement between partners. It consists in a set of plugins which take into account a MARTE-annotated model of a system and enable to run various schedulability analysis provided by a timing analysis tool. So far the plugins make a bridge to two specific tools:


The model taken as input is a MARTE or EAST-ADL design architecture model, enriched with schedulability analysis parameters provided by MARTE annotations: these make up for a schedulable task and allocation to hardware view of a system. Based on this, the plugin transforms the model into the specific analysis tool format, provides an interface to the tool and returns analysis results. A typical scenario enables to test whether a task allocation scheme allows for a system to be schedulable given hardware performance, or provide parameter values for task priority assignment.

The first version of the plugin is currently evaluated in the EDONA project.
8 References


[5] I3.5.1 Analysis-driven architecture evaluation and optimization. Available on request at www.atesst.org