



Contract number: 2004 - 026976

## Advancing Traffic Efficiency and Safety through Software Technology (ATESST)

**Report type**

**Deliverable D2.1.1**

**Report name**

Evaluation Report of EAST-ADL Requirements Support

**Dissemination level**

PU

**Status**

Final

**Version number**

1.1

**Date of preparation**

2006-01-30

**Authors****Editor**

Matthias Weber

**E-mail**

Matthias.weber@carmeq.com

**Authors**

Lars-Olof Berntsson

Philippe Cuenot

Rolf Johansson

Henrik Lönn

Matthias Weber

**E-mail**

Lars-Olof.Berntsson@volvo.com

Philippe.cuenot@siemens.com

Rolf\_Johansson@mentor.com

Henrik.Lonn@volvo.com

Matthias.weber@carmeq.com

**The Consortium**

Volvo(S)

ETAS (D)

The Royal Institute of Technology (S)

Carmeq GmbH (D)

DaimlerChrysler (D)

Mentor Graphics (Hu)

Technische Universität Berlin (D)

Volvo Cars (S)

Siemens (F)

CEA (F)

Mecel (S)

**Revision chart and history log**

<b>Version</b>	<b>Date</b>	<b>Reason</b>
0.1	2006-10-05	First Draft for review
0.2	2006-12-19	Existing concept of EAST-ADL
0.3	2006-12-22	Evaluation of Support for Timing Requirements
0.4	2006-12-23	Redactional Changes
1.0	2006-01-30	Changed status to Final
1.1	2007-07-04	Updated according to observations/comments from EC review 29/03/2007, Review no. 1 covering project month 1 to 12 (01/01/06 – 31/12/06)

## Executive Summary

The objective of WP2 is to ensure that the EAST-ADL provides adequate development support, ranging from requirements capturing and management, to comprehensive systems analysis (incl. verification and validation), and to final system integration and realization. As an important part of this work-package, WT2.1 aims to further extend the requirements Engineering support of the EAST-ADL and in particular to include linking requirements and tests.

The purpose of this report is to analyse and evaluate the current means to express requirements in the EAST-ADL as defined in the project EAST-EEA [2]. This also includes evaluating the current means of related approaches such as SysML.

**Table of contents**

Advancing Traffic Efficiency and Safety through Software Technology (ATESST).....	1
Authors .....	2
Revision chart and history log.....	3
List of abbreviations.....	<b>Error! Bookmark not defined.</b>
Executive Summary.....	4
Table of contents .....	5
List of Figures .....	<b>Error! Bookmark not defined.</b>
List of Tables .....	<b>Error! Bookmark not defined.</b>
1 Contribution to overall ATESST objectives.....	6
2 Summary of Existing Concepts in the EAST-ADL .....	7
3 The relation between EAST-ADL and Requirements Management (RM) / and Quality Assurance (QA) .....	10
4 Recent Developments in SysML Requirements Concepts.....	11
5 Information Exchange with RM/QA tools.....	12
6 Analysis and Evaluation of EAST-ADL Support for Timing Requirements.....	13
7 Conclusions .....	15
8 References .....	16

## **1 Contribution to overall ATESST objectives**

Requirements are the input information for any system design effort, and contains the constraints and rules during the entire design effort. To improve dependability and in particular safety, rigorous treatment of requirements is critical. An architecture description language captures the elements that compose the modelled system architecture in a comprehensive way. It is therefore desirable to also investigate requirements, and how they can be associated to the system elements. An evaluation of the support for requirements in EAST ADL is therefore an important input for the further refinement of requirement constructs.

## 2 Summary of Existing Concepts in the EAST-ADL

The existing Requirements concepts, developed in the EAST-EEA project, is part of the cross-cutting concepts of EAST-ADL.

Requirements are implemented on multiple or different abstraction level, meaning that they could be introduced at the different stages of an automotive engineering process supported by the EAST-ADL meta-model with modeling entities. Requirements could be expressed by different engineers from different disciplines (marketing department, system engineer, control engineer, software engineer, driver/OS developers ...) at the EAST-ADL level fitting to their activity domains. They could be then detailed or derived to express dedicated concerns, embedded in a hierarchical requirement structure.

Concept supports modeling of either textual descriptions or using more formal descriptions, and allows specifying required properties of the system, at the varying degree of abstraction. This concept is integrated into the EAST-ADL language to trace requirements between system refinement and system decomposition levels, and complete requirement satisfaction for system component. In addition it is also link to verification and validation activities and artifacts in order to verify requirement.

This concept is tightly linked to the basic SysML requirements concepts as defined during 2004 (Draft adopted specification ptc/2006-05-03). Since this date the OMG adopted final specification has been released in August 2006 (<http://www.sysml.org>), and requirement concept shall be revisited to be adjusted and evaluated toward the new specification. Typically, a requirement modeled on class-level is unique, and shall be marked with a unique ID as attribute; hierarchical organization and association between requirements shall inherit from SysML association (see chap 5).

In EAST-ADL 1.0, the basic and general concept for requirement is the "requirement" element, where specific requirement types will inherit from this element and will have their own attributes and relations. Requirement entity could be classified in different type as described below.

Specialized requirements with general scope are:

- EFeature : Entity used to describe required functionality and include relation to variability mechanism (Variant and VariationPoint)
- AllocationConstraint : Requirement used to constraint allocation of Efeature (AnalysisFunction or CompositeSoftwareFunction) in a target ECU
- DesignConstraint : Requirement used to constraint the solution
- Description : Provide a reference to a requirement described in specific formalism
- FunctionalRequirement :Entity used to refine "what" shall do an EFeature
- Specify : used for relationship between EFeature and FunctionalRequirement
- QualityRequirement : Used to express external properties on the system (HMI, Performance...)
- QualityRequirementType : Enumeration for Quality Requirement
- Constraint : relationship between an EFeature and Design Constraint
- DesignConstraintType : Enumeration for DesignConstraint
- InteractionType : Used for association between EFeatures or between an EFeature and an actor

They are several concepts, as described below, that need to be improved in order to restructure requirement clustering and better fit with an engineering process.

EFeatures are seen as a specialization of requirement, it was an initial pragmatic approach, but conceptually, they are more related to vehicle view than a requirement entity.

The requirement concept is mainly centered on EFeature (at least in the initial metamodel available). All relationship between requirements objects and ADL modeling entities shall be extended and redefined in order to allow link of requirement anywhere on the EAST-ADL entities.

Specifications in arbitrary formalisms, e.g. use-cases, state charts, block diagrams, scenario can be attached to any kind requirements, in order to refine the requirement with details description, or be considered as primary requirement. At present time only interaction with limited capabilities (and missing semantic in the metamodel) allows this concept at EFeature level. This has to be extended and expressed for the overall requirement entities.

Some enumeration types for attributes are explicitly specified (e.g. types of design constraints). For some enumeration types, there is an informal indication of the possibility of project specific adaptation (e.g. status attribute); this needs to be stated in a more formal way.

Specialized requirements with time relation:

- TimingRestriction : Entity used to give bound on timing system attribute, specialized in EndToEndDelay or PeriodRestriction
- EndToEndDelay : Entity used to constraint delay between FunctionPort
- PeriodRestriction : Entity used to restrict allowed Period for FunctionPort or LogicalCluster
- TimeBase : Group entities that share common time base
- Bound : used for bound timing measure to refine timing requirement entity
- Unit : Unit for a bound value specification

Timing requirement evaluation and potential improvement will be worked out further in the document (chap7).

Specialized requirements for Verification and Validation purpose:

- VerificationMethod : Used to require specific method for requirement validation
- VerificationMethodType : Enumeration of VerificationMethod

Validation methods and concepts are sketched out the document in deliverable D1.1.2, to link the requirement and test artifacts.

In addition some global items need to be revised for a more integrated structure with clear role of each attributes; but also in the direction of nesting and grouping of requirement. Requirement shall be depicted as a real cross-cutting concept enabling requirement tree organization for sound definition of requirement dependency and relationship (based on new SySML concept), but also supporting the different abstraction view for introduction of discipline specific requirements. Furthermore full traceability concept between test, requirement and ADL entities is mandatory to guarantee automotive product quality, and being able to track progress of the requirement during the project development.

The traceability concept shall also be the base for relation of requirement with variability mechanism, including upward and backward traceability to support the reuse concept starting from requirement entity. This concept is highly motivated by automotive domain specific constraint where productivity and cost reduction is mainly driven by reuse and product line concept. So relationship between variability and requirement shall be improved with transversal mechanism and specialized association dedicated to this purpose.

Last but not least, focusing to the scope of the project call on safety automotive system, Safety Requirement must be stated and introduced in EAST-ADL. Starting from safety requirement

perceptive, concept shall allow safety assessment of vehicle feature and define additional attributes expressing relevant properties of dependable systems, such as severity, frequency, exposition and controllability in relation to hazardous event. The new global concept shall allow description of hazard scenario as part of requirement associated to safety requirement, to finally support analysis and risk assessment to fulfill the integrity level expected from the system or sub-system. This implies of course to support relation to both hardware and software entities to support a seamless safety process with EAST-ADL.

### **3 The relation between EAST-ADL and Requirements Management (RM) / and Quality Assurance (QA)**

In the automotive industry, both activities (RM and QA) extend far beyond software development, comprising the entire lifecycle of the complete vehicle.

The multitude of process contexts and domains has led to concepts and tool support (e.g. DOORS, QualityCenter) that expose some infrastructure ruptures with respect to a "model-based" approach such as the EAST-ADL.

- RM/QA tools are used for a wide range of activities in requirements engineering (risk analysis, cost calculation, test planning, test execution, test evaluation, ....). The EAST ADL has to decide which of these information to represent in the language.
- RM/QA tools have very flexible configuration possibilities, including metamodel adaptations. On the one hand, there exist well-known public and company internal templates for e.g. requirements, but development projects often end up using an extended subset of these templates. In contrast, the EAST-ADL currently has a single fixed meta-model, which is not surprising considering the goal of standardization especially on the Design and implementation level. However, this seems not to be flexible enough to represent RM/QA information in the EAST-ADL.
- RM/QA tools are tightly coupled to databases and administer information elements (including history, user rights, versions etc.) on a fine-grained basis (e.g. individual requirements or groups of requirements). Tools for model-based development (e.g. ML/SL) are file-based and administer on the level of entire models. The EAST-ADL has reflected this by introducing "infrastructure concepts" (e.g. versioning, history) and allows to require these on the level of individual elements (e.g. requirements). However, while this is conceptually clean, it does not really solve the practical problem of different tool infrastructures.

## 4 Recent Developments in SysML Requirements Concepts

Version 1.0 of SysML [1] has introduced a number of changes and refinements with respect to previous versions. The following ones which are of special interest for the EAST-ADL are briefly presented and discussed

- Decomposition of requirements
  - o It is semantically not clear what the Decomposition means, i.e. is it more than the sum of its parts? Methodologically, decomposed requirements introduce problematic redundancies, e.g. how are the attribute values for status and maturity related between a composite requirement and its parts, similarly, does a “test-by” association from a composite imply similar associations from all its parts?
- Multiple copies of requirements with read-only text attribute (to support product families)
  - o This is something for WP4 to consider.
- “derive” relation between requirements
  - o This should be adopted by the EAST-ADL.
- “satisfy” relation between requirements and model elements
  - o This should be adopted by the EAST-ADL. However, typically the whole – or at least a large part of the - system model must be considered to really “satisfy” a requirement. Hence the practicability of this attribute is not quite clear.
- “verifies” relationship between requirement and test case
  - o This should be adopted by the EAST-ADL.
- “refine” relationship to further specify a requirement by a suitable model element
  - o (to be discussed) This could be adopted for native state machines?
- “trace” relationship between requirement and any model element
  - o This should be adopted by the EAST-ADL.

Furthermore, SysML defines some non-normative requirements types (which restrict the satisfy relation)

- Functional requirements (satisfied by operation or behavior)
- Interface Requirements (satisfied by port, connector, item flow, or constraint property)
- Performance Requirements (satisfied by a value property)
- Physical Requirement (satisfied by a physical element)
- Design Constraint (satisfied by a block or part)

The “old” EAST-ADL is supporting these concepts (except for interface requirements). See also the above remarks about the “satisfy” relation.

Furthermore, there are some general new mechanisms in SysML which are worth considering in the EAST-ADL, however, they are generally applicable

- Rationale objects: special objects to be attached to any object in the EAST-ADL
- tabular views of compositions: could also be used for composite SW functions
- package views: could be used for any diagram in the EAST-ADL

## 5 Information Exchange with RM/QA tools

The EAST-ADL aims to support information exchange scenarios with requirements and quality assurance tools. These tools support a fine-grained administration of elements. EAST-ADL does not intend not replace the concepts in these tools but provide a usable synchronisation of information from these tools to its “model-based” approach.

The minimum for a usable synchronisation is to map elements from external tools to the EAST-ADL meta-model. Given the EAST-ADL as it is, this might imply means that some information will be collapsed, making it hard to transfer back changes. However, at least the following scenario should be supported:

- (in a roundtrip manner): To create and edit requirements and test cases (and links) in the EAST-ADL and export this information to RM and QA tools

If the EAST-ADL is supporting meta-model adaptations, then these can be used to represent information outside the basic EAST-ADL elements. Therefore the following two scenarios should be supported:

- (in a roundtrip manner): To create and edit requirements and test cases (and links) in the EAST-ADL and export this information to RM and QA tools
- (in a roundtrip manner): To import information form RM and QA tools, and, where allowed, modify it in the EAST-ADL, and export this information to RM and QA tools

## 6 Analysis and Evaluation of EAST-ADL Support for Timing Requirements

To be able to evaluate the current means to express timing requirements in EAST-ADL, the first thing is to identify what timing requirements that preferably should be possible to express. One of the goals for the ATESSST project is to be compliant with AUTOSAR. Among the released AUTOSAR documents the VFB Specification [3], includes some timing. However there has also been a timing team that although they haven't produced any publicly released AUTOSAR document, there are two working documents where, among other things, requirements of what a timing model preferably should express are discussed. [4] and [5] have been input for the requirements on what EAST-ADL should support concerning timing requirements.

In EAST-ADL1 [2], there is a requirement entity called `TimingRestriction` that allows giving bounds on system timing attributes. There are two specialisations of `TimingRestriction` that is defined in [2]: `EndToEndDelay` and `PeriodRestriction`. These are each specializations of `TimingRestriction` and `QualityRequirement`. `PeriodRestriction` can be allowed to restrict allowed periods for e.g. `Ports`, `ElementarySoftwareFunctions` and `LogicalClusters`. `EndToEndDelay` can be used to constrain the delay between the arrival of a signal in a port and the writing of a signal in another port. The `EndToEndDelay` can only reference one source port and one destination port for its timing definition. It is possible, however, to link several `EndToEndDelays` to the same `TimeBase`. This way, start events and offsets are constrained to the same time base, i.e. the events are required to be synchronous.

In the timing model of EAST-ADL1, there are also the attributes `ExecutionTime` and `Period`. These are not requirement entities, but can anyhow be seen as a part of the EAST-ADL timing meta model. `ExecutionTime` represents the execution time of a `Function`, while the `Period` is the interval between function invocations. There are no other implementation identities in [2], expressing timing.

In AUTOSAR VFB specification [3] there a defined a number of timing attributes of communication patterns. In general attributes in AUTOSAR can be either model-attributes or implementation attributes. The former specify the intention of the developer as they are assigned to SW component descriptions. They are also independent of deployment scenario such as network topology or SWC allocation to ECU platform. The latter defines how a specific implementation is realized and depend on platform, bus etc.

For sender-receiver communication the set of model attributes on a connector includes `TRANSFER_TIME` and `JITTER`. `TRANSFER_TIME` specifies an end-to-end requirement on how much time it may take to transfer data from the sending component to the receiving component. Its value is either of three: `deadline` (max age), `exact_transfer_time` (must be defined together with `JITTER` attribute, or no time constraint set. The `JITTER` attribute is either a maximum value, or no constraint set. For both these attributes are defined as being on AUTOSAR meta level I (Instance and not Type), abstraction level M (Model and not Implementation), and abstraction type R (Requirement and not Behaviour).

The timing model of EAST-ADL1 as can be seen to cover what is needed for sender-receiver communication in AUTOSAR as defined in [3]. However, the problems described by the AUTOSAR timing team in [4] and [5], implies that an ADL must be capable of expressing more than this. Some of what the AR timing team pointed out, they have no solution for in the context of AUTOSAR. One reason for this might be related to the possibility to express timing requirements (and behaviour) on a relevant level of abstraction. If the VFB level of AUTOSAR can be seen as on design level in EAST, and the System level and Implementation level respectively can be seen as on the implementation level in EAST, there is obviously a need for an EAST analysis level. On this level it should be possible to describe detailed and complicated timing requirements and properties with respect to algorithmic behaviour and its realization. This implies that at least at the analysis level, EAST ADL should be able to express thing like synchronization, where there is no

single timing chain that should have a certain timing property. In general synchronization is dealing with the relation in time between two or more nodes that are related by fork and/or join of a timing chain, but not following after each other on the same chain. Special cases of this, is input synchronization and output synchronization. Input synchronization describes the difference in time between a number of inputs that eventually joins to the same timing chain. Similarly, output synchronization concerns the difference in time between a number of outputs following a fork of a timing chain. In general, this is not possible to express in EAST-ASL1.

Concerning end-to-end deadlines, EAST-ADL should not only be able to analyze a single timing chain, but also redundant chains realising the same end-to-end timing requirement should be able to handle. This is not generally the case in EAST-ADL1.

In this document is mainly considered the modelling of requirements. However, when talking about timing, the entire timing modelling should cover both requirements and properties.

.

## 7 Conclusions

Requirements are critical elements in a system model, in particular when safety is addressed, as in the ATESSST project. For that reason the evaluation of existing requirement approaches is important. Based on what is found in EAST ADL1, there are several areas that need further refinement in EAST ADL2. New SysML concepts that evolved since EAST ADL1 need to be included and extended. Timing constructs need to be amended to cover more specification and analysis scenarios. The RIF standard has emerged since EAST ADL and should be covered. Means for integration with dedicated requirements engineering tools should be specified. Overall, the basic mechanisms are there, and an ADL is the ideal framework for requirements engineering and the expressiveness it needs.

## 8 References

- [1] [www.sysml.org](http://www.sysml.org)
- [2] EAST-ADL version 1.0
- [3] AUTOSAR Specification of the Virtual Functional Bus, Version 1.0.7, 2005-05-04 (part of AUTOSAR Release 1.0.0)
- [4] AUTOSAR Timing Meta Model, Version 0.95, 2005-06-28 (internal draft document)
- [5] AUTOSAR Timing Model SPS (Software Problem Summary), Version 0.1, 2005-06-10 (internal draft document)