



Contract number: 2004 - 026976

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

**Report type**  
**Report name**

**Deliverable D2.1.2**  
**Linking Requirements and Test Artifacts**

**Dissemination level**

**PU**

**Status**

**Final**

**Version number**

**1.1**

**Date of preparation**

**2007-01-30**

**Authors****Editor**

Matthias Weber

**E-mail**

Matthias.weber@carmeq.com

**Authors**

Lars-Olof Berntsson

Henrik Lonn

Matthias Weber

**E-mail**

Lars-olof.berntsson@volvo.com

Henrik.lonn@volvo.com

Matthias.weber@carmeq.com

**The Consortium**

Volvo (S)

ETAS (D)

Kungliga Tekniska Högskolan (S)

Carmeq (D)

DaimlerChrysler (D)

MGH (Hu)

Technische Universität Berlin (D)

Volvo Cars (S)

SiemensVDO (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	2006-12-18	Second Draft
0.21	2006-12-21	Third Draft
0.3	2006-12-21	Small editorial corrections
1.0	2007-01-30	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 propose a concept to link requirements and test artifacts in the EAST-ADL.

## Table of contents

Advancing Traffic Efficiency and Safety through Software Technology (ATESST).....	1
Authors .....	2
Revision chart and history log.....	3
Executive Summary.....	4
Table of contents .....	5
List of Figures .....	6
List of Tables .....	<b>Error! Bookmark not defined.</b>
1 Input from previous ATESST work .....	7
2 Contribution to overall ATESST objectives.....	8
3 Language Scope for Linking of Requirements and Test artifacts.....	9
4 Existing Approaches .....	10
4.1 Linking of Requirements and Testing in EAST-ADL 1.0.....	10
4.2 OMG UML2 Testing Profile.....	10
5 Basic Approach followed in ATESST.....	12
5.1 Basic requirements relations .....	12
5.2 Requirements types.....	14
5.3 Safety-related requirements .....	14
5.4 Timing requirements.....	14
5.5 Explicit modeling of verification and validation (V&V) artifacts .....	15
6 Current Specifications.....	16
6.1 Overview.....	16
7 Conclusions .....	19
8 References .....	20

**List of Figures**

Figure 1: Linking Requirements and Testing in EAST-ADL1 .....	10
Figure 2: Main concepts of U2TP .....	11
Figure 3: The basic requirements relations <i>refine</i> , <i>verify</i> , <i>derive</i> , and <i>satisfy</i> . (adopted from SysML).....	12
Figure 4: The figure illustrates requirements tracing (derive) and linking to system components (satisfy). .....	13
Figure 5: The figure illustrates the refinement relation.....	14
Figure 6: The figure illustrates the verification of a requirement. ....	15
Figure 7: Overview of the Verification and Validation Concept of the EAST-ADL .....	16
Figure 8. The ADLVerify association .....	18

**1 Input from previous ATESST work**

Evaluation report D2.1.1

## **2 Contribution to overall ATESST objectives**

Verification activities are critical elements of system development. The information they rely on, and the information that is produced are well-defined among the system development artifacts. It is also important to keep track of the exact verification setting that is used. Test and verification artifacts should therefore be part of the ADL, and is an important contribution to the correctness of the developed system.

### 3 Language Scope for Linking of Requirements and Test artifacts

Verification and validation activities are becoming increasingly important in the automotive domain due to the increased impact of embedded electronics. For that purpose, the EAST ADL2 shall contain modeling support for methods and techniques to verify and validate embedded electronics and to link this information to requirements.

In the area of embedded software-intensive systems, integration and testing typically makes up about 50% of the overall development effort. Testing artefacts contain a huge amount of information including all the details of test definition, test (tool) environments, and test scripting languages, test execution, and test evaluation.

It is not the goal of the EAST-ADL to present all the information related to testing, but rather to offer means to model key aspects of the validation state of EAST-ADL requirements and to offer means to link these requirements with respective testing and validation information.

Inside the ATESSST consortium, the following requirements have been agreed upon, concerning the information to be represented in the ADL related to the validation state of a requirement.

- *Which tests (or which parts of a test) are related to the requirement*

In more detail, the following information about these tests should be represented

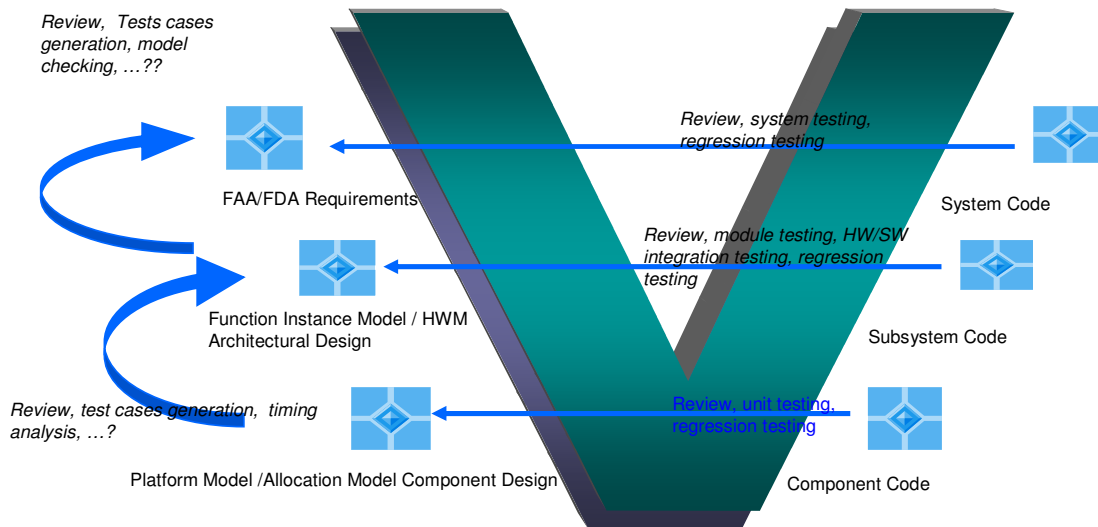
- *What is the general behaviour of the test (textual description, model)*
- *In which test environments, and on which (EAST-ADL) abstraction levels are the concrete tests.*
- *Which inputs are used for the test*
- *What is the status of the test*
- *Which outputs have been obtained by the test*
- *What are the criteria for test success*
- *Further process-specific information about the test (e.g. process milestones reached, people & roles responsible)*

## 4 Existing Approaches

### 4.1 Linking of Requirements and Testing in EAST-ADL 1.0

There are several V&V techniques, methods and tools applied during the design of embedded systems. Their use depends on the design stage and on the properties to validate and / or verify. Furthermore, each partner of a project develops specific V&V processes and activities. So it is impossible to introduce in EAST-ADL a way for the modeling of all the objects that can be used by all the V&V techniques. As a conclusion, EAST-ADL furnishes just the means for describing the links between V&V activities. On the one hand, the “requirement” objects that need one or several V&V activities and on the other hand, the objects modeling the system (Functional Analysis Architecture, Functional components, ..., Logical Tasks, ...). Furthermore, the common parts of all V&V techniques have to be described in EAST-ADL: which results are expected from the V&V activities, which results are obtained by applying the V&V techniques, how the V&V activities are constrained. What is specific to a V&V technique is not described in EAST-ADL.

Verification and validation techniques must be used in one or several steps of general verification and validation process. Some of them are specific to one model of the ADL, others can be applied at several layers, but they are all necessary to verify and validate completely the system.



**Figure 1: Linking Requirements and Testing in EAST-ADL1**

Hence this is a mapping picture between verification and validation process and EAST-ADL hierarchy (Figure 1).

### 4.2 OMG UML2 Testing Profile

The new edition of the Unified Modeling Language (UML2) - a widespread visual modeling language that is standardized by the Object Management Group (OMG) - is a common ground for system and test modeling. Its generic extension mechanism supporting different UML profiles for different purposes and domains is an ideal basis for better integration of test and system development. Since UML2 still suffers from missing concepts for describing test models efficiently,



## 5 Basic Approach followed in ATESSST

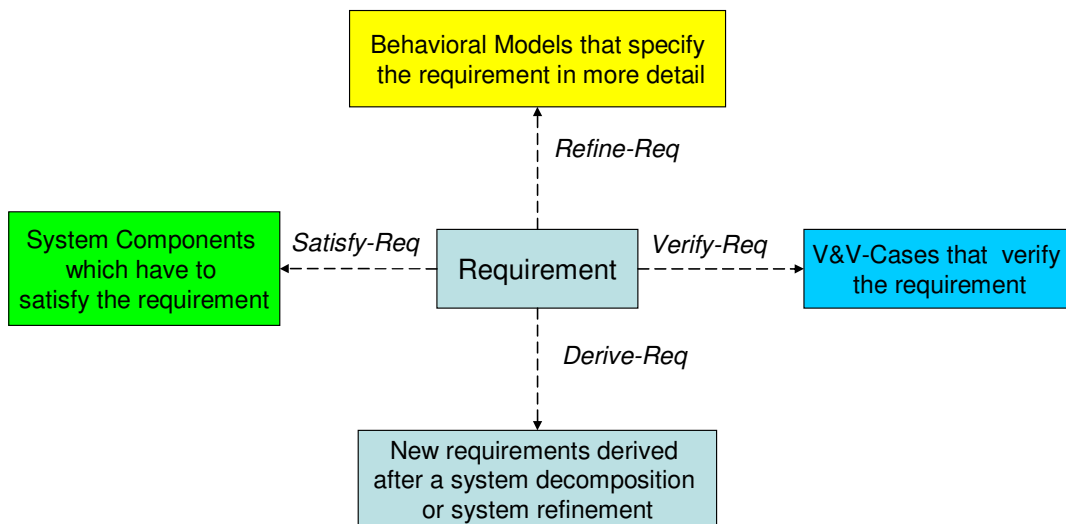
In order to better support the development of dependable systems, the EAST-ADL does not only include means to create analysis and design models of the system to be developed (at varying abstraction levels), but also language means

- to specify required properties of the system (at varying degrees of abstraction),
- to trace requirements between system refinement and system decomposition levels
- to require satisfaction of requirements for system components,
- to refine the specification of requirements by behavioral models
- to verify requirements by verification and validation activities.

The EAST-ADL does not start from scratch but closely aligns its requirements concepts to SysML 1.1 [1], as currently standardized by the OMG organization. However, extensions and adjustments are made to these proposals based on the needs of the automotive application domain.

### 5.1 Basic requirements relations

Four requirement relations from SysML are used in the EAST ADL, Figure 3.



**Figure 3:** The basic requirements relations *refine*, *verify*, *derive*, and *satisfy*. (adopted from SysML).

First and foremost, requirements may be used to textually specify required properties of the system to be developed. The textual specification of a requirement can be refined (using the “refine” relationship from SysML) by attaching behavioral models – such as use-cases, activity diagrams or state machines - to requirements.

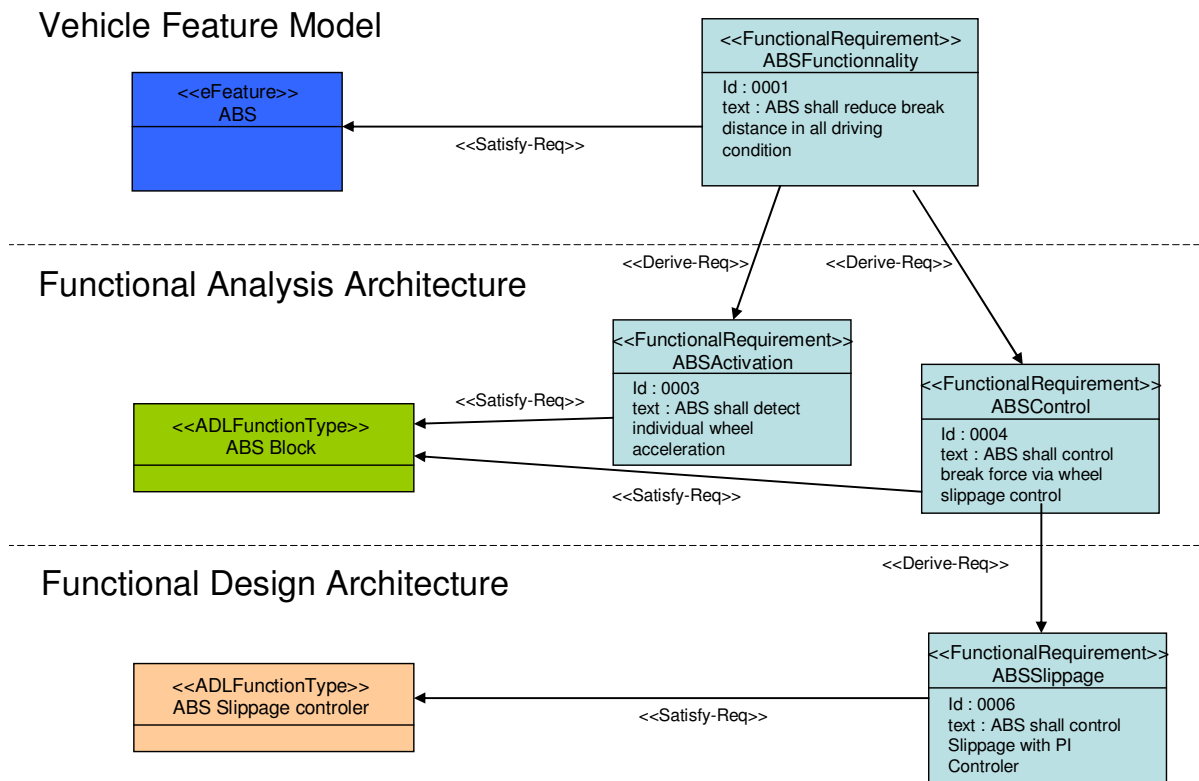
Requirements are refined into more detailed requirements after a system refinement or a system decomposition step. This concept is supported by the “derived requirement” relationship to allow a hierarchical view of requirements to be defined. It allows analysis of requirements to determine multiple derived requirements that support a source of requirement, and the document requirement over the system decomposition, but also over the different abstraction levels of feature development from different disciplines. Conversely, it allows tracing detailed requirements on a low abstraction level, to the overall requirements that motivates them.

Specific UML constructs such as use case or activity diagram assist requirement analysis for better description or prepare further refinement. These constructs are associated to requirement entities via a “refine” construct, as mentioned above.

Requirements apply to the various system components which are introduced to satisfy them. This concept is modeled using the “satisfy” relation from SysML. Note that this relation does not indicate that the referenced entity really satisfies the requirement, only that it should do it.

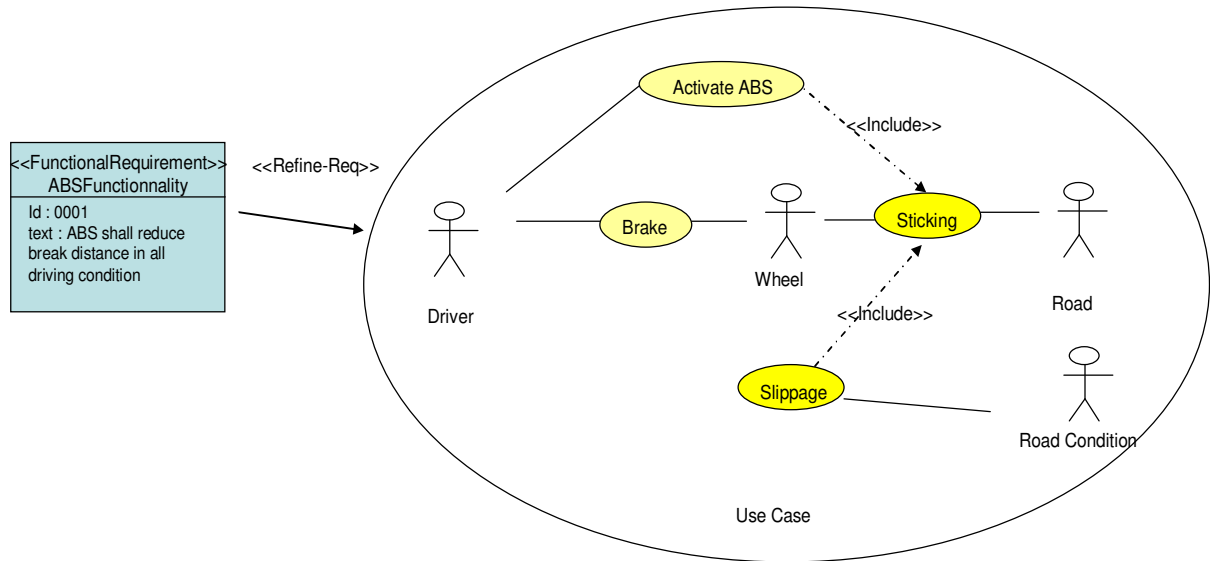
Requirements traceability is completed by the verification and validation relationships. These defines how the verification and validation activities, such as the testing activities fit criteria explicit for the verification and validation goals and how its associated verification and validation cases, such as test cases, verify the requirement. This is modeled using the SysML “verifies” relationship along with constructs that capture and relate V&V to the requirement and system components.

The “satisfy” and “derive” relations are illustrated in Figure 4.



**Figure 4:** The figure illustrates requirements tracing (derive) and linking to system components (satisfy).

The refinement of a requirement by means of a use case is illustrated in Figure 5.



**Figure 5:** The figure illustrates the refinement relation.

## 5.2 Requirements types

Methodically, EAST ADL differentiates between *functional requirements*, which typically focus on some part of the “normal” functionality that the system has to provide (e.g. “ABS shall control brake force via wheel slip control”), and *quality requirements*, which typically focus on some external property of the system seen as a whole (e.g. “ABS shall have an MTTF of 10.000 hours”).

Quality requirements are further classified; two examples, “safety requirements” and “timing requirements” are discussed in more detail below.

## 5.3 Safety-related requirements

In order to perform safety assessments of the vehicle systems, safety requirements in the refined EAST-ADL have attributes and related entities to define the requirement and the hazard it mitigates. Hazards or hazardous events are part of the environment model and are characterized by attributes for severity, exposure and controllability [4]. The hazardous event may be further detailed by e.g. use cases, sequence or activity diagrams.

Safety requirement attributes includes safety integrity level (SIL), operation state, fault time span, emergency operation times, safety state, and functional redundancy to record dependability characteristics [4]. A requirement can be traced from the abstract vehicle model all the way to its derived requirements allocated to the final hardware and software components. Depending on abstraction level, some or all of these attributes are applicable.

## 5.4 Timing requirements

Embedded systems have several timing requirements. On the top level, there are performance requirements based on e.g. ergonomics or safety. To meet such top level timing requirements, or to sustain the selected design regarding resource scheduling or interaction between components, timing requirements can be seen on all abstraction levels of an automotive system. Timing errors

are the source of many failures, and it is thus important to correctly express and subsequently analyze timing properties. Examples of timing requirements that are supported by the EAST-ADL include end-to-end deadlines, period timing and worst case execution time. The goal is to be able to support the analysis techniques necessary for high integrity automotive systems.

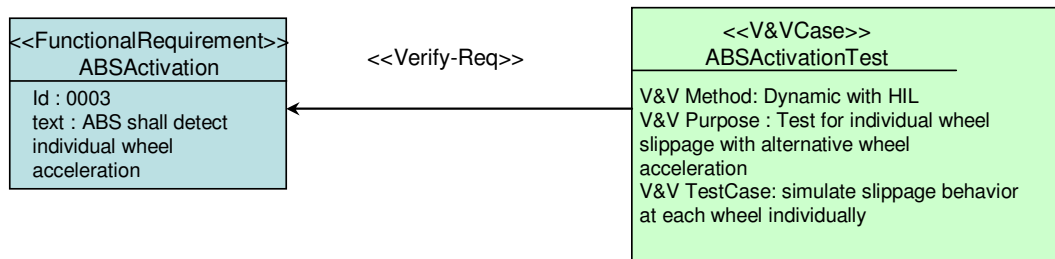
**5.5 Explicit modeling of verification and validation (V&V) artifacts**

In order to support the development of dependable systems, the EAST-ADL offers detailed means to explicitly model central artifacts of verification and validation activities and to relate these artifacts to requirements. This allows for explicitly and continuously planning, tracking, updating and managing important V&V-activities and their impact on the system in parallel to the development of the system.

The combination of a V&V-case, its environment and its target object is described as a V&V context. A V&V-case will take very different forms, depending on the kind of V&V activity performed, e.g. safety analysis, specification, design or implementation review, functional analysis by simulation, SIL-testing, HIL-testing, or vehicle testing. In general it consists of a number of V&V-procedures to be applied to the target object. Each procedure may contribute a dedicated aspect in the verification of some requirement and the EAST-ADL allows documenting this relationship by means of the “verifies” relation.

Of course, in addition to these basic structures all the relevant testing data must recorded in order to support analysis and automaton in regression testing.

The basic association of a V&V-case to a requirement is illustrated in Figure 6.



**Figure 6: The figure illustrates the verification of a requirement.**



VVContext specifies an instance of a V&V setting by identifying all entities that are relevant for a V&V activity in a specific case. Its vvSubject identifies the system entities that are subject to verification or validation; its vvEnvironment identifies the environment; Its VVCases identifies the V&V cases; VVStimuli, VVIntendenOutcome and VVActualOutcome identify the data for each VVProcedure in each VVcase. vvTarget, finally, identifies the realization of the V&V subject, i.e. the code, model, implementation, etc. that is the actual target of the VVProcedures.

### **VVTarget**

VVTarget is the realization of the entities that are subject to V&V (vvSubject). The VVTarget may be software, hardware or systems that are appropriate for the intended V&V activities, as specified by the VVProcedures. VVTarget may be tailored for the V&V activity, or be the regular realization of the V&V subject. By defining the VVTarget explicitly, the V&V engineers can decide whether the V&V setting is sufficient for the V&V subject and its requirements.

### **VVCase**

VVCase is the definition of a V&V case, such as a test case. It identifies the operations, VVProcedures that compose the V&V case. An external association, ADLVerify, identifies which requirements are verified by the VVCase. VVStimuli, VVIntendenOutcome and VVActualOutcome contains the input and output data of the VVProcedures that are part of the VVCase.

### **VVProcedure**

VVProcedure is the detailed specification of a V&V operation. It assumes that input and output data is defined in an appropriate format in the VVStimuli, VVIntendenOutcome and VVActualOutcome entities.

### **VVIntendedOutcome**

VVIntendedOutcome is the data that represents the intended outcome of a test or other V&V procedure. The syntax and semantics is tool specific but must comply with the needs of the VVProcedure.

### **VVStimuli**

VVStimuli is the input data for a certain VVProcedure. Its format is tool and methodology specific, but should comply with the VVProcedure that uses it.

### **VVIntendenOutcome**

VVIntendenOutcome is the expected output data for a certain VVProcedure. Its format is tool and methodology specific, but should comply with the VVProcedure that uses it.

### **VVActualOutcome**

VVActualOutcome is the actual output data for a certain VVProcedure. Its format is tool and methodology specific, but should comply with the VVProcedure that uses it.

### **ADLSatisfy**

ADLSatisfy is a dependency relationship between a requirement and a model element that shall fulfill the requirement.

### **ADLVerify**

ADLVerify is a dependency relationship between a requirement and V&V Procedures and Cases that verify it. Figure 8 shows the concept in more detail.

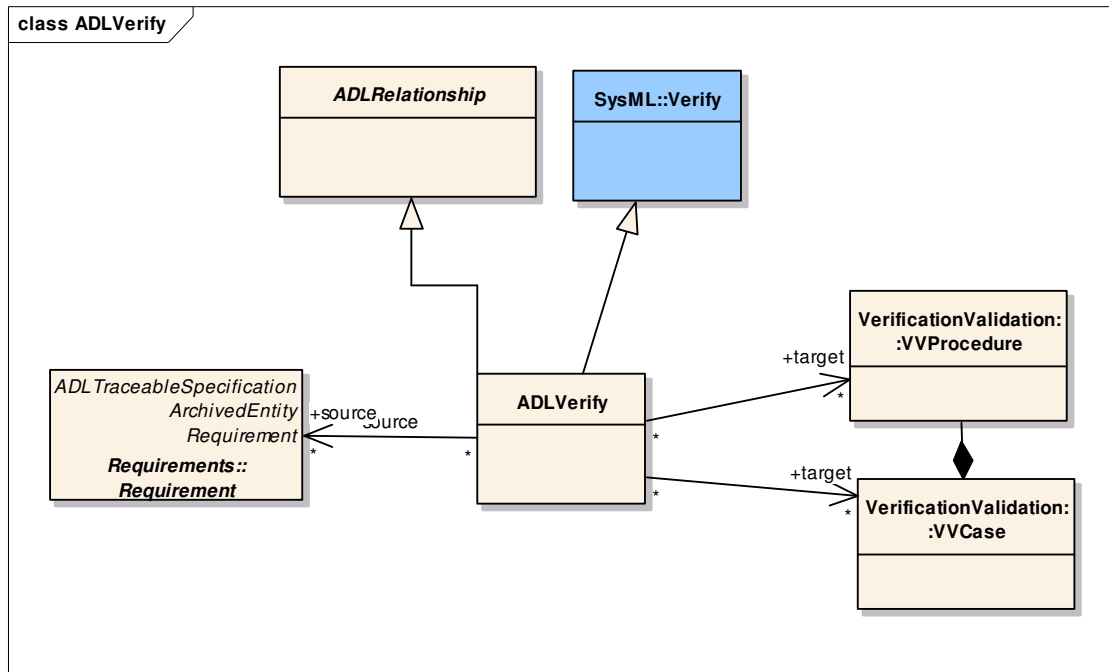


Figure 8. The ADLVerify association

## 7 Conclusions

This report has summarized the EAST ADL2 concepts for the modeling of Verification and Validation artifacts. Because of the importance of V&V for the dependability of the EE system, dedicated support for V&V related information is important. Moreover, the complexity of V&V information and its close relation to the system specification it operates on, it is particularly suitable for being included in the ADL. Compared to EAST ADL1, the V&V constructs are now more suitable for re-use, as the context of the V&V procedures and data is clearly defined. Also, the definition of which entities are part of the V&V model is more clearly specified.

Currently, the V&V model is completely generic. Specializations of the metaclasses for specific V&V methods and tools are possible, but not part of the current report.

## 8 References

- [1] [www.sysml.org](http://www.sysml.org)
- [2] EAST-ADL version 1.0
- [3] OMG UML2 Testing Profile: [www.omg.org/cgi-bin/doc?ptc/2004-04-02](http://www.omg.org/cgi-bin/doc?ptc/2004-04-02)
- [4] ISO CD 26262 (planned for end 2007)