Synligare utveckling av inbyggda fordonssystem – visualiserad kravhantering och samverkan

Visibler Development of Embedded Automotive Systems -Visualised Requirements Management and Collaboration



Report type Report name

Deliverable D4.1

Report on Validator System and Validation Results

Dissemination level	Public
Status	Final Release
Version number	2.0
Date of preparation	2016-05-20

D4.1

Authors

Mattias Ekberg

Urban Ingelsson

Ali Shahrokni

Editor	E-mail
Henrik Lönn	Henrik.lonn@volvo.com
Authors	E-mail
Henrik Kaijser	henrik.kaijser@volvo.com
Henrik Lönn	henrik.lonn@volvo.com
Bengt-Arne Nyman	bengt-arne.nyman@autoliv.com

henrik.lonn@volvo.com bengt-arne.nyman@autoliv.com mattias.ekberg@arccore.com ali.shahrokni@systemite.se urban.lngelsson@semcon.com

The Consortium

Volvo Technology Corporation AB - Autoliv AB - Arccore AB- Systemite AB - Semcon AB

Table of contents

Aut	thors.			3
Tal	ole of	conte	nts	5
1	Intro	oducti	on	7
2	Bac	kgrou	nd	8
2	2.1	Proje	ect Objectives	8
2	2.2	Mea	ns	8
3	Vali	datior	of Goals Fulfillment	9
3	3.1	Metr	ics	9
3	3.2	View	/s	9
3	3.3	Prec	lictability	10
3	3.4	Qua	litative improvement: Fewer Misunderstandings	11
3	3.5	Qua	ntitative improvement: Shorter Development Time	11
4	Key	Syste	em Technologies	12
5	Vali	dator	System	21
Ę	5.1	Adju	stable Speed Limit	21
6	Vali	datior	Scenario	24
6	6.1	Ove	rview	24
6	5.2	Core	e Scenario	25
	6.2.	1	OEM Views on Original System	25
	6.2.	2	OEM Metrics on Original System	32
	6.2.	3	OEM – Tier1 iteration	35
6	5.3	Dep	endability Scenario	40
	6.3.	1	Hazard and Risk Analysis	40
	6.3.	2	Functional Safety Concept	44
	6.3.	3	Technical Safety Concept	46
	6.3.	4	Tier1 Technical SafetyConcept	48
7	Sun	nmary	and Conclusions	49
7	7.1	Refle	ection	49
7	7.2	Con	clusions	50
8	Ref	erenc	es	51

1 Introduction

This document reports on the validation of project results performed in Synligare project. Needs and concepts from Workpackage 1 and 2 has resulted in tooling prototypes in Work package 3. These tools have been applied to example systems identified and refined in Work package 4. During example modeling, tools and methods proposed and developed in the project have been used, refined and assessed.

The report describes project objectives in Chapter 2 and a description on how project goals were met in 3. Some of the Technologies needed to meet the goals were characterized in Chapter 4. The example system used to validate tooling and concepts are described in Chapter 5. Chapter 6 summarizes the validation activity using sample diagrams and screenshots from the project tooling. As a complement to the summary, [7] reports on the validation of the Requirement Allocation plugin for EATOP. The report is closed with a Summary and reflection.

2 Background

The Synligare project proposal identified 5 project objectives, that was guiding the work. Below, these are summarized as a basis for their evaluation in the subsequent chapter.

2.1 **Project Objectives**

The expected measurable, quantitative and qualitative results of the projects were stated in the project description according to below:

- Identify 5 metrics for the characterization and follow-up of software development
- Identify 5 relevant views to provide overview of a complex requirement set and related entities
- Increase predictability concerning safety, quality and performance
- Reduce the amount of misunderstandings in the communication of specifications
- Reduce the time for development and verification by 10%

2.2 Means

In order to reach the project objectives, a set of engineering needs and use cases have been explored. The basis has been literature surveys, interview studies and clinics, resulting in a set of items to detail, prototype and validate. The main items were as follows:

• Modelling support

EAST-ADL modelling support in three tooling platforms

• Tooling support for metrics calculations

Support for computing model based product and progress metrics in two tooling environments

• Views

Model based graphical, tabular and tree based views in three tooling environments

• Analysis methods

Model based safety analysis and property analysis

• Methodological concepts

Collaboration and specification evolution concepts in a model based setting

3 Validation of Goals Fulfillment

In this chapter, the fulfillment of project objectives will be assessed.

1	Metrics		
,,,,			

The project goal "Identify 5 metrics for the characterization and follow-up of software development" has been fulfilled by the identification, definition and prototyping of a set of metrics. In addition, a flexible and portable metrics definition format has been defined and prototyped.

• Requirement validation ratio

The fraction of requirements that has been validated and approved

• Requirement allocation ratio

The fraction of requirements that are assigned to structural elements

• Function-to-node allocation ratio

The fraction of functions that have been allocated to hardware components

• Feature and function realization ratio

The fraction of features or functions that are realized by concrete entities

• Architecture complexity

The Henry-Kafura complexity of a system architecture providing a metric of the structural complexity

Custom metric

Any metric that can be expressed in terms of model element existence, property values, etc.

The above metrics have been deemed useful for engineers to assess the product and work progress, and for project managers to assess project progress.

Some of them are useful for control and progression of work, while others are more of an assessment of the result. The Henry-Kafura metric is an example of the latter.

	3.2	Views				
--	-----	-------	--	--	--	--

The project goal "Identify 5 relevant views to provide overview of a complex requirement set and related entities" has been fulfilled by by the identification, definition and prototyping of a set of views.

• Graphical architecture view

Functional architecture, hardware architecture and fault propagation structures are presented in a way that respects element hierarchy, ports and connectors. This is critical for understanding systems, although work tasks are more effective in other views.

The view has support for dynamic population of diagrams and automatic, architectureaware place-and-route.

• Graphical view with custom elements including safety

Any model element can be visualized in a diagram, showing element properties and relations. Because elements have dedicated shapes and icons, model content can be

interpreted graphically. The complex structure of models, and the organization into a tree structure with packages for different parts of the model makes such graphical views useful or even necessary to explore and understand model content.

• Tree view with dynamic context

A standard tree view only shows properties and elements directly owned by each element. The augmented view that shows related elements when browsing a tree provides immediate access to the context surrounding a model element. This is a useful view to quicly get an overview of models and thus the often complex system specifications.

• Virtual Tree View

As described above, standard tree views only show direct containment hierarches of architecture models. EAST-ADL and AUTOSAR uses a type-prototype concept for defining hierarchies in a way that supports reuse of components and substructures. The virtual tree view has been found indispensible in browsing and understanding such system architectures, as it allows smooth traversal of these structures.

Table view

When browsing models elements, the properties of one element at a time is shown. With the table view, it is possible to see several elements simultaneously: One element in each row, and the properties of each element in the columns. This view has been found to provide excellent overview of the information and quick access to updating the values.

In addition to the listed views, various aid views, such as search views, connector creator view, version assignment, etc. have been prototyped. In general, each such view increase productivity and understanding, and is appropriate in the context of the complex information that is handled in model based systems engineering.

3.3 Predictability

The goal "Increase predictability concerning safety, quality and performance" was addressed and is deemed as fulfilled by a set of analysis capabilities complemented by view and editing support.

Safety

Safety analysis is provided by the support for error propagation modeling and analysis. The automatic generation of error propagation models from architecture models as well as the support for FTA and FMEA analysis, have been found efficient and useful for increasing predictability of the functional safety.

Quality

Predictability of quality is has been addressed by a combination of view and editor support. Compared to document-based collaboration, models secure consistent and correct exchange of data. The improved view support provided by the project provides for overview of content, and the ability to spot missing or erroneous content. Simularly, the editing support provided by the project reduce the number of mistakes and thus quality issues.

• Performance

Product performance can be observed in many dimensions or domains, such as energy consumption or weight. The project has provided means to increase performance predictability by means of analysis support for property annotations. Because these are mode-based, it is possible to predict e.g. power consumption in a given mode across a complete system or vehicle, which is otherwise tedious and error prone.

3.4 Qualitative improvement: Fewer Misunderstandings

The goal "Reduce the amount of misunderstandings in the communication of specifications" was addressed and is deemed as fulfilled by the combination of methodology, representation and views developed in the project.

Because syntax and semantics are specified for the selected system representations (EAST-ADL and AUTOSAR), OEM and supplier engineering organizations can interpret models in the same way. This is a more compact and unambiguous specification compared to documents.

The views and metrics developed in the project provide further facilitation. While models may be complete and consistent, views and metrics makes it easier to explore and understand the specifications, thus reducing the risk of misunderstanding. Architecture diagrams with established shapes and icons are examples of such powerful views.

3.5 Quantitative improvement: Shorter Development Time

The goal "Reduce the time for development and verification by 10%" requires two subsequent and identical projects to measure precisely. However, this goal was deemed as met by qualitative reasoning regarding the efficiency improvements expected by deployment of project results.

With document-based collaboration as baseline, it is clear that less re-work is required on information exchange when models are used, in particular when the basis is a common exchange format (EAST-ADL).

Model based engineering represents large efficiency improvement for verification. Provided requirements are formulated in a non-ambiguous and testable way, they can be used both for specification and testing. The traceability provided by architecture models means that the right requirements can be identified for a specific verification task. Other efficiency gains for verification comes from clear interface definitions and the opportunity to define plant and environment together with models of the subject systems.

Agile development and short loops is often seen as a prerequisite for shorter development time. Model based engineering is a prerequisite for frequent iterations among stakeholders, as the sheer information management would take up too much time in a document based setting.

Because automotive embedded systems are large, complex and coupled, each engineer needs to integrate with several legacy systems and signals. It has been assessed that as much as 20%-35% of engineering time is spent on seeking information. With model based engineering, system specifications are formalized and seekable or browsable, providing faster access to at least that part of the information needed.

The initial modelling effort may be larger compared to document-based specifications, but as discussed above, it is well compensated for by the potential for automatic and rigorous analyses and inspections provided by models.

4 Key System Technologies

Synligare has developed a set of technologies supporting model based, collaborative development. Below, these will be characterized based on a set of criteria related to technology readiness level assessment used at Volvo. The technologies covered are

- 1. Diagram exchange
- 2. Model based Graphical visualization
- 3. Model aware Place and Route
- 4. Property calculations
- 5. Model based calculations of metrics
- 6. Model based diff and merge
- 7. Requirements Allocation Assistant
- 8. Model based views
- 9. Fault Propagation Model generation and analysis
- 10. Model Based Version Manager.

Technology:	Diagram exchange based on sgraphml format		
Intended purpose:	Sgraphml diagram exchange format allows exchange of model-aware diagrams across modeling tools.		
Inputs and expected outputs	Sgraphml and corresponding model file following the AUTOSAR M3 principles is both output from the source tool and input to the target tool.		
Environmental and functional constraints	Model file containing architecture model/system description must be an XML file complying with AUTOSAR M3 principles for representation		
Benefits:	 A portable format for diagram exchange has several benefits: It represents an infrastructure for handling graphical views enabling improved tooling for views. Adequate graphical views enables more effective communication and understanding of engineering information. Separation of view information from the system description/architecture model Ability to preserve the work spent on organizing system description in views 		
Standards and Regulations:	Sgraphml is an extension to the de-facto standard graphml. The model representation is based on AUTOSAR M3 principles, i.e. arxml or eaxml.		
Scalability:	The format has been validated using diagrams with a size and complexity corresponding to what is suitable to have in a single view.		

Table 1. Characterization of the technology Diagram exchange based on sgraphml format

Technical Risk:	The main risk of the technology is lack of support in tools. This risk is mitigated by the fact that this exchange format can be used as an intermediate format and supported in two steps.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL2, based on Impact: TI2, there is a risk that tools producing diagram exchange files introduce diagram failures that may cause undetected errors in developed items Error Detection: TD2, there is a medium degree of confidence that such failures are detected by downstream activities.

Table 2. Characterization of the technology Model based Graphical visualization

Technology:	Model based Graphical visualization
Intended purpose:	The purpose of the technology is to provide visualization of engineering information with icons and shapes based on the represented element kinds, and to select presentation content based on model content. The latter includes exploitation of meta-model based associations and containments.
Inputs and expected outputs	The input is an EAST-ADL model and the output is a diagram reflecting the content of the model, including its semantics.
Environmental and functional constraints	The input model shall be compliant with the EAST-ADL metamodel.
Benefits:	Presentation of engineering information according the syntax and semantics of the model based representation, makes diagrams non-ambiguous and understandable by domain experts.
Standards and Regulations:	The visualization technology uses EAST-ADL syntax and semantics and the underlying AUTOSAR M3 principles. AUTOSAR models are thus representable with the same approach.
Scalability:	The visualization principles have been validated using diagrams with a size and complexity corresponding to what is suitable to have in a single view.
Technical Risk:	There is no technical risk associated with a model based graphical visualization technology.
ISO26262 related	The generic Tool Confidence Level is TCL2, based on
Qualification Need:	Impact: TI2, there is a risk that tools producing graphical views introduce failures that may cause undetected errors in developed items Error Detection: TD2, there is a medium degree of confidence that such failures are detected by downstream activities.

D4.1

Technology:	Model aware Place and Route
Intended purpose:	The purpose of the technology is to automatically organization diagram layout in a way that respects the meaning of different elements. For example, elements of a hardware architecture and a set of requirement may reside in the same diagram, but shall be placed differently.
Inputs and expected outputs	The input is an EAST-ADL model and an associated sgraphml diagram. The output is an updated sgraphml diagram, where diagram entities are organized according the element kinds.
Environmental and functional constraints	The input model shall be compliant with the EAST-ADL metamodel and sgraphml metamodel respectively.
Benefits:	Presentation of engineering information according the syntax and semantics of the model based representation, makes diagrams non-ambiguous and understandable by domain experts.
Standards and Regulations:	Syntax and semantics is assumed to follow EAST-ADL and sgrapghml, respectively.
Scalability:	Automatic diagram layout has been validated using diagrams with a size and complexity corresponding to what is suitable to have in a single view.
Technical Risk:	There is no technical risk associated with Architecture aware Place and Route.
ISO26262 related	The generic Tool Confidence Level is TCL2, based on
Qualification Need:	Impact: TI2, there is a risk that tools manipulating diagrams introduce failures that may cause undetected errors in developed items Error Detection: TD2, there is a medium degree of confidence that such failures are detected by downstream activities.

Table 4. Characterization of the technology Model Based Property calculations

Technology:	Model Based Property calculations
Intended purpose:	This technology allows mode-based property annotations to be summed over a product hierarchy in order to assess properties such as cost, weight or energy consumption. The subject of analysis may be a subsystem, product or entire product line.
Inputs and expected outputs	The input is an EAST-ADL model representing the product line/product/system/subsystem depending on scope. A requirement corresponding to the expected property value defines what to compute and the model elements need to be annotated with values corresponding to the property kind. Optionally, different values are provided for each applicable mode. The output is the total value in each mode.
Environmental and functional constraints	The input model shall be compliant with the EAST-ADL metamodel and a modeling pattern for property annotations.

Benefits:	Being able to calculate properties such as energy consumption and cost makes it possible to assess which candidates are best among alternatives or checking if all requirements are met. Because well- defined syntax and semantics are used, analysis is rigourous and automatic. The latter can be used for optimization.
Standards and Regulations:	Syntax and semantics is assumed to follow AUTOSAR or EAST-ADL for structure and EAST-ADL for annotations.
Scalability:	The annotation approach has been validated with small-medium sized models, but the low complexity of the analysis suggests that there is no scalability issue.
Technical Risk:	Model annotations are made manually by engineers. Because the analysis is rigorous and automatic there is a risk that too much confidence is put in the results, even if modeling mistakes and uncertain input data may threat validity. Careful validation of the input models is required to mitigate this risk.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL2, based on
	Impact: TI2, there is a risk that tools calculating property values introduce failures that may cause undetected errors in developed items Error Detection: TD2, there is a medium degree of confidence that such failures are detected by downstream activities.

Table 5. Characterization of the technology Model based calculations of metrics

Technology:	Model based calculations of metrics
Intended purpose:	The purpose of this technology is to establish metrics of architecture models. Such metrics allow assessment of the state of engineering data, and thus the product or process progress.
	Metric calculations do not concern product properties like cost or power consumption.
Inputs and expected outputs	The input is an EAST-ADL model and the output is a set of diagrams or numbers representing the selected metric. Examples of current metrics are Henry-Kafura structural complexity and requirement allocation progress.
Environmental and functional constraints	The input model shall be compliant with the EAST-ADL metamodel.
Benefits:	Being able to establish model metrics allows assessment of the current state of engineering data. A complexity metric characterizes the quality of the model, while a completeness metric provides a progress assessment. This can be used to follow up engineering work and to increase the quality of the models, which indirectly improves the product itself.
Standards and Regulations:	Syntax and semantics is assumed to follow EAST-ADL.
Scalability:	Metrics calculations scale linearly with model size and would therefore only slowly reach any limitations. If the scope increases from individual products to entre product lines, aspects like navigation and data management would still be the more challenging concerns.

Technical Risk:	Metrics like requirement allocation completeness assume a specific modeling pattern to be used. Careful validation of the input models is thus required to mitigate this risk. Another risk is to use metrics without full understanding of its meaning. For example, if a problem is complex by nature, it may be inappropriate to enforce changes to the solution to reach a lower model complexity index.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL1, based on Impact: TI1, there is no risk that metrics calculation tools introduce failures that may cause undetected errors in developed items

Table 6. Characterization of the technology Model based diff and merge

Technology:	Model Based diff and merge
Intended purpose:	The purpose of this technology is to compare and highlight differences between models on the basis of model syntax, i.e. the metamodel. A target model may be updated based on the identified differences between source and target models, automatically or manually per identified deviation.
Inputs and expected outputs	The input are two EAST-ADL models or subtrees within the same model. The output is the set of tree entries that deviates.
Environmental and functional constraints	The input model shall be compliant with the EAST-ADL metamodel.
Benefits:	Comparing text files is not sufficient for model based development, since small or large differences in the file may correspond to no, small or large differences in the model. For example, changing the order of elements in the textfile may not influence the model at all.
	By analyzing differences in the model, engineers are provided with a syntactically and semantically relevant presentation of differences between models.
Standards and Regulations:	Syntax and semantics is assumed to follow AUTOSAR or EAST-ADL.
Scalability:	Model diff and merge has been validated with small-medium sized models without scalability issues.
Technical Risk:	There is no technical risk associated with model based diff and merge.
ISO26262 related	The generic Tool Confidence Level is TCL2 based on
Qualification Need:	Impact: TI2, there is a risk that use of tools comparing and merging models introduces failures that may cause undetected errors in developed items Error Detection: TD2, there is a high degree of confidence that such failures are detected by downstream activities.

Technology:	Requirements Allocation Assistant
Intended purpose:	The purpose of this technology is to assist in linking requirement to structural entities. Traceability in the structural model is related to the traceability of the requirement model, which can be exploited when finding suitable target elements. The suggestions are based on Realization links from structural elements to more abstract structural elements and derivations from requirements to more abstract requirements.
	It is also possible to use searching and filtering functions to assist finding the right target entity for a specific requirement.
Inputs and expected outputs	The input is an EAST-ADL model with requirements and an EAST-ADL model with functional or hardware hierarchy. The output is an updated EAST-ADL model with satisfy relations between requirements and structure.
Environmental and functional constraints	The input model(s) shall be compliant with the EAST-ADL metamodel and contain both requirements and components.
Benefits:	Allocation of requirements to a large and complex functional or hardware structure is potentially complex and error prone.
	By helping users finding the right target element, less time is spent on amortizing requirements over architectural models.
Standards and Regulations:	Syntax and semantics is assumed to follow AUTOSAR or EAST-ADL.
Scalability:	Requirement allocation assistant has been validated with small- medium sized models without scalability issues.
Technical Risk:	There is no technical risk associated with Requirements Allocation Assistant
ISO26262 related	The generic Tool Confidence Level is TCL1, based on
Qualification Need:	Impact: TI2, there is a risk that tools allocating requirements to structure introduce failures that may cause undetected errors in developed items Error Detection: TD1, there is a high degree of confidence that such failures are detected by downstream activities.

Table 8. Characterization of the technology Model based views

Technology:	Model based views
Intended purpose:	Model based views have the purpose to present model content according to the syntax and semantics of the model.
Inputs and expected outputs	The input is an EAST-ADL model and the output is a dynamic view that changes content depending on which model element is in scope, and is updated based on model updates.
Environmental and functional constraints	The input model(s) shall be compliant with the EAST-ADL metamodel.

Benefits:	Complex models can be understood easier by presenting content according to its inherent semantics and according to actual relations in the model. For example, showing the related hazards when a safety goal is marked, provides context to the latter. Similarly, showing the voltage attribute of a set of sensors provides overview.
Standards and Regulations:	Syntax and semantics is assumed to follow EAST-ADL.
Scalability:	Model based views are largely independent of model size, as only few elements at a time are concerned.
Technical Risk:	There is no technical risk associated with model based views.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL1, based on Impact: TI2, there is a risk that tools presenting model content introduce failures that may cause undetected errors in developed items Error Detection: TD1, there is a high degree of confidence that such failures are detected by downstream activities.

Table 9. Characterization of the technology Fault Propagation Model generation and analysis

Technology:	Fault Propagation Model generation and analysis
Intended purpose:	The purpose of automatic generation of fault propagation models is to represent errors and how they propagate through each component. Automatic fault propagation analysis provides Fault Tree Analysis or Failure Modes and Effects analysis on the basis of a fault propagation model. The goal is to understand system vulnerabilities and identifying mitigations.
Inputs and expected outputs	The input is an EAST-ADL model representing functional or hardware architecture. The output is a structurally equivalent model representing fault propagation.
	The input for analysis is the fault propagation model, and the output is an FTA (fault tree analysis) or FMEA (failure modes and effects analysis).
Environmental and functional constraints	The input model(s) shall be compliant with the EAST-ADL metamodel.

Benefits:	Error propagation typically occurs along the logical or physical links across components. For this reason, a mirror of the functional or physical architecture is a good starting point for defining how errors occur and propagate. By automatically generating a separate propagation model, engineering judgment can be used to remove or add sources, connections and propagation logic. For example, there may be 100 signals between two subsystems, but the dependability analysis may suffice with a single failure mode and error propagation. On the other hand, two subsystems may be linked by a single logical or physical interface, but there may be several complex ways in which they may interfere with each other.
	Automatic fault propagation analysis on the basis of an integrated architecture model is a way to secure consistency between system representation and a critical system analysis activity.
Standards and Regulations:	Syntax and semantics is assumed to follow EAST-ADL.
Scalability:	Error propagation model generation and analysis has been validated on a medium sized system. Typically, the scope is limited to individual functions or systems, suggesting that scalability is not a concern in most cases. Analyzing complete vehicles or system of systems will be challenging from a tooling and methodology perspective. It typically requires abstraction and divide and conquer to be feasible.
Technical Risk:	This technology requires appropriate input models to produce useful results. Confidence in results is thus dependent on valid assumptions and correct representation of those assumptions.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL2, based on
	Impact: TI2, there is a risk that tools presenting model content introduce failures that may cause undetected errors in developed items Error Detection: TD2, there is a medium degree of confidence that such failures are detected by downstream activities.

Table 10. Characterization of the technology Model Based Version Management

Technology:	Model Based Version Management
Intended purpose:	With model based system engineering, model elements rather than files are relevant for configuration management. Model Based Version Management supports version annotation and management of individual model elements.
Inputs and expected outputs	The input is an EAST-ADL model with or without version annotations. The output is an EAST-ADL model with version annotations, possibly with incremented versions of one or several elements.
Environmental and functional constraints	The input model(s) shall be compliant with the EAST-ADL metamodel.

Benefits:	With increased granularity of version management, it is possible to work efficiently with product lines and variants, and to reuse engineering work in a rigorous manner.
	By supporting version annotations of elements represented by an open exchange format, the information can be shared across tools and organizations.
Standards and Regulations:	Syntax and semantics is assumed to follow EAST-ADL.
Scalability:	Version annotations use a pattern where a separate version annotation element is added for every version annotated element. For this reason, version information grows linearly with the size of the architecture.
Technical Risk:	System integrity may be jeopardized if flaws in version management cause incompatible system elements to be integrated.
	This can largely be mitigated by appropriate integration testing and analysis.
ISO26262 related Qualification Need:	The generic Tool Confidence Level is TCL1, based on
	Impact: TI2, there is a risk that tools manipulating element versions introduce failures that may cause undetected errors in developed items Error Detection: TD1, there is a high degree of confidence that such failures are detected by downstream activities.

5 Validator System

To validate Synligare results, an example system is being used, Adjustable Speed Limit with Traffic Sign Recognition. To illustrate the OEM-supplier collaboration, the Traffic Sign Recognition system is considered as a separate subsystem delivered by a supplier.

5.1 Adjustable Speed Limit

Adjustable Speed Limit is a Vehicle Feature that sets an electronic limit on the vehicle speed. By temporary limiting the maximal speed, accidental overspeed in restricted or sensitive areas is possible. Another scenario is to set the temporary speed limit to the currently allowed road speed.

Adjustable Speed Limit can be combined with both Advanced and Standard Cruise Control. At any given time, speed will not exceed

- Cruise Control Setspeed (or ACC)
- Legal speed limit
- Temporary Speed Limit



Figure 1. User interface of Adjustable Speed Limit

🖨 🧰 TopPackage	N
🖮 🗲 ASLSystemModel	N
🖨 🍘 ASLVehicleLevel	N
🚊 🐓 ASL_TFM	۱
🚊 🖳 ASLRootVehicleFeature	١
🖻 🖳 VehicleMotion	N
🖨 🖳 Longitudinal	- 1
AdvancedSpeedLimit	- 1
🔤 CruiseControl	١.
🖶 🔤 ASLDesignLevel	Ν.
🖃 🧰 ASLRequirements	Ν.
🚔 😽 ASL_RequirementsModel	Ν.
🚊 ↔ SatisfyDeactivateASL	N
8 ReqDeactivateASL	N
General Set SatisfyDetectFault	N
S DetectFault	N
🖇 ReqDeactivateASL	N
§ DetectFault	N
🛓 😼 ASL Requirements	F

Figure 2. Feature Tree of Adjustable Speed Limit







Figure 4. Requirements

6 Validation Scenario

In this section a validation scenario covering Synligare concepts and tools will be presented. The scenario was used to exercise tooling and illustrate project results.

6.1 Overview

The Synligare methods and prototypes have been validated by pursuing a set of engineering activities on the example system described in Chapter 5. Some of the characteristics of the scenario are:

- Supplier-OEM collaboration
- Functional Safety
- Representation of Synligare technologies
- Illustration of how project objectives are met

Figure 5 shows the main steps of the validation scenario.



Figure 5. Main steps of validation scenario

The overall scenario reflects a system development effort where a legacy ststem is extended with new functionality. The OEM has the overall design responsibility, supported by a Tier1 for one part of the system. The next sections will cover the core and dependability related scenario steps.

6.2 Core Scenario

The core scenario covers collaborative function development assuming that a new capability, Traffic Sign Recognition, is added to an existing system, Advanced Speed Limiter.

6.2.1 OEM Views on Original System

Below are some views of the legacy Advanced Speed Limiter system, starting with Vehicle level and going down to the more concrete design level. In this part of the scenario, SystemWeaver was used, and 10 views were selected on the three abstraction levels Vehicle, Analysis and Design Level:

- Vehicle Level: Technical Feature Model in tree and diagram view
- Analysis Level: Functional Analysis Architecture with Neighborhood view
- Design Level: Functional Design Architecture with Neighborhood view
- Design Level: Hardware Design Architecture with Neighborhood view
- Design Level: Allocation of Design Functions to nodes in diagram view
- Design Level: Function Feature mapping in list view
- Design Level: Function Feature mapping in diagram view
- Design Level: Function Metrics
- Design Level: Hardware Metrics
- Requirement Allocation View

FFI 2013-01296



Figure 6. Vehicle Level: Technical Feature Model in tree and diagram view in SystemWeaver

		Synligare		Root[synligars.systemite.net] - SystemWeaver Collaborative Environment ?								? 🗉 🗕 🗆	×
File Welcome Dashboard Items	Projects	Synligare exam	ples										
Back Forward Open New Open -	Overview	📥 Attributes	Filter on search result	Adversions Province Add note Library Complete Statuse Add note Library Complete Statuse Add note the terms. A status Library Complete Statuse Library Complete Statuse Add note the terms. Library Complete Statuse Library Complete Statuse Add note the terms. Library Complete Statuse Library Complete Statuse Add note the terms. Library Complete Statuse Library Complete Statuse Add note the terms. Library Complete Add note									
Navigation Items	Edit		Find	CM	Issues and Notes	Securit	ty	Realization	Extensions	Graph	Analysis		^
AliModels(1)	📋 Final der	10(1)											* ×
EAPackage - Default(modified) -	Analy	sis design •		1		A	SL_Funct	tionalAnalysisArchitecture			I.	AnalysisFunction	nType
Name							Neight	borhood					
Construction C									8 (ii pAF_Srignethangper	Creation	ALLO HERVIS TVILLAN ALLO HERVIS TVILLAN TO NOLUBUL, TO NEL DESIGNAL, JO NEL DESIGNAL, JO NEL DESIGNAL, JO		
۲ <u> </u>	View D	efinition Desc	iption										

Figure 7. Functional Analysis Architecture with Neighborhood view in SystemWeaver

D4.1

File Welcome Dashboard Items	Synligare Projects Synligare examples	Root	synligare.systemite.net] - SystemWeaver Collabora	tive Environment		? 🗈 🗕 🗆 X
③ Back ⑤ Open item □ Copy ○ Open ⑤ Forward ● New item □ Open □ Open <td< th=""><th>Attributes Attributes Attributes</th><th>▲ Versions ~ It New issue Add note ▲ Status ∑ Essues ∑ Notes CM Issues and Notes</th><th>Subtraries Interview Move items f^{**} PortAllocation Manage libraries Security Security FunctionAllocation</th><th>Analysis to Design Feature AF Realization Feature DF Realization Realization</th><th>d* DFSatiafy III Requirement Allocation EAImportExport d* FeatureSatiafy ¹/₁₀₀, RIF Export Meta Model Graph d* TestSatiafy ¹⁰⁰/₁₀₀, RIF Migrid Meta Model Graph Requirements Extensions Extensions</th><th>IS Graph</th></td<>	Attributes	▲ Versions ~ It New issue Add note ▲ Status ∑ Essues ∑ Notes CM Issues and Notes	Subtraries Interview Move items f ^{**} PortAllocation Manage libraries Security Security FunctionAllocation	Analysis to Design Feature AF Realization Feature DF Realization Realization	d* DFSatiafy III Requirement Allocation EAImportExport d* FeatureSatiafy ¹ / ₁₀₀ , RIF Export Meta Model Graph d* TestSatiafy ¹⁰⁰ / ₁₀₀ , RIF Migrid Meta Model Graph Requirements Extensions Extensions	IS Graph
AiModels(1) EAPackage - Default(modified) *	E Final demo(1)	1	FunctionalDesignArc	hitecture	1	- × DesignFunctionType
Name ● Total deme	View Definition Description					(Section)

Figure 8. Design Level: Functional Design Architecture with Neighborhood view in SystemWeaver



Figure 9. Design Level: Hardware Design Architecture with Neighborhood view in SystemWeaver

FFI 2013-01296

欎							Synligar	e	Root[synligare.systemite.net] - SystemWeaver Collaborative Environment						ve Environment	? 📧	×
File	Welcon	ne Dashb	oard	Items	Projec	ts	Synligare exa	mples									
🕒 Back	<u>[]</u> c	pen item 🔹	h (Сору т	🥌 Attr	ibutes	M Find	<i>8</i> 6		▲ Version	s *	№ New is	sue 🛞 Add note	2	¢•	EAImportExport	
Forward	🔒 N	lew item 🔹	Oper	n verview	💡 🍓 Part	ts	Filter on se	arch resu	ult	Status	ete Status	∑! Issues	🔊 Notes		E	Meta Model Graphs	
Navigation		Items		E	dit		-	ind		CN	1	Issue	es and Notes	Security	FunctionAllocation	Extensions	^
0	🗋 AliM	odels(1)			🗋 Fina	I demo	1)										+ X
🗋 EAPack	age - D	Default(mod	dified) -	<u>۲</u>	llocate	d function	•		I.	desig	gnLevel				I.	DesignLevel
Name	Final der	no			OV	мси							ф сюм			1	
ė- <u>C</u>	OEM																
Ē	A 🙆 H	SL Structure				E pLC	C_CruiseControlN	lanager_v1					48 C10	53 C9	83 C8		
		DesignLo	evelele	ments													
		Analysis	Levelel	ements Model													
	÷	systemM	lodel	moder	E	a pLDC_R	oadSpeedLimitMa	nager_v2					EII C12	E C13	G C11		
		📄 🍘 vehic	leLevel	alFeatur													
		ASL_	Analysi	sLevel					Γ								
		🖶 🎰 A	SL_Fur	ictional		al proc_s	ingineSpeedContri	pingr_v1		El proc_Accele	ratorPedalCtri			- C3	14 C2		
		😟 🕀 🔁 F	unctio	nalDesi													
	⊕ - ∩	Extensions	hysical	Design		F8 pVehic	eSpeed ctrl	Γ	fil eldo R	RetarderCtrl v1			f3l C1	51 C6	12 C5		
ė- <u>C</u>	Tier 1							L									
	- 🛄 In - 🧰 Es	nportedFromC (portToOFM	DEM														
1	- 🗋 TS	iR				18 ovehi	cle_model						181 C4				
					ФH	MIIOM			_				() EMS		C SWS2	© IC_Instrume	nt_Cluster_System
						🖽 pLDC_	VehicleSpeedCont	rol_HMICtrl	_v1				DLDC_EngineMa	inagerCtrl_v2	1 S1		
																-	
									_								
						🔠 pLDC.	_SpeedControlMod	e_HMICH_	v1								
					6	E pLDC_V	ehicleIndicationMa	nager_v1									
						🔛 pSpe	edControlButtons_	2_hdlr	ŧ	DSpeedControl	FreeWheel_hd	r					
												_					
					0): Cruise_(Control_Free_whee	۹ <u>ـ</u>									
					View	Defi	nition Dec	crintion									
•				Þ	view	Dell		caption									

Figure 10. Design Level: Allocation of Design Functions to nodes in diagram view in SystemWeaver



Figure 11. Design Level: Function - Feature mapping in list view in SystemWeaver

FFI 2013-01296



Figure 12. Design Level: Function - Feature mapping in diagram view in SystemWeaver

	ollaborativ Sy	nligare			?	A _		×
File Welcome Dashboard Items	Projects Synliga	re examples						
Image: Constraint of the second se	Issues and Security Notes *	FunctionAllocatio	on Realization	Requirements *	Extensions Graph	Functions		
								- × ×
Allwodels(1)	E Function metr	ics - Functional	DesignArchitec	ture		DesignFu	nctionT	ype
	Function		Subfunctions	Inputs	Outputs			
	FunctionalDesignArchi	tecture	15	0	0			
	С		0	0	1			
🗄 🦲 Structure	LDC_AcceleratorPedal	Ctrl	0	3	3			
Designlevelelements	LDC_CruiseControlMa	hager	0	4	1			
Analysisl eveletements	LDC_EngineManagerC	trl	0	6	5			
	LDC_EngineSpeedCon	trolMgr	0 1		1			
systemModel	LDC_RetarderCtrl	-	0	1	5			
vehiclel evel	LDC_RoadSpeedLimitN	lanager	0	13	5			
ASI Analysislevel	LDC_SpeedControlMo	de_HMICtrl	0	6	5			
	LDC_VehicleIndication	Manager	0	14	6			
a P EunctionalDesi	LDC_VehicleSpeedCon	trol_HMICtrl	0	12	9			
	S		0	1	0			
	SpeedControlButtons	2_hdlr	0	1	1			
E Tier 1	SpeedControlFreeWhe	el_hdir	0	3	2			
ImportedEromOEM	VehicleSpeed_ctrl		0	1	1			
	vehicle_model		0	7	1			
in Control China								
	View Definition	Description						

Figure 13. Design Level: Function Metrics in SystemWeaver

th Root[synligare.systemite.net] - SystemWeaver	Collaborativ	Synligare			?	T _	×
File Welcome Dashboard Items	Projects	Synligare examples					
Image: Constraint of the set of the s	rch result	▲ ⊭ ⊕ ≥ Σ! ≥		ک Exten	sions Graph	Hardware	
Navigation Items Edit F	nd	CM Issues and Not	tes Security Funct	tionAllocation			^
AliModels(1)	📋 Final der	no(1)					▼ X
EAPackage - Default(modified) -	Hardv	vare metrics - Ph	ysicalDesignArchite	ecture		I	
Name	Componen	t	Subcomponent	Ports	Pins		
🖃 🧾 Final demo	PhysicalDe	signArchitecture	7	0	0		
	CIOM		0	4	0		
	Cruise_Cor	trol_Free_wheel_	0	2	0		
Structure	EMS		0	4	0		
DesignLevelelements	HMIIOM		0	5	0		
AnalysisLevelelements	IC_Instrum	ent_Cluster_System	0	2	0		
ProductFeatureModel	SWS2		0	4	0		
systemModel	VMCU		0	4	6		
ASL_AnalysisLevel							
esignLevel							
문 관광 FunctionalDesi							
Entersioner							
Extensions							
iner I ImportedEcomOEM							
		afinition Description					
4 III +	view L	Description					

Figure 14. Design Level: Hardware Metrics in SystemWeaver

豑					Synli	nligare Root[synligare.systemite.net] - SystemWeaver Collaborative Environment ? 💿 🗕 🗆 🗙								×			
File ү	Welcome Dashb	oard	Items	Projects	Synligare	examples											
G Back	S Open item 👻	Open	opy * • verview *	 Attribu 🛃 🕹	tes A Find	▼ [™] search resul] • t	➢ Versions ▼ ➢ Status ➢ Complete Status	 № New issue Memory issues 		C Req MF	(C) (C)	EAImpo Meta M	ortExpoi	rt raphs		
Navigation	Items		Ed	it		Find	CM Issues and Notes Security Requirements Req model Extensions										^
C	AliModels(1)			📋 Final d	emo(1)												* X
🗀 EAPacka	age - Default(mo	dified)	• v	III Requ	uirement All	ocation *	on - VehicleLevelRequiremenets Req								remen	ntsMo	del
Name				Function			Pequirement Pequirement Text										^
Name Function Requirement Image: Construction of the second seco								the ACC shall This requirement not try to cont warnings.	(within 500 ent says tha rol the vehic	ms) be in at the ACC cle speed	n state C syst and/o	ACC em sh r send	off. nall i				
	ASL_	Analysis	Level				Res	spectSpeedLimit			The speed lim	it should be	respecte	ed.			
	🗈 🧰 desig	gnLevel	cional.				Kee	epSpeedBelowSpeedLimitati	ion								
	Extensions	abilityPa mentsPa	ickage Ekage	:			ASL	L_SpeedLimitation			ASL shall sec minimum of se RequestedSpe	ure Vehicles elected spece eed	Speed to ed limitat	be the ion and	i d		
	😥 🎒 Impl	ementat	ionSta.				Avo	oidUnintendedSpeedDrop									
		cleLevell ysisLeve	Requir. IRequi. Requir.				TSR	R_V2			Vehicle speed when RSL is a	l shall not ex active	xceed lim	it of ro	ad sig	gns	
	in safe in safe in safe in safe in safe	tyRequir ty mentPac	ements kage	TrafficSig	nControlled		The TSR shall work for the following output rang x = [0m, 100m], y = [-30m, 30m] z = [0m, 20m x = longitudinal, y = transverse, z = vertical								ge: n];		
	TakeRat TakeRat TakeRat TakeRat	eAnnota ionValio ackage	itions Jation				TSR	R_V4			TSR shall wor speed sign ty	k for the foll pes: SE, Dk	owing co K, SF, D,	untries UK	and		
🖨 - 🚞	Tier 1	Linuge		TrafficSig	nControlled		TSR	RVehicleReqPlaceholder									~
•	ImportedFrom	DEM	+	View	Definition [Description											

Figure 15. Requirement Allocation View

6.2.2 OEM Metrics on Original System



Figure 16. Metric: Ratio of realized vs. non-realized Features (SystemWeaver)

D4.1

FFI 2013-01296



Figure 17. Configurable Metric Dashboard: Ratio of allocated requirements (SystemWeaver)



Figure 18. Configurable Metric Dashboard: Requirements' states (SystemWeaver)

D4.1



Figure 19. Configurable Metric Dashboard: Requirements' states (SystemWeaver)



Figure 20. Configurable Metric Dashboard: Ratio of verified requirements (SystemWeaver)

FFI 2013-01296

帶					Synligare	Root[sy	nligare.systemi	ment	? 🗹 _	□ ×			
File	Welcome Dashb	oard Items	Projects	Syn	ligare examples								
G Back	🕵 Open item 🔹	Copy 🔹	i Attribut	es 🛔	Find 🝷 🦄	A Ver	sions 🔻	🕂 New issue	<u>></u>	C III	()	EAImportExport	
Forward	🔒 New item 🔹	Open +	💡 🏪 Parts	Fi	ter on search re	sult 🔏 Cor	nplete Status	∑! Issues ∑		DIF RM	[*]	Meta Model Graphs	
Navigation	Items		Edit		Find	-	CM	Issues and Notes	Security	Requirements	Req model	Extensions	~
	AliModels(1)		🗋 Final de	mo(1)									+ X
🗀 EAPacka	age - Default(mod	lified) - Edit	•		Safety r	equirement met	rics - S	Safety				Requiremen	tsModel
Name			Status Ve	sion	[
- 📄 Fi	inal demo		Work (1)	*									
<u> </u>	OEM		Work (1)					Unallocated Funct	tional safe	ty requirements			
T à	ASL		Work (1)					Upallocat	~				
	E Structure		CS Re (1)					Unanocat	<u></u>				
	Extensions		Work (1)										
	🚊 🦲 Dependa	bilityPackage	Work (1)										
	🖶 🦳 Situat	tions	Work (1)										
	🖶 🦳 Usera	acec	Work (1)										
	🖶 🧰 Datat	vnes	Work (1)										
	Baba	vior	Work (1)										
	Error	Model	Work (1)								$\langle \rangle$		
		rdandBick	Work (1)										
		IUdilukisk	Work (1)										
			Work (1)										
			WORK (1)	Ξ									
		ycase	Work (1)		:								
	e Requiren	nentsPackage	WORK (1)		:								
		cationStatus	Work (1)										
	🖽 🚃 Imple	mentationsta	VVORK (1)										
	🖽 🧕 Sig Vehic	ieLevelRequir	vvork (1)									/	
	🖽 ss Analy	sisteveikequi	Work (1)										
	🖽 🤒 Desig	nLevelRequir	vvork (1)										
	⊞ <mark>š§</mark> Safet	yRequirements	Work (1)										
	E- Safet	У	Work (1)										
	₽ ° 🗳 🖸	erive1	Work (1)										
	- 3	TSR	Work (1)										
	- 5	FSR1	Work (1)										
	E 😽 TS	5C1	Work (1)					Allocate	b				
	§	TSR	Work (1)										
	2T 💦 -	5C2	Work (1)										
	🕀 😽 FS	5C1	Work (1)					1 Unalloca	ated 🔲 1	Allocated			
		5C2	Work (1)										
	- 🚞 Environm	nentPackage	Work (1)										
	🗄 🚞 TakeRate	Annotations	Work (1)	*	View Defi	nition Description	n						
•				•	- new Den								

Figure 21. Configurable Metric Dashboard: Safety requirements (SystemWeaver)

6.2.3 OEM – Tier1 iteration

Below, the Advanced Speed Limiter has been extended with Traffic Sign Recognition. Only external interfaces are defined, since the component is developed by a Tier1 supplier.

To spot the difference, a diff-and-merge plugin was used in the EATOP environment, and a similar functionality is available in SystemWeaver.

载臣		Synligare	Root[sy	/nligare.systemite.n	et] - SystemW	eaver Collaborative Envi	ronment ?	—	□ ×
File Welcome Dashboard Items	Projects Sy	nligare examples							
Back Gopen item Dopen Open Open	Attributes	👫 Find 🔻 👬	· · ·	1+ 🔶 Σ! Σ	∑= 4 ≧ 4	* El (C)	EAImportExport Meta Model Graphs	<u>a</u>	
		Filter on search resu	lt 25						
Navigation Items Edit	1	Find	CM	Issues and Notes	Security Req	uirements Req metrics	Extensions	Synligare	
AliModels(1)	Final demo(1)								• ^
EAPackage - Default(modified) Edit			Versions	- St	ructure			E	APackage
Name Sta	tus Version	Next V	Version	Status Name	•	Final	demo Chan	ige log	
E- 📋 Final demo 🛛 🛛 Wo	ork (1)	*	(1)	CS_Rel 🛄 S	tructure	(1)			
E OEM We	ork (1)		(2)	CS_Rel 🛄 S	tructure				
🖨 🧰 ASL 🛛 🗰 Wo	ork (1)		🖕 (З)	CS_Rel 🗋 S	tructure				
E 🗋 Structure CS	_Re (1)	(2)							
🕀 🛄 DesignLevelelements CS	_Re (1)	(2)							
🕀 🧰 AnalysisLevelelements CS	_Re (1)								
🕀 🧰 ProductFeatureModel CS	_Re (1)								
🖹 🚋 systemModel 🛛 CS	_Re (1)	(2)							
🕀 🍘 vehicleLevel CS	_Re (1)	(2)							
E Main ASL_AnalysisLevel CS	_Re (1)		Properties			☆ Proper	rties		*
Generational CS	_Re (1)		Name: Str	ucture		Name:	Structure		
iii designLevel CS	_Re (3)	(4)							
🖻 🛄 Extensions 🛛 🛛 😡	ork (1)		Version: (1)			Version	(3)		
🕀 🛄 DependabilityPackage 🛛 Wo	ork (1)	= :	Attributes			Attribution	utes		*
🖻 🧰 RequirementsPackage 🛛 Wo	ork (1)		Description	-			ntion		^
🕀 🚟 VerificationStatus 🛛 Wo	ork (1)		Description			A Descri			
🕀 🎒 ImplementationSta Wo	ork (1)		Basic ASL da	ata before includin	g the TSR fur	The upda	ted ISR data is rece	ived from the s	supplier.
🕀 🛐 VehicleLevelRequir Wo	ork (1)								
🕀 😽 AnalysisLevelRequi Wo	ork (1)		Parts			Parts			*
🕀 🛐 DesignLevelRequir Wo	ork (1)		subPackage		Version	subPacka	ge	Version	
	ork (1)		DesignLeve	elelements	(1)	Design	nLevelelements	(3)	
I Safety Wo	ork (1)		AnalysisLev	relelements	(1)	Analys	isLevelelements	(1)	
EnvironmentPackage Wo	ork (1)		ProductFea	atureModel	(1)	Produ	ctFeatureModel	(1)	
TakeRateAnnotations Wo	ork (1)		element		Version	element		Version	
VerificationValidation Wo	ork (1)		🛫 systemMod	fel	(1)	system 🖉	Model	(3)	
imingPackage Wo	orκ (1)								
We lier 1 We	οrκ (1)								
ImportedFromOEM Wo	огк (2)	-							
ExportIoOEM Wo	ork (1)	•							

Figure 22. Variant and Version Support in SystemWeaver



Figure 23. Function Design Architecture with Added Traffic Sign Recognition component (EATOP).

A tot studyer = Charlene			
	Long Couper SUMS 10041 Long Tohong December 7 File die Sought Sand Hauf Kan Wales May	0 0 0 0 0 0	
C Ex Dirit	A bill all halos II Colored halos III B	Company (MARAY - MARAY) 11	
C fant time G fan Card Gap Card Fall Fall Fall Fall Fall Fall Fall Fal	Brann Brannn Brannn Brannn Brann Brann Brann Brann	Note::::::::::::::::::::::::::::::::::::	

Figure 24. Diff and merge plugin used by requirement engineer to detect changes to the system model (EATOP).

The next step is to distribute requirements to the system elements.

Eatop - Eatop Technology	y Demonstrator	-			and the second			×		
File Four Davidate zeli	O Poject Bun Window	w Help					ch Account mit			
	0141315					Qu	CEACCES EB	top		
EAST-ADL Explorer 12 EAST-ADL Explorer 12 ADVAL Example 1 EAST-ADL Explorer 12 AS_ODMSrep1_3 Structure AS_ODMSrep1_3 Example 1 AS_ODMSrep1_3 Example 1 Ex	Project Explorer 18.2-exem ML [EAPackage] [EAPackage] [EAPackage] 18.2-exem New Child Goto type Open In Editor Open NEderences Open Wah		Mousian Assume				-			
o 🚞 Struk	Cut	Ctrl+X	EASTADL Example Editor							
	Copy	Ctrl+C	Eatop - Demo/ASL OEMStep2 1.8.2 eavrol - Eatop Technology Demonstra	tor		1000				0 0 8
×	Delete	Delete	Elle Edit Navigate Search Broject Bun Window Help							
15	Cut	Ctrl+X	B • E 6 ≙ \$ • 0 • % • # • # • 8 • 8 • 6	(n + m +					Quick Access	E Estop
	Сору	Ctrl+C	🔥 "EAST-ADL Explorer 😫 🏠 Project Explorer 🛛 🗎 🎭 🤝 📼	ASL_OE	MStep2_1.8.2.eaoml 33					- 0
10 X	Paste Delete	Delete	- 羿 Demo	Content	sTree					
	Undo	Ctrl+Z	 ASL_OEMStep1_1.8.2.exxml "5_EAXML [EAXML] 							
	Redo	Ctrl+V	Extensions [EAPackage]	REQUIR	MENTS: ASL_OEMStep2_1.8.2.eaxml	Search and filter Show types		MODEL: ASL_OEMStep2_1.8.2.eaxml	Search and filter	Show types
	Rename	F	ASL_OEMStep2_1.8.2.exml	This sec	ion enables the contents of this element to be ec	dited.		This section enables the contents of this elen	ient to be edited.	
	more .		A A A A A A A A A A A A A A A A A A A	Search and filt	er			EAXML Structure		
* [Metrics Unexe Property Analyzer Version Control			Search Search Search Search the search that have the a with the value. Search Remove the search Remove	Indited Isoth type Registered Isoth type Registered dat over all SearchAddRiter TextCoards TextC		Hints Remove Hints Allocate) _ 300046	2 B 3	

Figure 25. Requirement allocation assistant used to allocate requirements to components

After allocating requirements, the requirement metric is checked for completeness.



Figure 26. Requirement allocation metric.

In the next step the Tier 1 receives the model and adds internal structure to the Traffic Sign Recognition component. This is done after importing the structure to EnterpriseArchitect. There is now a difference between the legacy TrafficSignRecoognition component, and the one requested by the OEM that requires changes by both parties: The OEM has overlooked EgoMotionData which is a necessary input and the Tier1 previously did not supply a separate "Confidence" output.



Figure 27. Imported Component Specification from OEM (top) vs. Legacy component at Tier1 (bottom) (EnterpriseArchitect)

R TON 0.2 Lange Falameter Landstord	
W ISALSZ, jeggaly - Enterprise Ardinecti ≥ Elle Fort View PROJECT PACKAGE DIAGRAM FIEMENT TOOIS ANALYZER EXTENSIONS WINDOW HEIP	= - ~
× Ži CompositeSinuture Diagram. "TSR"	Project Browser 🗸 🕂 🗙
Ø start Page	💁 🎦 😫 🙀 🖻 - 🗐 - 🚹 🤘
Start Page 当 TSR × A M + A + A + I C S I Default Style + 比 日 田 田 松 田 田 松 田 田 松 I 田 松 I 田 松 I I C I I I I I I I I I I I I I I I I	Image: Section 2016 Image: Section 2016 Imag
tinPorts tinPor	Properties 4 × 28 21 0 0 10
4 >	Properties Notes
CompositeStructure Diagram/TSP: created: 2016.03.15 modified: 2016.05.17.10:53:48 100% 799 v 1067	

Figure 28. Traffic Sign Recognition in Enterprise Architect

On receiving and importing the new component specification, the OEM sees the added interface and updates the design accordingly, see below.



Figure 29. Updated Advanced Speed Limiter with EgoMotionData component added.

6.3 Dependability Scenario

The dependability scenario covers exchange of dependability related information between OEM and supplier.

6.3.1 Hazard and Risk Analysis

Hazard and risk analysis is performed on solution-independent information and represented on Vehicle level in EADST-ADL. Below we show diagrams and other views from SystemWeaver and their counterparts in EATOP.

D4.1

Synlig:	are manufar	Root[synligare.systemite.net] - SystemWeaver Colla	borative Environment	? T _ T ×
Image: Second	xes A Find • A A Versions •	Image: Provide the state of the s	CafetyConstraints EAImportExport EtaImportExport Meta Model Graphs	
Navigation Items Edit	Find CM	Issues and Notes Security	AnomalyInstanceRef Extensions Dependability	A
AliModels(1)				* ×
EAPackage - Default(modified) * Edit *	IARA - I	Hazardand	Risk	Dependability
Name Statu Vertion ● ↑ find dens Work 0 ● ↑ find dens Work 0 ● ↑ Oth Work 0 ● ↑ Oth Work 0 ● ↑ Sourchare Vertice 0 ● ↓ DependabilityPatage Vertice 0 ● ↓ TackAtelenemBrait Vertice <td>Acceleration SpeedExceeds:</td> <td>SpeedLimitation</td> <td>oliedAcceleration</td> <td>edAccelerationHE RespectSpeedLimit dsSpeedLimitationHE KeepSpeedBelowSpeedLimitation</td>	Acceleration SpeedExceeds:	SpeedLimitation	oliedAcceleration	edAccelerationHE RespectSpeedLimit dsSpeedLimitationHE KeepSpeedBelowSpeedLimitation
	Perintise Description	cedTooMuch	LicedTooMuchHazard → ▲ SpeedRec	IucedTooMuchHE

Figure 30. Hazard and Risk related elements in diagram view (SystemWeaver)

豑						Synligare	:		Root[synlig	gare.systemite.net	t] - (SystemWe	aver Coll	abor	ative B	nviro	nment		? 📧	_	□ ×
File	Welcome Dashb	oard Items	Projec	ts	Synli	gare exan	nples														
G Back	🕵 Open item 👻	Copy -	🤙 Attri	butes	A	Find -	d + ∭		✓ Versions ▼	🕂 New issue 🛭 🏟 Add note		Add note	Σ •		<i>p</i> •			EAImportExport	E		
Forward	🔒 New item 🔹	Overview	, 📲 Part	s	Filt	er on sea	rch result		Complete Status	∑! Issues	<u> ک</u>	Notes	2			•		Meta Model Graphs			
Navigation	Items	Ec	lit			Fi	nd		CM	Issues and	No	otes	Security	An	omaly	Insta	nceRef	Extensions	Dependabil	ity	^
C	AliModels(1)																				
🗎 EAPack	age - Default(mod	lified) - Edit -				III HAF	RA -		I	HazardandRisk								I		Depe	ndability
Name	🔹 🛄 Osec	ases types	Status work Work	Versio (1) (1)	<u>n</u>	Hazard		Traf	ffic situation	Env. situation	ŀ	Hazardous e	event	Seve rity	Expo sure	Cont rolla bility	ASIL	Safety Goal	Safe state		
	🕀 🧰 Beha	vior Model	Work Work	(1) (1)		Uncontro ation	olledAcceler			Highway	Ļ	Uncontrolled ationHE	dAcceler	S3	E2	C2	A	RespectSpeedLimit	EngineOff		
	Haza	rdandRisk	Work	(1) (1)	1:	SpeedEx dLimitatio	ceedsSpee onHazard			Highway	5	SpeedExcee dLimitationH	edsSpee IE	50	E3	C0	QM	KeepSpeedBelowSp eedLimitation	ASL_inactive		
	🗄 🤌 TSC	-	Work	(1)	:	SpeedRe MuchHaz	ducedToo ard			Highway	5	SpeedRedu MuchHE	cedToo	51	E3	C1	QM	AvoidUnintendedSp eedReduction	ASL_inactive		
	🖅 🥭 Safet	ycase nentsPackage	Work	(1) (1)																	
	- Cnvironn	nentPackage	Work	(1)																	
	🕀 📃 TakeRate	Annotations	Work	(1)																	
•	U Verificat	onvalidation	work	(1)	*	View	Definition		Description		_										

Figure 31. Hazard and Risk related elements in table view (SystemWeaver)

D4.1

Eatop - OEM-StepD2/DepGraphs/FromSW/HARA.sgraphml - Eatop Technology	Demonstrator	- 0 ×
<u>File Edit Navigate Search Project Run Window H</u> elp		
📑 • 🗄 🐚 🗁 🕸 • 💽 • 💁 • 🔗 • 📳 • 🖢 • 🖗 • 🖙 🗇	🔹 🔿 💌 🚧 🖄 💥 🏢 Snap to Geometry 📃 👻	Quick Access 🔹 😰 📄 Resource 💶 Eatop
🔥 EAST-ADL Explorer 🙁 🏠 Project Explorer 🛛 🖨 😓 🗢 🗖	🖶 FDA.sgraphml 🔄 DesignLevel 📑 System.sgraphml 📑 HARA.sgraphml 🔀	HaRaSpeedRe "1 🗖
> 2 ASL_19		4
▶ 🙀 BBW_Public		
BBW_Public-Changed		
21 OCM Strep2		
A CoroGraphs		
_T ASI EDA.sgraphml		
A Coperaphs	Appaleration I popptralled	A Lineantralia décembrati
A 🍃 FromSW		
HARA.sgraphml		
6_TSRComponentDependability.sgraphml		
ASL_FSC		
ASL_FSC_Complete.sgraphml		
ASL_FSC_EM		
SL_FSC_EM.sgraphml	ASL - SpeedExceedsSpeedLimitation	SpeedExceedsSpeedLimitati
SI ASL_FSC.sgraphml		
SL_TSC_Complete.sgraphml		
ASL_TSC_EMISgraphmi		
-T HaRal arrer sgranhml		
HaRaSpeedReduction.jpg		
HaRaSpeedReduction.sgraphml		
HaRaSpeedReduction2.jpg	SpeedReducedTooMuch	SpeedReducedTooMuchH
TSR_ErrorPropagation.sgraphml		
ASL_1.8.2.eaxml		
A CONTRACTOR (EAXML)		
a 📄 Extensions [EAPackage]		
DependabilityPackage [EAPackage]		
Behavior [Benavior]		
Dependabilitys [7 items]		, 1
In DL ErrorPropagation [Dependability]	🔲 Properties 🔮 Error Log 📩 Table View 🕂 Explorer Context 🕱 📩 Realized Features 📩	Allocated Require 📩 Visual Model Ove 📮 🗖
FSC [Dependability]		8
HazardandRisk [Dependability]	A 🗢 Referenced by	
💭 ASL [Item]	AccelerationUncontrolled [FeatureFlaw] [featureFlaw]	
FeatureFlaws [3 items]	SpeedExceedsSpeedLimitation [FeatureFlaw] [featureFlaw]	
HazardousEvents [3 items]	A SpeedExceedsSpeedLimitationHazard [Hazard] [hazard]	
 Hazards [3 items] 	SpeedReducedTooMuch [FeatureFlaw] [featureFlaw]	
SpeedExceedsSpeedLimitationHazard [Haz	A SpeedReducedTooMuchHazard [Hazard] [hazard]	
Speedkeduced i confuction [Hazard]	UncontrolledAcceleration [Hazard] [hazard]	
SafetyGoals [3 items]	▲ ⇒ References ■ A due and Consult inside O/abiale Contract (abial)	
IntegrityConstraints [Dependability]	MavanceuspeedLimiter [veniciereature] [childivode]	
SafetyCase [Dependability]		
TSC [Dependability]		
TSR_Dependability [Dependability]		
(
ASL		Chaur dastrian
¥ ····		, Snow desktop

Figure 32. Hazard and Risk, diagram imported from SystemWeaver (EATOP).



Figure 33. Diagram view: Item, Hazard, HazardousEvent, SafetyGoal (EATOP)



Figure 34. HazardousEvent, Property view (EATOP)

D4.1



Figure 35. Hazardous Event, Context view(EATOP)



Figure 36. Hazardous Event, Table view (EATOP)

6.3.2 Functional Safety Concept

FunctionalSafetyConcept is based on artefacts on AnalysisLevel, i.e. the hardware and topology independent representation in EAST-ADL. Below, the requirements, constraints and error propagation constructs are shown in various diagram, tree and list views.



Figure 37. Functional Safety Concept: Context and diagram view (EATOP)



Figure 38. Functional Safety Concept, Error propagation and constraint annotation: Context and diagram view (EATOP)

D4.1



Figure 39. Generated Error Propagation Model (EATOP)



Figure 40. Error propagation model, autorouted diagram (EATOP)

6.3.3 Technical Safety Concept

TechnicalSafetyConcept is based on artefacts on DesignLevel, i.e. the hardware and topology aware representation in EAST-ADL. Below, the requirements, constraints and error propagation constructs are shown in combined views with diagram, tree and list views.



Figure 41. Technical Safety Concept, Overall Diagram View (EATOP)



Figure 42. Technical Safety Concept, Error propagation and ASIL constraints in context view and Diagram View (EATOP)

6.3.4 Tier1 Technical SafetyConcept

Assuming that the tier1 is responsible for a part of the system, the OEM is not exposed to the internals of that subsystem or component. Below, the generatied fault propagation disgram and corresponding fault propagation analysis using HiP-HOPS are shown.



Figure 43. Tier1 autogenerated error propagation model (EATOP)



Figure 44. Fault Tree Analysis using HipHops

7 Summary and Conclusions

This document has described the Synligare project objectives how they have been addressed by the project. The conclusion is that they are all met by applying the technologies identified, detailed and prototyped.

In order to validate project results, example system development has been pursued. This was summarized in this document in terms of engineering activities performed as a validation scenario.

7.1 Reflection

There are a lot of challenges in a multi-company business environment like Synligare. We have learned that any project involving more than one company is a challenge and in Synligare we had representatives from three links in the value chain between the raw material and the final product in the hands of the end customer. The OEM, the Tier 1 and the Tier 2 were represented in the Synligare project and had the opportunity to play the "real game" in a small scale.

The OEM perspective is in the automotive industry governed by delivery plans and just in time delivery is the first and easiest quality measure that should be applied to multi-company business relations. Even small delays in delivery times are generally hiding other bigger quality issues and this is something that should be a trigger of systematic activities to find the root causes to the delay. In the Synligare context, we experienced small but significant delays early in the project and were able to handle them thanks to the early discovery of small delays of deliveries. The countermeasures were fixed deadlines and significantly higher sampling rate for the follow up loop in the project.

The Tier 1 perspective is the customer-supplier perspective. The OEM is at first sight the Customer and the Tier 2's are the Suppliers – but this simplified view is only an illusion. The OEM is an important supplier of information in terms of requirements and expectations on the supplier chain, and the Tier 2 companies are important 'information customers' for the Tier 1. The conclusion is that all actors in the value stream are both customers and suppliers to more or less all other actors. The interdependence between all the links in the value chain has to be clear for all involved parties, regardless of where they are located in the chain. The automotive industry uses the lean concept in many areas inside each company but the lean concept is often forgotten in the relations between the companies. The OEM's can create significant unnecessary labor in the supplier chain if they deliver their requirements late or in an incomplete state. Synligare has shown that requirements and models has to been associated to each other to avoid misunderstanding between the parties but the challenge of translating these blocks of requirements and models between different tools has also been highlighted. A standard language could have reduced this challenge and made Synligare redundant. On the other hand, other standardization initiatives (one example is Autosar) has shown that the winners in commercially driven standardization projects is the Tier 2 level, due to the concentration of expertise in the Tier 2 companies.

The Tier 2 perspectives are several and very dependent on the size and nature of the Tier 2 company. Small Tier 2 companies are either suppliers of products or services, and therefore easy victims for the divide and conquer strategies of the often significantly bigger purchasing departments of the Tier 1 companies. On the other hand, bigger and/or specialized Tier 2's with good relations to the OEM's can often hide behind unclear OEM requirements where they as they have the experts are the only trusted source of knowledge. Examples can be Tier 2's that has delivered tools and/or services to the OEM and therefore has created a monopoly on these services or tools (and tools related services). This dilemma becomes clear when the automotive industry with its cost based purchasing philosophy meets the software industry and their value based pricing paradigm. Customer lock in using dependencies on (different) tools is common in the automotive supply chains and often not considered by the individuals that signs the agreements. On the other hand, customer lock in is the only way of handling the price competition strategy that the bigger players uses to create an effective supply chain (from the big company).

perspective). The eco system of small and bigger companies needs a balance and fair cooperation is always more effective than unfair competition. Trustful relations and transparency between the actors in the value chain are therefore keys to success for the automotive industry, and how this trust building can be improved on national level needs to be studied more.

To summarize, Synligare has shown that the automotive industry has a lot of contributions when analyzing value streams between companies and the success factors in these business relations. Establishment and maintenance of lean, trustful and long term profitable customer-supplier relations are not created without effort but their value cannot be underestimated.

7.2 Conclusions

The project validation effort included applying tools developed in the project onto an example system identified and modeled by the project. On this basis, the project goals have been assessed. It was concluded that the five goals were fulfilled, based on the various project results and a related validation and refinement effort.

8 References [1] EATOP Eclipse Open platform. Source Project: EAST-ADL Tool http://www.eclipse.org/eatop [2] Sparx Systems Inc: Enterprise Architect. www.sparxsystems.com Needs [3] Synligare Consortium: Synligare Deliverable D1.1 Identification. http://www.synligare.eu/ [4] Synligare Consortium: Synligare Deliverable D2.1 Modeling Concepts and Methods. http://www.synligare.eu/ [5] Synligare Consortium: Synligare Deliverable D3.1 Report Tooling. on http://www.synligare.eu/ [6] Synligare Consortium: Synligare Deliverable D3.2 Report Tooling. on http://www.synligare.eu/ [7] Synligare Consortium: Synligare Deliverable D4.1 Appendix: Validation of Requirement Allocation Assistant. http://www.synligare.eu/Deliverables/D4.1A

[8] Systemite AB: SystemWeaver. http://www.systemite.se

[9] HiP-HOPS: <u>www.hip-hops.eu</u>