Contract number: ITEA2 – 10039

Safe Automotive soFtware architEcture (SAFE)

ITEA Roadmap application domains:

Major: Services, Systems & Software Creation

Minor: Society

ITEA Roadmap technology categories:

Major: Systems Engineering & Software Engineering

Minor 1: Engineering Process Support

WP2, WT2.3

Deliverable D2.3.2.b:

Detailed description of use case scenarios and list of requirements from industrial practice

Due date of deliverable: 28/02/2012

Actual submission date: 01/12/2011

Start date of the project: 01/07/2011 **Duration:** 36 months

Project coordinator name: Stefan Voget

Organization name of lead contractor for this deliverable: Continental Automotive

Editor: Jürgen Lucas

Contributors: Philippe Cuenot, Hans-Leo Ross, Jan Hoffmann, Stefan Voget, Roland Geiger, Marion Suerken, Thomas Peikenkamp, Christoph Ainhauser, Tilmann Ochs, Maged Khalil, Harald Günther, Andreas Eckel, Christophe Etienne, Eric Andrianarison, Loic Queran, Michael Schulze

Revision chart and history log

Version	Date	Reason
0.1	20.12.11	Initialization of document
0.2	22.02.12	Incorporation of Review Results (08.02.2012)
0.3	27.02.12	Incorporation of Review Results (27.02.2012)
1.0	22.02.12	Preparation of Draft Deliverable
1.1	25.02.12	Incorporation of Review Results

1 Table of contents

2 List of figures

3 Executive Summary

The goal of this WT 2.3 is to collect requirements for SAFE that are based on the current industrial experience or are based on the experience of tool suppliers. The deliverable D2.3.2 collects these requirements and describes the underlying use cases and methods.

Together with the requirements that have been elicited based on the ISO 26262 (WT2.1. see D2.1) and with the requirements coming from a State of the Art analysis performed in WT2.2 (see D2.2), the requirements collected in WT2.3.2 form the basis for the further work in SAFE. Validation of project results will be performed using this requirements basis.

Based on the rules defined in D2.1, the requirements are classified and allocated to the different SAFE work tasks in WP 3, 4 and 6.

Further, the links to the validation use cases in WP5 are defined and described.

The document you are currently reading documents the different use case scenarios and method scenarios and explains the work performed. The resulting requirements are collected in separate Excel based tables that are added as an appendix to this document. The collection of this document together with the tables represents the complete deliverable D2.3.2.

4 Introduction and overview of document

The goal of WP2 is the elicitation of the project requirements for the work-packages of SAFE. This is done by the analysis of three different sources: the ISO26262 standard, a state of the art analysis and a use case analysis.

- ISO 26262 related Requirements
  - elicited in WT 2.1
  - Documented in tables that are structured according to ISO Structure (Part 2, 3, 4, 5, 6, 7, 8, 9)
- Requirements related to State of the Art analysis
  - elicited in WT 2.2
  - Joint table template with requirements identified by WT 2.3
- Requirements related to model based development
  - elicited in WT 2.3
  - Joint table template with requirements identified by WT 2.2

This document handles with the analysis of industrial use cases and methods. During project preparation phase, a first set of use cases and methods has been identified. This has been a first exercise for the partners to (a) clarify their goals they want to achieve with their participation in SAFE and (b) illustrate their problem range (→ use cases) or their contribution to a possible solution (→ methods).

These use cases and methods have been analyzed in SAFE WT2.3. With the help of templates, requirements have been elicited, that should be considered and handled in SAFE project.

The use cases are collected in Section 5.1: First the template is presented together with a short explanation of the requested input, then for each use case the filled template is given.

The methods are collected in Section 5.2: First the template is presented together with a short explanation of the requested input, then for method the filled template is given.

In section 6, the use cases and methods presented in this deliverable are linked to the validation tasks performed in WP5.

Requirements are collected in a table (see appendix, details on the table format are given in section 8).

The requirements elicited in WT2.3 are complete with respect to all requirements listed in the different use cases, but not all requirements are further considered in SAFE. The collection of included requirements is a first separation of project relevant requirements. This statement is based on the fact that a research project like SAFE always has limited resources. SAFE project has to select a subset of all requirements such that it can be guaranteed that SAFE project creates a substantial contribution at least for the selected subset of requirements.

The detailed analysis and synthesis of requirements derived from the ISO standard are documented with respect to:

- Terms and definitions (to facilitate common understanding)

- Relation with model based development and mapping to WP3 tasks for safety goal definition, safety requirements definition and tracking, architecture definition at different levels, i.e. system, hardware, software
- Identification of needs for methods in WP3 and tools from WP4 to analyze, assess and verify safety criteria and properties of the different assets and reduction of the systematic error rate
- Identification of needs for application rules in WP6

A subset of Use Cases has been introduced by members of the OEM advisory board, which has been installed in 2013. The editor of this document would like to thank the contributing persons (in alphabetical order): Bernd Hedenetz, Henrik Lönn, Philippe Quéré, and Markus Schurius for their input and helpful discussions.

4.1 Overview about requirements traceability in SAFE project

The following picture shows the relationship between the work-packages with respect to the requirements traceability in the SAFE project.

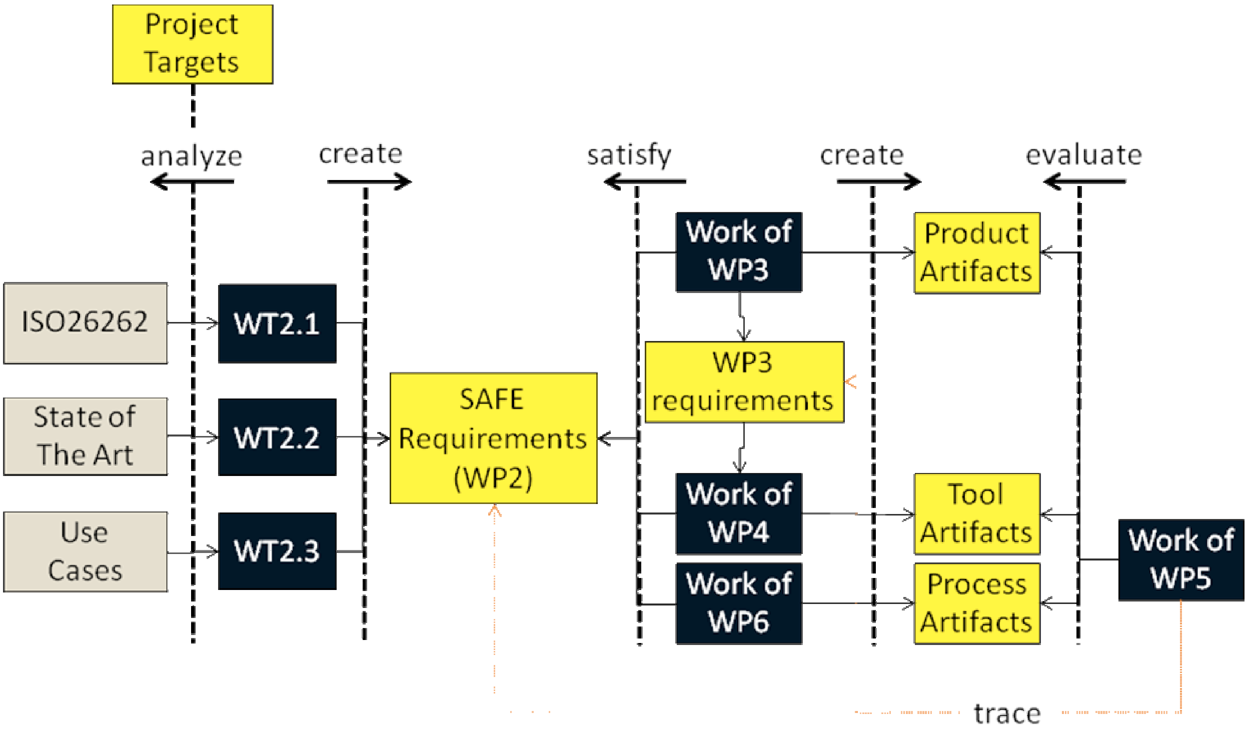


Figure 1: Requirements tracing throughout the project work-packages

WT2.1 analyses the ISO26262 norm, extracts the requirements that are compliant with the project targets and maps these requirements to the work-packages WP3, WP4 or WP6.

The source for WT2.2 is the state of the art analysis and the source for WT2.3 is the use case analysis. But, with respect to creation or extraction and mapping of requirements all three tasks do the same. At the end the SAFE requirements consists of a collection of requirements to be satisfied by WP3, WP4 and WP6.

With respect to requirement satisfaction the product artifacts created by WP3 are used in WP4 for the technology platform.

WP5 has the job to evaluate the results of the project, i.e. the outcome of WP3, WP4 and WP6. This is done in comparison with the traced requirements from WP2 and WP3.

5 Analysis performed

The work in WP2 results in a detailed scope of the SAFE project with respect to the existing work-packages. A key role takes the differentiation of project artifacts into “product model” and “process model”. The definitions of these wordings are presented in D2.1.

Based on these definitions inclusion-, exclusion-lists are created that give examples for the scope definition. These lists are presented in the second section of this chapter. Both together helped to assign the requirements to the project (having a look to the inclusion-, exclusion-lists) and to the work-packages (having a look to the definitions).

5.1 Use Case Scenarios

Scenario Identification	Scenario Title
S01	Model based analysis and code generation for safety aspects in safety relevant systems
S02	Safety assessment of engine management system based on models
S05	Preliminary Hazard Analysis and safety requirements definition
S06	Variant Management Function
S07	Optimization of Model Based Design with safety handling including re-use
S10	Integrated model based safety
S11a	Hazard and Risk Analysis
S11b	Generation of Safety Concepts
S11c	Safety Collaboration
S12	Connect safety analysis with a model-based development process, including requirements management and code generation.
S17	Functional and Technical Safety Concept including analysis and verification according ISO 26262 of a integrated brake system
S18	Integration of safety-related and none safety-related software
S19-1	Safety case contents
S19-2	Variability-aware Safety Case
S19-3	Safety Case in distributed development (OEM / Tier-1)
S19-4	Safety Case notation
S19-5	Model-based safety engineering and integration across system abstraction levels
S19-6	Model-based and compoundable Safety Concepts
S19-7	Combined Safety Analysis in one Safety Model

S19-8	Model-based Safety Patterns
S19-9	Modular Hazard Analysis on model-based function or system (item definition)
S19-10	Synchronization between Hazard and Risk Analysis (H+R) and Item Definition (System Definition)
S19-11	Consistency checks between Modular Hazard and Risk Analysis (H+R) and model-based Item Definition
S19-12	Comparability of Hazard and Risk Analysis (H+R)
S19-13	Guided Hazard and Risk Analysis (H+R)
S19-14	Safety Case properties
S19-15	Supported Analysis on Safety Case Contents
S19-16	Safety Case analysis due to context modifications (Change Impact Analysis)
S19-17	Safety Case incremental compilation/development
S19-18	Safety Case incremental assessment
S19-19	Safety Model Interoperability with various Modeling Tools (XML)
S19-20	Safety Model Abstraction Levels
S20	Verification of software behavior and the effectiveness of implemented safety measures in the presence of faults injected into the microcontroller hardware

5.1.1 Use Case Template

In order to describe and analyze the use case scenarios, a template has been defined as follows:

SCENARIO ID: Sxx <Identification number (see overview table)>	CONTACT PERSON: <Name of contact person(s) (initiators)>
SCENARIO NAME: <Name of the scenario>	
Starting point of the process step that is under consideration: <Explanation> e.g. functional safety concept established	
End point of the process step that is under consideration: <Explanation> e.g. verified system design	
LINK TO Validation Task 5.x: <Indicate to which industrial use case(s) this scenario is attached>	
SCENARIO RELEVANCE: <Indicate which Tasks of WP3/WP4/WP6 are related to activities of this scenario> (This is meant to be a draft allocation to the technical work tasks. Final and detailed allocation will be performed based in the requirements table)	
SCENARIO JUSTIFICATION: <Please explain progress w.r.t. state of practice and quantitative measure of success>	
SCENARIO ACTIVITY: <Please explain the main activities and major steps of this scenario.>	
Output Artifacts <Output documents and work products of the process step(s) under consideration> (e.g. behavior model, algorithms, architecture models, …)	
Requirements generated for SAFE <Collection of requirements that are elicited in this scenario>(will be further handled in parallel to the requirements coming from WT2.1 and WT2.2) Numbering Scheme: Sxx_<local number> (Link in excel-Table)	

5.1.2 Scenario 01: Model based analysis and code generation for safety aspects

SCENARIO ID: S01

CONTACT PERSON:

Christoph Ainhauser

SCENARIO NAME:

Model based analysis and code generation for safety aspects in safety relevant systems

Starting point of the process step that is under consideration:

Planning of safety activities + Hazard and Risk Analysis

End point of the process step that is under consideration:

Implemented system which safety mechanisms realized via safety code generation

LINK TO Validation Task 5.x:

Development of ECU’s for ZF drivetrain systems (WT 5.6)

SCENARIO RELEVANCE:

The activities of this scenario are related to …

WP3:

Task 3.1.1: Hazard analysis, safety goal and ASIL definition

Task 3.1.2: Safety Requirement Expression

Task 3.2.1: System & software models enhancement

Task 3.3.1: Failure and cut set analysis

Task 3.3.3: Safety evaluation

Task 3.5: Meta Model Definition

Task 3.6: Safety Code Generation

WP6:

Task 6.1: Methodology definition

Task 6.2: Application rules

SCENARIO JUSTIFICATION:

With the described scenario, we intend to explore, how code generation can be utilized, in order to ensure safety properties in vehicle systems.

In future we will need to be able to realize high safety relevant functionality e.g. autonomous driving, that may even use services which are potentially not reliable, like GPS or other off-board services. Our vision, when defining this scenario, was to use model based design, with a focus on code generation, in order to enable the flexible and fast integration of safety and non-safety relevant functionality in embedded systems.

Today, even when relying on the AUTOSAR methodology in safety relevant systems, quite a bit of manual effort remains, in order to ensure compliance to safety requirements. Our goal in this scenario is to diminish manual integration effort to near zero.

Apart from enabling the development of new functionality, by increasing process reliability during integration, the gained flexibility will allow for extensive system level optimization, as well as for predictable reuse. Code generation automatically forces a tight link between system design and implementation without relying on conformance checks of design models and according implementations in late stages of the development process.

This will result in a decreasing number of iterations necessary for integration. In consequence, if appropriate tooling is available and other general requirements are fulfilled, concerning e.g. appropriate processes, license-models or warranty regulations, we expect a mayor decrease of verification and certification cost in system development.

We evaluate our methods, based on use cases derived from an existing series system

We will be able to compare state of the art methods with our approach, which will allow us to analyze benefits and drawbacks, in terms of applicability, necessary effort and resulting system quality of our approach.

SCENARIO ACTIVITY:

1. Planning of Safety Process that leverages the SAFE methodology -> Task 6.1:Methodology definition; Task 6.2: Application rules
2. Definition of Functional Safety Concept based on safety goals derived from Hazard and Risk Analysis -> Task 3.1.1: Hazard analysis, safety goal and ASIL definition; Task 3.1.2: Safety Requirement Expression
3. Creation of system model augmented by the following safety relevant information: safety goals, safety requirements, fault model -> Task 3.2.1 System & software models enhancement;
4. Explicit modeling of safety mechanisms, which shall be applied to satisfy the safety requirements associated with the system. → Task 3.5 Meta Model Definition; Task 3.6: Safety Code Generation
5. Analysis and/ or formal verification of behavior in the presence of failures, i.e., if applied safety patterns are capable of fulfilling given safety requirements → Task 3.3.1: Failure and cut set analysis, 3.3.3: Safety evaluation
6. Identification of software safety mechanism that can be generated automatically. Derive software assets to realize the software safety mechanisms and modification of existing system model to allow integration of generated artifacts → Task 3.6: Safety Code Generation

Output Artifacts

Output documents and work products of the process step(s) under consideration are

- Safety Plan
- Hazard and Risk Analysis incl. safety goals
- Functional Safety Concept incl. preliminary architecture
- Technical Safety Requirements
- Technical Safety Concept
- System model driven by functional requirements as well as safety requirements
- Fault model and fault propagation model of the system
- Specification of safety mechanisms to satisfy safety requirements
- Selection of safety mechanisms realized via safety code generation, extension of model to cover required information of generation approach
- Reports of conducted Safety Analysis

Requirements generated for SAFE

Requirements S01-001 – S01-003 in Requirements collection table. (see appendix)

During the specification of the scenario specific requirements, it has been detected that all of the requirements are duplications of original SAFE requirements derived from the ISO analysis. The reason is that the ISO 26262 analysis has already performed with this scenario in mind.

5.1.3 Scenario 02: Safety assessment of engine management system based on models

<table>
<tr><td><b>SCENARIO ID: S02</b></td><td>CONTACT PERSON:
Philippe Cuenot, Christophe Etienne</td></tr>
<tr><td colspan="2">SCENARIO NAME:
Safety assessment of engine management system based on models</td></tr>
<tr><td colspan="2">Starting point of the process step that is under consideration:
Item definition</td></tr>
<tr><td colspan="2">End point of the process step that is under consideration:
System analysis performed and safety case proved in regard to functional architecture and hardware architecture</td></tr>
<tr><td colspan="2">LINK TO Validation Task 5.x:
Continental Engine Management System (EMS)
Dedicated function safety critical of the EMS.
Description of use case in
https://safe.offis.de/svn/svndav/20_Meetings/Project_Meetings/2011_10_10-12_PMC-PTC_Munich/Presentations/SAFE_WP5_Task5.2_Conti_Automotive_Fr.pdf</td></tr>
</table>

SCENARIO RELEVANCE:

The activities of this scenario are related to …

WP3:

Task 3.1.1: Hazard analysis, safety goal and ASIL definition

Task 3.1.2: Safety Requirement expression (Adaptation to automotive)

Task 3.1.3: Safety case documentation (new)

Task 3.2.1 :System & software modeling (standardized)

Task 3.2.2: Hardware modeling (New)

Task 3.2.3: Failure propagation (Adaptation to automotive)

Task 3.3.1: Failure and cut set analysis (Adaptation to automotive)

Task 3.3.4: Safety and multi criteria benchmark (new)

Task 3.4 : Variant management (improvement)

Task 3.5 Meta-Model definition

WP4:

Task 4.1: Meta Model Implementation

Task 4.2.1: Plug-in for traceability and requirement import

Task 4.2.3: Plug-in for failure and cut-sets analysis

Task 4.2.5: Plug-in for pure::variants seamless integration

Task 4.2.6: Plug-in for safety and multi criteria architecture modeling and benchmarking

Task 4.3: PREEVision extension

SCENARIO JUSTIFICATION:

EMS system includes today non safety and safety critical functions. The engineering is based on platform approach and reuse mechanism where safety requirement are managed in database. Failure analysis is performed by hand (recorded in tool) with manual traceability and impact analysis over the overall architecture and component.

The progress is to be able to represent on model level, a several abstraction view of the system from the functional architecture (based on SysML and EAST-ADL2 concept), in relation to vehicle architecture from OEMS with clear interface. Then seamless traceability versus Autosar component level and hardware low level description compliant to Autosar driver interface is targeted. Form this description and based on failure mode of each architecture and component model, propagation of the failure for critical safety shall be evaluated in the context of projection on execution platform.

Failure mode composition, architecture exploration considering failure impact, for final fault prevention on safety critical function is a real progress.

Furthermore demonstration of no segregation with adequate mechanism is required by ISO WD26262.

- Automatic generation of FTA and FMEA in relation to hazard define at vehicle level
- Capability to model all abstraction of the system
- Exchange of models with OEMs at high abstract level (back box)
- Manage the diversity of the function and relation to impact of failure mode

SCENARIO ACTIVITY:

Main activities associated to this scenario will be :

All these activities are done in an iterative mode

- Item definitions and impacts of business models and variants
- Hazard analysis is captured and functional safety concept, technical safety concept is defined
- Technical safety requirements are derived
- Architectural and component modeling , functional level model, hardware level model
- Mapping between artifacts is provided
- Consistency check are performed at each stage
- dysfunctional model is generated
- This models are decorated with safety attributes extracted from the requirements
- The failure modes are identified and annotated in functional models and hardware topology
- Verification/generation of functional/dysfunctional model is performed
- The architectural projection on hardware platform is explored by an analyzer to evaluate safety critical aspect at abstract levels. Possible safe allocation software on hardware should be proposed.
- Execution platform compliant to AUTOSAR R4.0 is characterized by failure modes and error propagation:
    - Hardware ASIC composed of functional block interconnect hardware element (ASIC or microprocessor)
    - Microprocessor functional block to software driver and run time environment
    - Driver and run time environment to software component
    - ASIC component consideration
- The functional architecture is mapped on the configured execution platform with Autosar compliance
- Failure mode analysis with error propagation is performed to verify initial safety results on abstract models.
- Quantitative failure rate is computed for hardware
- Assessment of safety case is performed , if targets not achieved a new iteration is launched starting at the functional safety concept
-

Variants have to be considered at all stages

Output Artifacts

Output documents and work products of the process step(s)

(e.g. behavior model, algorithms, architecture models…)

(TBD)

- Hazard analysis, safety goals Safety requirements, functional safety concept
- Architecture model of the functional safety concept
- logical functional model with allocation and mapping of safety requirements
- Hardware architecture with mapping of safety requirements
- Dysfunctional model based on previous architecture
- FTA,FMEA generated based on Dysfunctional model
- Quantitative analysis results for hardware
- Safety assessment

Requirements generated for SAFE Requirements S01-006 – S01-006 in Requirements collection table. (see appendix) Plus requirements from WT2.1 assigned to tasks defined in this document. (cf. Section SCENARIO RELEVANCE)

5.1.4 Scenario 05: Preliminary Hazard Analysis and safety requirements definition

<table>
<tr><td><b>SCENARIO ID: S05</b></td><td>CONTACT PERSON:
Adaptation to SAFE: Stefan Voget</td></tr>
<tr><td colspan="2">SCENARIO NAME:
Preliminary Hazard Analysis and safety requirements definition</td></tr>
<tr><td colspan="2">Starting point of the process step that is under consideration:
Project started and project scope is defined.</td></tr>
<tr><td colspan="2">End point of the process step that is under consideration:
Safety concept available</td></tr>
<tr><td colspan="2">LINK TO Validation Task 5.x:
Fiat Powertrain Engine Management System (EMS)
Dedicated function safety critical of the EMS.</td></tr>
<tr><td colspan="2">SCENARIO RELEVANCE:
The activities of this scenario are related to …
WP3:
Task 3.1.1: Hazard analysis, safety goal and ASIL definition
Task 3.1.2: Safety Requirement Expression
Task 3.1.3: Safety Case Documentation
WP6:
Task 6.1: methodology definition
Task 6.2: application rules: definition of architecture constraints
Task 6.2: application rules: Decomposition recommendations and design for safety techniques</td></tr>
<tr><td colspan="2">SCENARIO JUSTIFICATION:
The development flow of a safety relevant control item begins with the definition of the perimeter of the application and the successive identification of the various scenarios in which the overall system operates. In the case of an engine management system it includes all the operating conditions and operating modes of the vehicle, both during normal use and maintenance. The interaction of the powertrain system with the rest of the vehicle must be taken into account, but especially the surrounding environment is of utmost importance.
The development team, based mainly on field experience, has to foresee all possible allowed and not allowed users' behaviours.
Once the list of the possible scenarios is available, their occurrence must be evaluated.
At this point, for each scenario it is necessary to identify the hazards which might occur and that have to be absolutely avoided. For each of these hazards the severity and the controllability must be estimated, which leads to the evaluation of an integrity level</td></tr>
</table>

(ASIL) for each hazard and to the consequent definition of the corresponding safety goal.

Safety goals are in turn assigned to a preliminary system architecture, in which for each sub-system safety requirements are developed.

Quantitative measure:

- Defining the system mission and operation profile
- Definition of the hazards in relation to the different scenarios
- Safety goals, safe states modelling
- Decomposition and attribution of ASIL’s
- Expression of formal requirements

SCENARIO ACTIVITY:

- Scenarios and operating modes are described
- Scenarios and operating modes are linked to hazards
- Hazards are linked to occurrence, severity and controllability
- ASILS and safety goals are estimated
- Requirement are described, modelled and mapped to architecture blocks
- ASILS are decomposed

Documentation of the process is generated.

Output Artifacts

Safety requirements

Safety goals

Hazards including ASIL’s

Requirements generated for SAFE

The scenario follows a strict process with regard to ISO26262. The requirements directly associated to the normal ISO part are not added here as they are already part of D2.1.

But in addition to that specifics of an engine control unit structure are expressed. This leads to additional requirements:

Requirements S05-001 – S05-002 in Requirements collection table. (see appendix)

5.1.5 Scenario 06: Variant Management Function

SCENARIO ID: S06	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Variant Management Function	
Starting point of the process step that is under consideration: Start of Requirements analysis	
End point of the process step that is under consideration: Detailed system design established and safety measures defined	
LINK TO Validation Task 5.x: *There is no validation task linked directly to this scenario, validation will be performed based on concept document.*	
SCENARIO RELEVANCE: The activities of this scenario are related to … WP3: Task 3.1.1: Tracking of Safety Requirements Task 3.1.2: Safety Requirement Expression Task 3.2.1: Introduction of safety attributes in E/E-architecture models Task 3.2.1: Component model of the safety concept Task 3.3.1: Support for safety analysis FMEA Task 3.3.3: Safety and multi-criteria architecture benchmark Task 3.4: Variant management of model lines and component supplier WP6: Task 6.1: methodology definition	
SCENARIO JUSTIFICATION: The number of functions and model lines are growing in the automotive industry. In parallel the number of variants is growing in the dimensions: functions, model lines, components, suppliers. In the development certain measures have been taken to prohibit the multiplication of development effort, e.g. platform concepts, modularization, model based development. By introduction of the ISO26262 it has to be ensured that the variants don't lead to a multiplication of the development effort for the functional safety. Quantitative measure:	

- Definition and tracking of the safety requirements - Introduction of safety attributes in the E/E-Architecture models - Definition of methods to describe and manage the variants of model lines and supplier components in the architecture model including the safety concept - Optimize effort of safety analysis for system variants - Support safety analysis FMEA methods by extraction of component information out of the safety concept of the architecture model - Consistent information base by using the architecture model including the safety concept for safety analysis and safety case documentation
SCENARIO ACTIVITY: Main activities associated to this scenario will be : - Definition and tracking of the safety requirements - Definition of safety attributes for the E/E-architecture models - Extension of the E/E-Architecture models with the safety concept - Identification of variants points, e.g. interfaces, parameter and handling from the perspective of safety - Modeling of the function, component, safety concept, including variants - Multi-criteria architecture benchmark based on the E/E-Architecture models - Support the FMEA by extraction of structure information out auf the model and generation of templates for the analysis - Documentation of the safety case - Compliance to ISO26262
Output Artifacts - Safety Requirements collected - Extended architecture - Model of functions and components - Safety Concept including variants - Architecture Benchmark - Safety Case Documentation
Requirements generated for SAFE Requirements S06-001 – S06-007 in Requirements collection table. (see appendix)

5.1.6 Scenario 07: Optimization of Model Based Design with safety handling including re-use

SCENARIO ID: S07	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Optimization of Model Based Design with safety handling including re-use	
Starting point of the process step that is under consideration: Item Definition	
End point of the process step that is under consideration: Technical safety concept established	
LINK TO Validation Task 5.x: *There is no validation task linked directly to this scenario, validation will be performed based on concept document.*	
SCENARIO RELEVANCE: The activities of this scenario are related to … WP3: Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.1.2: Safety Requirement expression Task 3.1.3: Safety case documentation Task 3.2.1: System and Software models enhancement Task 3.3.3: Safety and Multi Criteria Architecture Benchmarking WP6: Task 6.1: methodology definition Task 6.2: application rules: Safety mechanisms Task 6.2: application rules: Decomposition recommendations and design for safety techniques Task 6.2: application rules: Compliance with architecture constraints Task 6.2: application rules: AUTOSAR platform configuration for safety	
SCENARIO JUSTIFICATION: Today, most of the safety critical systems are designed using textual formalism: system requirements, functional and organic architecture, and requirement allocation to system parts, parts development, and validation plan. So safety aspects, along all the design process, are treated manually using system and software functional and dysfunctional analysis, FTA and FMEA.	

The two objectives of this use case are:

- to introduce Model Based Design with safety handling in the early phases of a system design and until system architecture and software component design, and
- to demonstrate the ability of tools and methods to optimize safety process efforts in case of "reuse based" system/software design.

The engineering process will be first applied using traditional methods with a set of measures concerning engineering effort, and then the same process will be applied using the project tools and methods with a new set of measures.

Then, the comparison of the two methods will give a quantitative measure of success.

It is expected that the system would contain Software components from different sources, both Renault and suppliers, some of which would be seen as black boxes with enough information to carry out the complete process.

SCENARIO ACTIVITY:

The scenario will have two cycles of these activities (from ISO26262 process), one with traditional methods and the second with the project tools and methods:

- Hazard analysis and risk assessment
- Functional safety concept
- Specification of technical safety concept and system design
- Specification of software safety requirements
- Software architectural design
- Software integration and testing
- Verification of software safety requirements
- Safety validation and Functional safety assessment

Output Artifacts

All the work products produced by the work steps listed under "Scenario activity"

e.g. HRA, Functional and technical safety concept, system design, …

Requirements generated for SAFE

Requirements S07-001 – S07-002 in Requirements collection table. (see appendix)

5.1.7 Scenario 10: Integrated model based safety

SCENARIO ID: S10	CONTACT PERSON: Eric ANDRIANARISON
SCENARIO NAME: Integrated model based safety	
Starting point of the process step that is under consideration: Once a safety concept has been established and considering a given sub system to be realized, use of SAFE to handle all the models, requirements and analyzes required by ISO26262 to achieve the functional safety design	
End point of the process step that is under consideration: Demonstration of the technical safety concept(s) which have been designed Requirements for next level of implementation Availability of required information for safety case documentation	
LINK TO Validation Task 5.x: WT5.5 Valeo Engine Management Systems and Valeo Starter Alternators Systems	
SCENARIO RELEVANCE: The activities of this scenario are related to… WP3: Task 3.1.2 :Safety requirement expression Task 3.1.3: Safety Case Documentation Task 3.2.1 : System and Software models enhancement Task 3.2.2: Hardware description Task 3.2.3: Failure and propagation Task 3.3.1: Behavior / Failure Translator Task 3.3.3: Safety evaluation and conformance checker (fault injection and multi core constraints) WP6: Task 6.1: methodology definition Task 6.2 application rules: decomposition, V&V techniques	
SCENARIO JUSTIFICATION: The Valeo industrial use case and related scenarios will target evaluation and demonstration of progress beyond current practices regarding: Requirement management Insure a seamless handling of safety requirements within overall requirement management providing relevant coverage and impact synthesis for the safety case documentation. Avoid inefficiency of document oriented traceability by introducing model centric requirement management in design activities (refer to dysfunctional	

modelling improvements)

Continuous modelling

By merging or at least coupling functional and dysfunctional modelling while sharing common abstraction levels (including AUTOSAR R4.0), consistency of the overall safety concept is achievable with an optimized effort. Furthermore sharing the same ground between designers and safety experts insure consistency during the complete lifecycle and especially while iterating the different increments or during maintenance.

Automated safety analysis

Due to the sound basis of functional / dysfunctional modelling it will be possible to capture elements and feed inputs in FMEA and FTA thus avoiding double filling and synchronization issues between design and safety teams. Dysfunctional modelling will allow some automatic computation in the safety analysis allowing safety experts to focus on critical topics. Above improvement on the coupling with design, lowering the effort to critical issues shall also allow to be more reactive during increments.

Continuous verification

Final objective of the whole set of improvements is to provide **SA**fety **I**n the **L**oop (SAIL) ability to allow continuous verification while walking through the development cycle and involving the different development teams.

Fault injection (e.g. on models) will be addressed.

Qualitative and quantitative measures:

- Safety concept consistency insured through relevant abstraction levels
- Efficient modelling mixing functional and dysfunctional focuses
- Formal exchange with OEM and subcontractor organizations based on models
- Consistency of safety analyses done in the different levels (hierarchical links, impacts)
- Efficiency of automated safety analyses realization and maintenance
- Consistency of safety traceability with overall traceability
- Efficiency of model centric requirement management
- Efficiency of safety products developments by tight coupling of designers with safety experts sharing the same technical ground

SCENARIO ACTIVITY:

Considering a subsystem having high level ASIL coming from Engine Management / Starter Alternator Systems, the main activities associated to scenarios of the use case will be :

- existing subsystem models (from SysML/Simulink) reuse and capture with the relevant editors (SysML / EAST ADL2 / simulink …) in the different architectural views
- [TOOL]:
  - Use of a pre industrial SysML / EAST ADL GUI, possibly not commercial but providing usability by regular end user (system people from HW, MK, electrotechnics, safety…)
  - GUI allowing capture of metamodel entities and related properties in an ergonomic way
  - nice to have upstream transformation to capture structure view of

simulink models into SysML/EAST ADL

- safety requirements capture in models to allow model centric requirement management
- [TOOL]:
    - Capability of SysML EAST ADL GUI to capture RIF / DOORS requirements in the model
    - Ability to capture traceability information in the model
- addition of safety artefacts due to safety concept decisions (attributes or whatever, failure modes, failure propagation …) is done in the relevant levels, either on functional blocks, hardware blocks (hardware description and IP descriptions at gate level), AUTOSAR SWC …
- [TOOL]:
    - Metamodel enriched with fault and propagation artefacts and supported by the GUI
- At each level, coupling to safety analyses of that level to achieve SAIL verification. Complete automatically generated skeleton to achieve the safety analysis (FMEA, FTA …)
- [TOOL]:
    - Coupling of both SysML and East ADL to Altarica or a commercial tool (ITEM toolkit and IQRM) and automatic generation of FTA and FMEA (FTA from altarica and FMEA from model ?)
- Simulation platform encompassing dysfunctional to allow fault injection and execute the safety concept
- [TOOL]:
    - Coupling of simulink simulation platform with TTCN scripting langage to insure fault injection and conformance tests
- Use of application rules whenever relevant and available

Output Artifacts

(TBD)

Requirements generated for SAFE

Requirements S10-001 - S10-009 in Requirements collection table. (see appendix)

5.1.8 Scenario 11a: Hazard and Risk Analysis

SCENARIO ID: S11a	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Hazard and Risk Analysis	
Starting point of the process step that is under consideration: Item Definition	
End point of the process step that is under consideration: Safety Goals established	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario, validation will be performed based on concept document.	
SCENARIO RELEVANCE: The activities of this scenario are related to … WP3: Task 3.1.1: Safety Requirement Expression Task 3.2.1: Introduction of safety attributes in E/E-architecture models	
SCENARIO JUSTIFICATION: (see Scenario Activity)	
SCENARIO ACTIVITY: An item is identified by referencing the features realized by the concrete elements. The concrete elements (FAA, FDA, HDA, SWC, runnable) correspond to the system or array of systems in ISO terms. In Hazard and Risk Analysis, it is considered helpful to establish a collection of driving states describing the different operational situation (on vehicle level). These states should be annotated with exposure times and maybe other criteria necessary for HRA. Based on the requirements of a given feature, it is necessary to generate the Flaws and hazards that occur in case requirements are not fulfilled in a specific scenario. Based on these hazards, the safety goals can be modeled as inverse of respective hazards and the ASIL can be calculated based on the operation states and the character of the hazard.	
Output Artifacts Hazard and Risk Analysis, collection of safety goals with ASIL	
Requirements generated for SAFE Requirements S11-001 - S11-004 in Requirements collection table. (see appendix)	

5.1.9 Scenario 11b: Generation of Safety Concepts

<table>
<tr><td><b><u>SCENARIO ID:</u> S11b</b></td><td><u>CONTACT PERSON:</u>
Adaptation to SAFE: Jürgen Lucas</td></tr>
<tr><td colspan="2"><u>SCENARIO NAME:</u>
Generation of Safety Concepts</td></tr>
<tr><td colspan="2"><u>Starting point of the process step that is under consideration:</u>
Hazard & Risk Analysis performed, Safety goals are present</td></tr>
<tr><td colspan="2"><u>End point of the process step that is under consideration:</u>
Functional and Technical Safety Concept established</td></tr>
<tr><td colspan="2"><u>LINK TO Validation Task 5.x:</u>
There is no validation task linked directly to this scenario, validation will be performed based on concept document.</td></tr>
<tr><td colspan="2"><u>SCENARIO RELEVANCE:</u>
The activities of this scenario are related to …
WP3:
Task 3.1.2: Safety Requirement Expression
Task 3.2.1: System and software models enhancement
Task 3.1.3: Safety Case Documentation</td></tr>
<tr><td colspan="2"><u>SCENARIO JUSTIFICATION:</u>
(see Scenario Activity)</td></tr>
<tr><td colspan="2"><u>SCENARIO ACTIVITY:</u>
In establishing safety concepts, it is necessary to generate safety requirements reflecting the goals identified in HRA. It must be traceable by which safety requirement a safety goal is fulfilled. It should be possible to semi automatically derive the requirements from the goals and to support the traceability by templates that enforce links between safety goals and functional/technical requirements. For functional and technical safety requirements, it will be necessary to understand and model the possible errors based on the nominal model of the component. The analysis level models support the identification of functional safety requirements while the design level models support the identification of technical safety requirements.</td></tr>
<tr><td colspan="2"><u>Output Artifacts</u>
- Error Model
- Functional Safety Concept
- Technical Safety Concept
- Preparation of Safety Case</td></tr>
<tr><td colspan="2"><u>Requirements generated for SAFE</u>
Requirements S11-005 - S11-012 in Requirements collection table. (see appendix)</td></tr>
</table>

5.1.10 Scenario 11c: Safety Collaboration

SCENARIO ID: S11c	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Collaboration	
Starting point of the process step that is under consideration: Functional and technical Safety concept established on system level	
End point of the process step that is under consideration: Components and responsibilities are clarified, target values (e.g. FIT) are defined, Safety plans are established	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario, validation will be performed based on concept document.	
SCENARIO RELEVANCE: The activities of this scenario are related to … WP3: Task 3.1.2: Safety Requirement Expression Task 3.1.3: Safety Case Documentation Task 3.2.1: System and software models enhancement Task 3.3.2: Safety Evaluation	
SCENARIO JUSTIFICATION: (see Scenario Activity)	
SCENARIO ACTIVITY: Most development activities are split between several parties (OEM, Tier-1, Tier-2 supplier …) In the area of functional safety the interfaces regarding safety must be clarified and monitored. E.g. The fault rates must be distributed over the different parties, safety goals and ASILs must be consistently defined. Hence, it shall be possible to divide work vertically and horizontally between stakeholders. Vertically: The OEM or tier-X defines nominal and safety specification for the tier-1 or tier X+1 which responds with nominal and safety specifications on the next lower abstraction level and down. Horizontally: The OEM or tier-X defines initial interfaces between nominal and safety specification of subsystems. Tier-1 or Tier X+1 respond with refined and agreed interface specifications between their subsystems.	

Output Artifacts Safety attributes on component level Safety contracts Component specific views of overall system Safety Plan on system and component level
Requirements generated for SAFE Requirements S11-013 - S11-016 in Requirements collection table. (see appendix)

5.1.11 Scenario 12: Connect safety analysis with a model-based development process

Scenario-ID: S12	CONTACT PERSON: Roland Geiger, Jürgen Lucas
SCENARIO NAME: Connect safety analysis with a model-based development process for drivetrain transmission systems, including requirements management and supporting code generation.	
Starting point of the process step that is under consideration: Requirements on the system as defined in the Functional Safety Concept	
End point of the process step that is under consideration: Validated code generation that fulfill certain safety requirements	
LINK TO Validation Task 5.x: Development of ECU's for ZF drivetrain systems (WT 5.6)	
SCENARIO RELEVANCE: WP3.1.2 / WP3.3 / WP 3.6/ WP6.x	
SCENARIO JUSTIFICATION: The development process of safety critical systems is (as far as possible) based on the reuse of predefined requirements and existing implementations. Nevertheless, it must be in line with the ISO 26262. The requirements are maintained within a database system which is connected to a change management system for the development. The resulting design forms the basis of the failure analysis which addresses the safety features of the product in its dedicated environment. The interfaces between all dependent work steps and between customer and supplier need to have well defined links and need to be traceable. Seamless traceability can only be guaranteed, if the relevant information contained in different work products is exchangeable. The requirements defined when performing the safety analysis have to be handled by the requirements management. In addition failure modes and their effects build a base to define test cases and test scenarios for the safety critical system in Failure Mode Effect Testing. With this scenario it is intended to identify and close the gaps between safety analysis and development process while respecting the ISO 26262 process requirements. This	

shall be elaborated for several safety requirements which are relevant for a drivetrain series ECU. Additionally it shall be explored, how code generation can be utilized, in order to ensure the related safety properties on related communication patterns. The goal in general is to reduce manual integration effort as much as possible. This will result in a decreasing number of iterations necessary for integration and will thus decrease integration effort related to iterations.

Possible quantitative measures:

- share of information reused between different safety analysis methods
- share of generated links between analysis, test requirements and system requirements
- share of generated safety relevant patterns
- share of seamless connected models in a development and analysis process

SCENARIO ACTIVITY:

Within the process range defined above, the following main activities are performed, which are related to SAFE project:

1. Define requirements on Software which are directly related to the functional safety concept and to the Hardware-Software-Interface. Goal is to achieve necessary architectural element attributes while maintaining transparency and single-source data with failure and cut-sets analysis results.
2. This activity must also consider the integration of a non-safe application while maintaining the intrinsic safety of a safety relevant system.
3. Enlarge architectural and component models by safety attributes, which are used to generate the documentation of the safety case.
4. Perform analysis of the behavior in the presence of failures.
5. Perform analysis of capability of the applied safety patterns to fulfill the given safety requirements, e. g. to fulfill ISO 26262 with respect to requirements on independence and non-interference.
6. Architectural models are converted to analysis models. Failure mode analysis with error propagation is performed to verify initial safety results on abstract models.
7. Derive requirements on Code generation for a drivetrain application.
8. Define suitable methodology steps and corresponding application rules.

Output Artifacts

Output documents and work products of the process step(s) under consideration are

- Requirements both functional and safety related
- Analysis reports
- Requirements based on technical safety concept
- Specification of safety relevant signals and the necessary safety measures for these signals
- Detailed specification of necessary SW safety measures
- Specification of test cases to verify the effectiveness of the safety measures

Requirements generated for SAFE

Requirements S12-001 - S12-009 in Requirements collection table. (see appendix)

5.1.12 Scenario 17: Functional and Technical Safety Concept including analysis and verification

Scenario-ID: S17	CONTACT PERSON: Hans-Leo Ross, Jan Hoffmann

SCENARIO NAME:

Functional and Technical Safety Concept including analysis and verification according ISO 26262 of an integrated brake system, including acceptance criterion for Functional Safety Assessment.

Starting point of the process step that is under consideration:

Item Definition, Preliminary System Architecture

End point of the process step that is under consideration:

Acceptance Criterion for Integrated Brake System as Part of the Public Road Release for Mass Production Design

LINK TO Validation Task 5.x:

Development of an Integrated Brake System

SCENARIO RELEVANCE:

WP2.1/ WP2.3/ WP3.1.1/ WP3.1.2 /WP3.1.3/WP3.2.1 / WP3.3.1 / WP5.1/WP5.2/WP6.1 / W6.2

SCENARIO JUSTIFICATION:

The development process of safety brake systems based on proven principles and legal requirements is defined in ECE R13H. In order to define acceptance criterion for the design and its verification, ISO 26262 becomes “State of Science and Technology” for brake systems.

The following steps need to be considered:

Requirement Elicitation: Who are the technical stakeholders of a brake system

Requirement Analysis: How to derive requirements from multiple Safety Goals for a complex item based on an array of systems.

The challenge is to define methodology for the functional and technical architecture and its dependability and a systematic approach for their verification and analysis by considering Part 9 of ISO 26262.

Based on the functional and technical assumptions a requirement management strategy shall be defined, so that a systematic deriving of requirements could assure vertical (from Safety Goal to parts or SW units) and horizontal (from stakeholder requirement to Test (verification and validation)) traceability.

A meaningful combination of deductive and inductive analysis should lead to sufficient transparency to assess Functional Safety and fulfillment of stakeholder requirements.

SCENARIO ACTIVITY:

Main activities associated to this scenario will consider the relevant results achieved in WP3 and are described as follows:

1. Identification of requirements from ISO 26262 relevant for “Safe” and considering

other legal requirements or other sources of requirements, which demonstrate "State of Science and Technology" (Task 2.1: ISO 26262 Analysis)

2. Defining Use Case Scenario relevant for "Safe" (Task 2.1: Use Case Scenario)
3. Definition of a methodology to derive a Functional and Technical Safety Concept based on assumptions for an item and given Safety Goals.
(Task 3.1.1: Hazard analysis, safety goal and ASIL definition).
4. Definition of interfaces for requirements, architecture, design and their verification. So that tools could be selected to support the necessary safety activities (Task 3.1.2: Safety Requirement Expression).
5. Generation of iteration to systematically develop work-products as part of the safety case (Task 3.1.3: Safety Case Documentation).
6. Definition of a generic model describing relation between functional and technical architecture, deriving relevant requirements and criterion for their verification and analysis. The model shall be applicable for System, mechanic, E/E hardware, software and functional structure in silicon. (Task 3.2.1: System and SW Models Enhancement).
7. The quantitative analysis shall be considered. The impact of systematic faults to derive values for the quantified fault mode and their probability of violating the safety goal shall be analysed. An adequate methodology to analyse cutsets shall be defined. (Task 3.3.1: Failure and cut-set analysis).
8. Definition of key criterion to assess Functional Safety of an Integrated Brake System (Task 5.1: Use Case Assessment Criterion).
9. Definition of a validation strategy for an Integrated Brake System. Define a systematic approach to assess the fulfilment of the relevant Safety Goals and achieve a "Public Road Release" (Task 5.2: Use Cases Continental).
10. Definition of methodologies and application rules experiences gained from the considered WTs (Task 6.1, Task 6.2).

Output Artifacts

Output documents and work products of the process step(s) under consideration are

- Sources of safety relevant requirements relevant for Integrated Brake System.
- Criterion for a methodology to develop functional and technical architecture
- Criterion for a methodology to develop Functional and Technical Safety Concept and their verification.
- Criterion for a methodology for deductive and inductive analysis
- Assessment criterion and application rules for "Functional Safety Assessment"

Requirements generated for SAFE

Requirements S17-001 - S17-006 in Requirements collection table. (see appendix)

5.1.13 Scenario 18: Integration of safety-related and none safety-related software

SCENARIO ID: S18	CONTACT PERSON: Bernhard Rumpler, Andreas Eckel
SCENARIO NAME: Integration of safety-related and none safety-related software	
Starting point of the process step that is under consideration: Specification of software (safety) requirements	
End point of the process step that is under consideration: Verification of software (safety) requirements	
LINK TO Validation Task 5.x: WT 5.3	
SCENARIO RELEVANCE: WT 3.3.2, WT 4.4	
SCENARIO JUSTIFICATION: There exists a lot of software for automotive ECUs that are not developed according to ISO26262 requirements (e.g., AUTOSAR basic software, legacy application functions). This QM software needs to be integrated with safety-related software components that are developed according to ISO26262. There exists no current practice how this can be achieved in a generic way. This scenario describes the typical activities that have to be performed by the integrator making use of the generic software “safety layer” that is proposed by TTTech.	
SCENARIO ACTIVITY: Specify software safety requirements and software architectural design • Allocate requirements for freedom from interference between QM and ASIL components to dedicated, generic software components (“safety layer”) • Specify requirements resulting from integration of “safety layer” (e.g., specific mechanisms that have to be used by supervised entities) Perform software unit design and implementation • Use mechanisms of safety layer in supervised software (e.g., add checkpoints) Perform software integration and testing • Configure AUTOSAR basic software and “safety layer” with configuration tools • Integrate all software components and test integrated components (especially correct function of safety mechanisms defined in the architecture) Perform verification of software safety requirements • Perform requirements-based test of software safety requirements (especially those allocated to the “safety layer”)	

Output Artifacts The following work products are usually created by the integrator: Specification of software safety requirements Software architectural design specification (describing the software architecture with the integrated safety layer) Configuration data for BSW and safety layer Integrated software Verification reports for integrated software (resulting from software integration and testing and verification of software safety requirements).
Requirements generated for SAFE Requirements S18-001 - S18-002 in Requirements collection table. (see appendix)

5.1.14 Scenario 19-1: Safety case contents

SCENARIO ID: S19-1	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas

SCENARIO NAME:

Safety case contents

Starting point of the process step that is under consideration:

Overall safety lifecycle

End point of the process step that is under consideration:

Overall safety lifecycle

LINK TO Validation Task 5.x:

There is no validation task linked directly to this scenario. Validation will be performed based on concept document.

SCENARIO RELEVANCE:

Task 3.1: Safety goals Modeling

Task 3.1.1: Hazard analysis, safety goal and ASIL definition

Task 3.1.2: Safety Requirement Expression

Task 3.1.3: Safety Case Documentation

Task 3.5 : Meta Model Definition

SCENARIO JUSTIFICATION:

Safety goals are top level safety requirements on vehicle level.

Safety requirements can occur on functional (what) and technical (how) levels of detail.

Safety arguments communicate the relationship between evidences and safety objectives.

A safety argument can be technical (e.g. the behavior of a timing watchdog as a constructive safety measure) or process-based (e.g. MC/DC coverage).

Safety cases are communicated through the development and presentation of safety case reports. The role of a safety case report is to summarize the safety argument and then reference the reports capturing the supporting safety evidence (e.g. test reports). The safety case documents how individual requirements are supported by specific arguments, how arguments are supported by evidence and the assumed context that is defined for the argument.

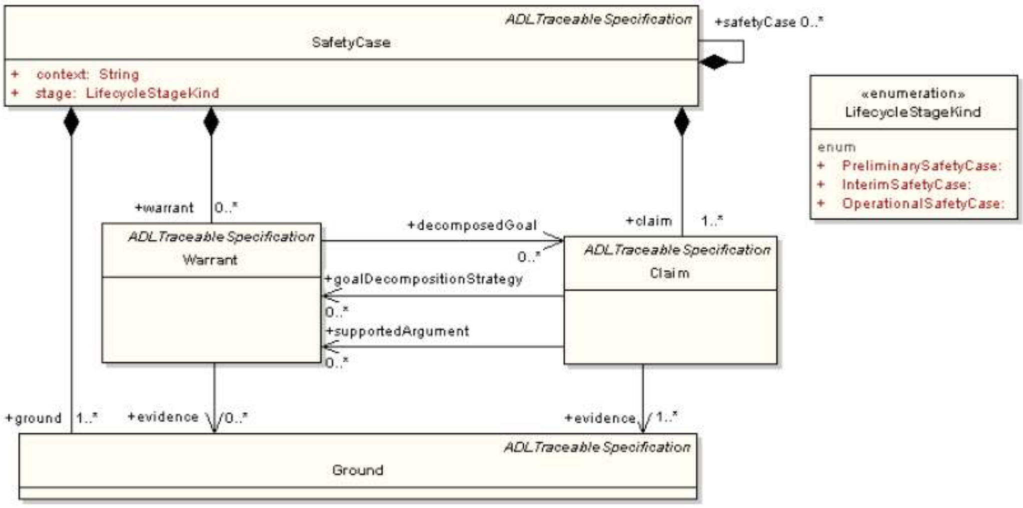
Evidence is structured verification and validation information (tests, analysis, reviews, etc.) in form of development artifacts.

Context needs to be verified and validated to apply the safety case in a safety assessment. The necessary context is the sum of the safety relevant aspects of the system scope (environment). Context may change.

SCENARIO ACTIVITY:

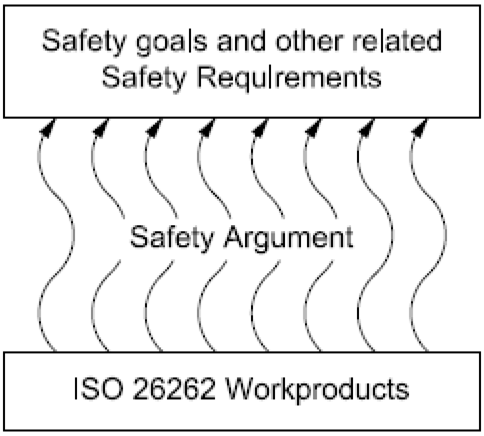
Output Artifacts

Enhanced Meta-Model



© MAENAD concept presentation, Dependability Analysis, 2011 Q3

©Refer to ISO 26262, part 10 **Fehler! Verweisquelle konnte nicht gefunden werden.** for structure of safety case (proposal):



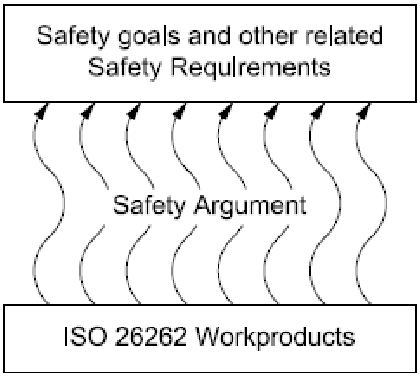
© ISO 26262

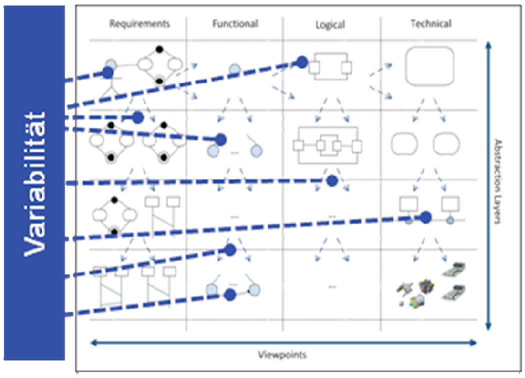
Requirements generated for SAFE

Requirements S19-001 - S19-04 in Requirements collection table. (see appendix)

5.1.15 Scenario 19-2: Variability-aware Safety Case

SCENARIO ID: S19-2	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Variability-aware Safety Case	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1: Safety goals Modeling Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.1.2: Safety Requirement Expression Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition Task 3.4: Variant Management	
SCENARIO JUSTIFICATION: Variability is a major concern for the automotive domain. Modular Safety Cases must support Variant Management because separation of variant and invariant parts enables reusability.	
SCENARIO ACTIVITY: Identify which safety goals and safety requirements, safety concepts and safety analyses are impacted by variants.	
Output Artifacts Enhanced Meta-Model with support of Variant Management, at a minimum, addressing H+R, FSK, TSK and all relevant safety case artifacts including verification, validation as well as confirmation measures.	





© ISO 26262, part 10

<u>Requirements generated for SAFE</u>

Requirements S19-005 - S19-006 in Requirements collection table. (see appendix)

5.1.16 Scenario 19-3: Safety Case in distributed development

SCENARIO ID: S19-3

CONTACT PERSON:

Adaptation to SAFE: Jürgen Lucas

SCENARIO NAME:

Safety Case in distributed development (OEM / Tier-1)

Starting point of the process step that is under consideration:

Overall safety lifecycle

End point of the process step that is under consideration:

Overall safety lifecycle

LINK TO Validation Task 5.x:

There is no validation task linked directly to this scenario. Validation will be performed based on concept document.

SCENARIO RELEVANCE:

Task 3.1.3: Safety Case Documentation

Task 3.3.2: Safety Evaluation

Task 4.2.1: Plug-in for traceability and requirement import

SCENARIO JUSTIFICATION:

Safety artifacts are work products which are produced during the safety lifecycle and are necessary to assess the functional safety of an item.

Besides reusability of safety artifacts within one single corporation across divisions and engineering roles, the top benefit of a Modular Safety Case is its support for distributed development across different companies and organizations.

SCENARIO ACTIVITY:

Define project scope in terms of model-based item definition.

Identify work break down structure based on the model.

Define Development Interface Agreement (DIA) structure and contents in the model (e.g. as packages that can be integrated)

Perform safety engineering activities (H+R, FSC, TSC, FTA, FMEA, etc.) in a distributed development but aggregate the resulting safety relevant information in the model defined packages (artifacts).

Output Artifacts

Instead of the traditional text based Development Interface Agreement (DIA) between OEMs and Tier-1s, the Modular Safety Case is based on a model-based Safety Case Architecture which defines interfaces between safety model elements (XML).

Not all DIA contents need to be addressed in the model, e.g. only safety analysis and safety requirements (concepts)

Requirements generated for SAFE

Requirements S19-007 - S19-009 in Requirements collection table. (see appendix)

5.1.17 Scenario 19-4: Safety Case notation

SCENARIO ID: S19-4	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Case notation	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1: Safety goals Modeling Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.1.2: Safety Requirement Expression Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: It is becoming increasingly popular to use graphical argument notations (such as Claims–Argument–Evidence and the Goal Structuring Notation [2]) to visually and explicitly represent the individual elements of a safety argument (requirements, claims, evidence and context) and the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument)	
SCENARIO ACTIVITY:	
Output Artifacts Goal Structure Notation (GSN) Safety Specification and Assurance Language (TBD)	
Requirements generated for SAFE Requirements S19-010 in Requirements collection table. (see appendix)	

5.1.18 Scenario 19-5: Model-based safety engineering and integration …

SCENARIO ID: S19-5	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Model-based safety engineering and integration across system abstraction levels	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1: Safety goals Modeling Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.1.2: Safety Requirement Expression Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: Top down and bottom up development must be capable to be integrated acc. to overall safety properties. Integration of Safety Elements out of Context (SEooCs) into a model-based Safety Case must be possible.	
SCENARIO ACTIVITY:	
Output Artifacts Safety Case interfaces Modular Safety Cases	
Requirements generated for SAFE Requirements S19-011 - S19-013 in Requirements collection table. (see appendix)	

5.1.19 Scenario 19-6: Model-based and compoundable Safety Concepts

SCENARIO ID: S19-6	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas

SCENARIO NAME:

Model-based and compoundable Safety Concepts

Starting point of the process step that is under consideration:

Overall safety lifecycle

End point of the process step that is under consideration:

Overall safety lifecycle

LINK TO Validation Task 5.x:

There is no validation task linked directly to this scenario. Validation will be performed based on concept document.

SCENARIO RELEVANCE:

Task 3.1: Safety goals Modeling

Task 3.1.1: Hazard analysis, safety goal and ASIL definition

Task 3.1.2: Safety Requirement Expression

Task 3.2.1: System and software models enhancement

Task 3.5 : Meta Model Definition

SCENARIO JUSTIFICATION:

Safety concepts are a required work product in an ISO 26262 safety case.

Safety concepts define all measures against safety-critical failures and the allocation of safety requirements to the architecture elements.

Main aspects of safety concepts:

- Integration Systems Engineering / Functional Safety
- Clear separation of Functional / Technical Safety Concept
- Safety Functions as Entry Points for Technical Safety Concept
- Stepwise Refinement with Traceability
- Structuring the TSC according Technical Architecture
- Relation Matrix of Failure Modes vs. Safety Mechanisms
- Establishing Safety Patterns
- Separation between different functions on same ECU

Functional Safety Concept needs not to be changed when implementation details change or variants are developed (possibility of reuse).

FSC: Represents the set of functional safety requirements allocated to the (preliminary) architectural elements that fulfill one or more safety goals. Each safety requirement may include:

- ASIL – a Safety Constraint associated with the requirement
- Operating Modes
- Fault Tolerant Time Spans
- Safe States
- Emergency Operating Times
- Functional Redundancies
- Specifications on how fault tolerance is achieved
- Acceptance criteria

TSC: Contains the technical safety requirements. Details the functional safety concept in the context of the architectural design

<u>SCENARIO ACTIVITY:</u>

<u>Output Artifacts</u>

The Functional Safety Concept (FSC) is realized by allocating functional safety requirements to the logical architecture from the logical viewpoint of the SPES2020 meta-model. The Technical Safety Concept (TSC) is realized by deriving technical safety requirements from the functional safety requirements and by decomposing the logical architecture into a technical architecture in the technical viewpoint of the SPES2020 meta-model.

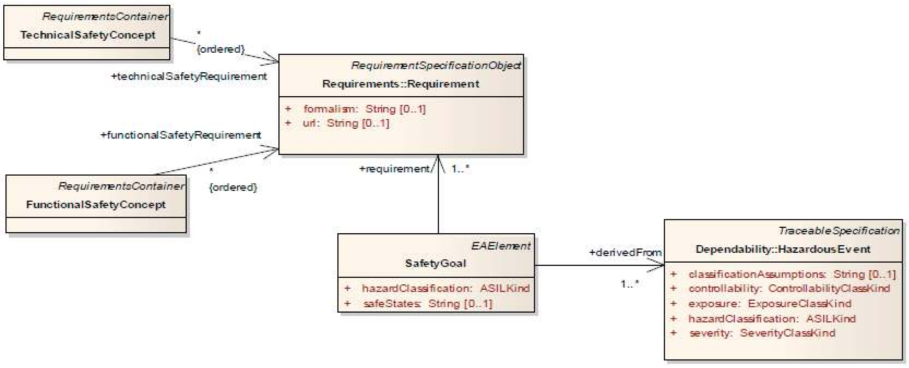
Assume/Guarantee Contracts on interfaces between FSC and TSCs. Minimum contents of Functional and Technical Safety Concepts acc. to ISO 26262.

Hierarchical model-based Safety Concept

Example:

Fault tolerance time modeling across different model layers with SysML / UML diagram types.

Refer to MAENAD research project:


RequirementsContainer
TechnicalSafetyConcept
*
{ordered}
+technicalSafetyRequirement
RequirementSpecificationObject
Requirements::Requirement
+ formalism: String [0..1]
+ url: String [0..1]
+functionalSafetyRequirement
RequirementsContainer
FunctionalSafetyConcept
*
{ordered}
+requirement
1..*
EAElement
SafetyGoal
+ hazardClassification: ASILKind
+ safeStates: String [0..1]
+derivedFrom
1..*
TraceableSpecification
Dependability::HazardousEvent
+ classificationAssumptions: String [0..1]
+ controllability: ControllabilityClassKind
+ exposure: ExposureClassKind
+ hazardClassification: ASILKind
+ severity: SeverityClassKind

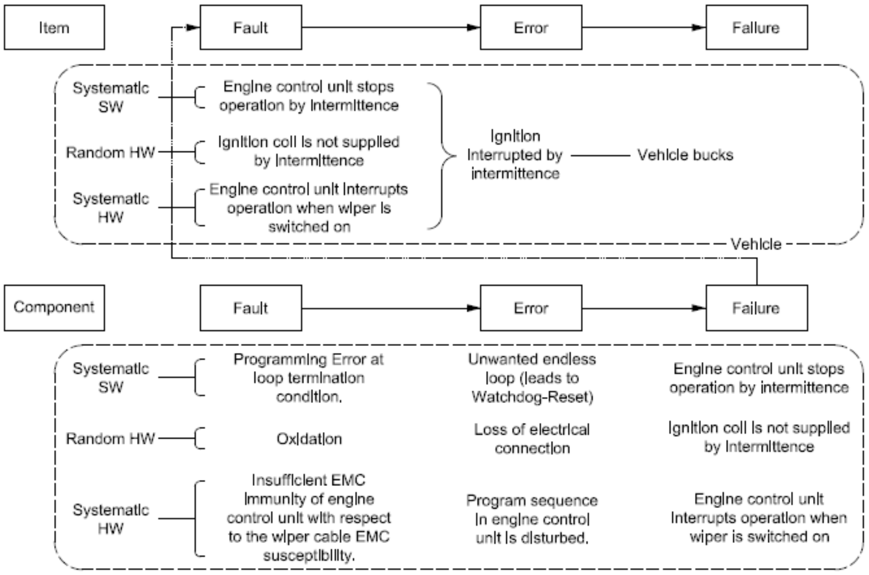
© MAENAD concept presentation, Dependability Analysis, 2011 Q3

<u>Requirements generated for SAFE</u>

Requirements S19-014 - S19-016 in Requirements collection table. (see appendix)

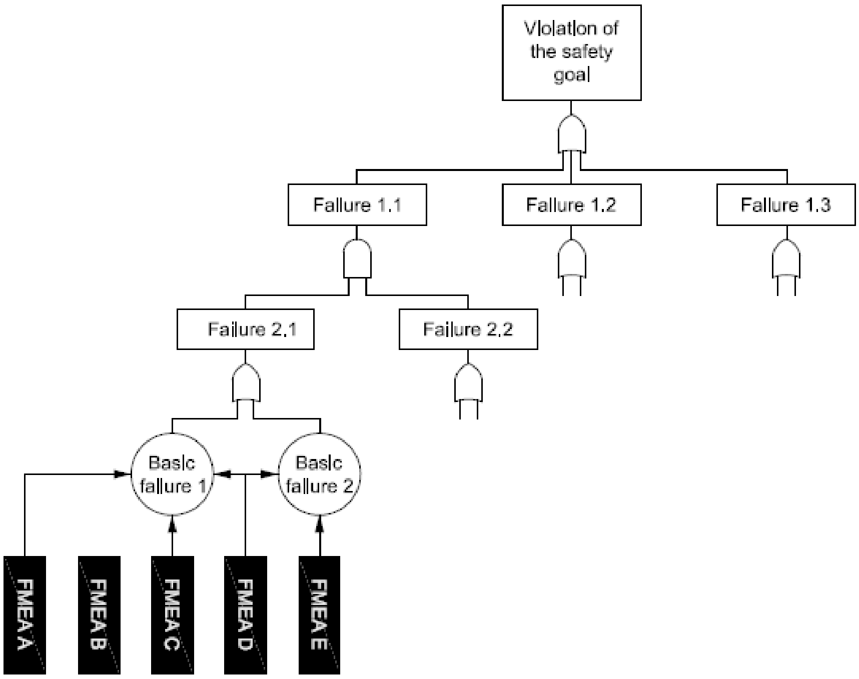
5.1.20 Scenario 19-7: Combined Safety Analysis in one Safety Model

SCENARIO ID: S19-7	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Combined Safety Analysis in one Safety Model	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition Task 3.3.1: Failure and Cut-sets Analysis	
SCENARIO JUSTIFICATION: Analysis must be done in both ways inductive (FMEA) and deductive (FTA). Safety Goals from a Hazard and Risk Analysis (H+R) are the top events of the FTA (C2FT) on functional level. This relates H+R and traditional safety analysis methods.	
SCENARIO ACTIVITY:	
Output Artifacts Single Fault view over different model layers (Item – Component):	


Item
Fault
Error
Failure
Systematic SW
Engine control unit stops operation by intermittence
Random HW
Ignition coil is not supplied by intermittence
Systematic HW
Engine control unit interrupts operation when wiper is switched on
Ignition interrupted by intermittence
Vehicle bucks
Vehicle
Component
Fault
Error
Failure
Systematic SW
Programming Error at loop termination condition.
Unwanted endless loop (leads to Watchdog-Reset)
Engine control unit stops operation by intermittence
Random HW
Oxidation
Loss of electrical connection
Ignition coil is not supplied by intermittence
Systematic HW
Insufficient EMC immunity of engine control unit with respect to the wiper cable EMC susceptibility.
Program sequence in engine control unit is disturbed.
Engine control unit interrupts operation when wiper is switched on

© ISO 26262, part 10

Combined FTA and FMEA:



© ISO 26262, part 10

Systems are composed of many parts and sub-parts. FTA and FMEA can be combined to provide the safety analysis with the right balance of top-down and bottom-up approach. Figure 2 shows a possible approach to combine an FTA with an FMEA. In this figure, the basic events are derived from different FMEA (labeled FMEA A-E within this example) which are done on a lower level of abstraction (e.g. sub-part, part or component level). Within this example, FMEA B does not impact basic events 1 and 2, while FMEA D impacts both.

Requirements generated for SAFE

Requirements S19-017 in Requirements collection table. (see appendix)

5.1.21 Scenario 19-8: Model-based Safety Patterns

SCENARIO ID: S19-8	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Model-based Safety Patterns	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition Task 3.1.3: Safety Case Documentation Task 3.6: Safety code Generation	
SCENARIO JUSTIFICATION: The issue of a generalizable, reusable representation of safety properties is a research topic. Several safety analysis methods use a compositional approach to determine the causes of potential hazards. The results of such analyses on simple components can often be generalized to enable their reuse across designs, where such reuse is possible. A greater level of flexibility can be achieved by generating context-free generalized expressions that describe failure behavior independently of the architecture of the component. Such expressions are better suited to being used as part of an iterative design process, preventing the need to re-annotate components in every annotation. They can also be more easily reused across applications where the same component may receive different types and number of inputs, or (more generally) where components have the same functionality but different interfaces.	
SCENARIO ACTIVITY:	
Output Artifacts Template for safety model elements (safety patterns) of reusable combinable components.	
Requirements generated for SAFE Requirements S19-018 in Requirements collection table. (see appendix)	

5.1.22 Scenario 19-9: Modular Hazard Analysis on model-based function or system

SCENARIO ID: S19-9	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Modular Hazard Analysis on model-based function or system (item definition)	
Starting point of the process step that is under consideration: Item definition initial version	
End point of the process step that is under consideration: Functional safety concept version n released	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1: Safety goals Modeling Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: The objective of the hazard analysis and risk assessment is to identify and to categorize the hazards that malfunctions in the item can trigger and to formulate the safety goals related to the prevention or mitigation of the hazardous events, in order to avoid unreasonable risk. Consistency between traditional Excel-based H+R and a separate model-based Item Definition or a separate textual Item Definition is weak if the preliminary architecture or function definition suffers from a high rate of changes in early project phases. Moreover it is necessary to analyze hazards that affect different items or vehicle functions.	
SCENARIO ACTIVITY:	
Output Artifacts	
Requirements generated for SAFE Requirements S19-019 – S19-026 in Requirements collection table. (see appendix)	

5.1.23 Scenario 19-10: Synchronization between H+R and Item Definition

SCENARIO ID: S19-10	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Synchronization between Hazard and Risk Analysis (H+R) and Item Definition (System Definition)	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: Changes in the system item definition shall be visible in an updated H+R. The focus should lie on synchronization between functional changes including their consequence for modified malfunctions for potential modified hazards.	
SCENARIO ACTIVITY:	
Output Artifacts	
Requirements generated for SAFE Requirements S19-027 in Requirements collection table. (see appendix)	

5.1.24 Scenario 19-11: Consistency checks between H+R and Item Definition

SCENARIO ID: S19-11	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Consistency checks between Modular Hazard and Risk Analysis (H+R) and model-based Item Definition	
Starting point of the process step that is under consideration: Concept phase	
End point of the process step that is under consideration: Development phase	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: Consistency between traditional EXCEL-based H+R and a separate model-based Item Definition or a separate textual Item Definition is weak if the preliminary architecture or function definition suffers from a high rate of changes in early project phases. Moreover it is necessary to analyze hazards originating from a specific actuator which is affected by different items or vehicle functions.	
SCENARIO ACTIVITY: Check if function has been changed (or renamed). Check if function is allocated to different actuator. Check if a new function is allocated to an actuator which has already existing functions allocated.	
Output Artifacts	
Requirements generated for SAFE Requirements S19-028 in Requirements collection table. (see appendix)	

5.1.25 Scenario 19-12: Comparability of Hazard and Risk Analysis (H+R)

SCENARIO ID: S19-12	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Comparability of Hazard and Risk Analysis (H+R)	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.2.1: System and software models enhancement Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: For safety engineers it is very helpful to compare between H+R analysis and related functions.	
SCENARIO ACTIVITY: Check if function has been changed (or renamed). Check if function is allocated to different actuator. Check if a new function is allocated to an actuator which has already existing functions allocated.	
Output Artifacts	
Requirements generated for SAFE Requirements S19-029 in Requirements collection table. (see appendix)	

5.1.26 Scenario 19-13: Guided Hazard and Risk Analysis (H+R)

SCENARIO ID: S19-13

CONTACT PERSON:

Adaptation to SAFE: Jürgen Lucas

SCENARIO NAME:

Guided Hazard and Risk Analysis (H+R)

Starting point of the process step that is under consideration:

Concept phase

End point of the process step that is under consideration:

Concept phase

LINK TO Validation Task 5.x:

There is no validation task linked directly to this scenario. Validation will be performed based on concept document.

SCENARIO RELEVANCE:

Task 3.1.1: Hazard analysis, safety goal and ASIL definition

Task 3.2.1: System and software models enhancement

Task 3.5 : Meta Model Definition

SCENARIO JUSTIFICATION:

Hazards originate mostly from actuators on system boundaries.

It would be helpful to have some sort of pattern-based hazard descriptions and generic failure modes. In combination with domain ontology they could be used for semi-automatic identification of hazards.

SCENARIO ACTIVITY:

Ontology based inference mechanisms can support a guided H+R. Hazard analysis ontology can help to deduce implicit knowledge about hazards from already existing explicitly modeled functions, malfunctions, and hazards.

Generic hazard lists can be useful as starting point for the explicitly modeled hazard.

In “Patterns in safety analysis”, Stalhane, T., Daramola, O., and Katta, V. develop an approach which enables semi-automatic generation of the complete H+R table based on system’s functional requirements, pattern based hazard descriptions and domain knowledge formalized as domain ontology. (Refer to CESAR project).

Output Artifacts

Requirements generated for SAFE

Requirements S19-030 – S19-031 in Requirements collection table. (see appendix)

5.1.27 Scenario 19-14: Safety Case properties

SCENARIO ID: S19-14	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Case properties	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.2: Safety Requirement Expression Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: Completeness can be assessed by showing that each safety goal results in safety requirements which are implemented by safety measures and a corresponding safety argumentation (claim) that is based on evidence in terms of work products (development artifacts). Consistency addresses vertical (left side of V-Model, requirements, architecture, code) and horizontal (right side of V-Model) traceability of safety case contents and absence of contradictions or gaps. Modularity refers to the logical partitioning /compartmentalization and the inter-relation of safety case contents. Modularity allows changes to be manageable. Comparability can be realized by a defined model-based format and safety terminology. Reusability can be achieved by modularity which allows reusing parts of a safety case respectively its artifacts by composition and decomposition along defined interfaces.	
SCENARIO ACTIVITY:	
Output Artifacts Structured information management (model-based) can be used as part of a safety argument in a safety case and supports systematic safety/reliability analysis.	
Requirements generated for SAFE Requirements S19-032 in Requirements collection table. (see appendix)	

5.1.28 Scenario 19-15: Supported Analysis on Safety Case Contents

SCENARIO ID: S19-15	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Supported Analysis on Safety Case Contents	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.3: Safety Case Documentation Task 3.3.1: Failure and Cut-sets Analysis Task 3.3.2: Safety Evaluation	
SCENARIO JUSTIFICATION: The Modular Safety Case shall contain annotated information (or metadata) to support assessments and analysis on contents, at a minimum for: • finding safety requirements that are not allocated to a model element and vice versa • finding model elements which have safety requirements with mixed ASILs / safety criticality allocated and vice versa • finding evidence that supports valid ASIL decompositions in terms of requirements which detail redundancy or safety mechanisms acc. to independency with respect to the same safety goal • comparing safe states of different items which share safety model elements	
SCENARIO ACTIVITY:	
Output Artifacts	
Requirements generated for SAFE Requirements S19-033 – S19-034 in Requirements collection table. (see appendix)	

5.1.29 Scenario 19-16: Safety Case analysis due to context modifications

SCENARIO ID: S19-16	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Case analysis due to context modifications (Change Impact Analysis)	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.3: Safety Case Documentation Task 3.3.1: Failure and Cut-sets Analysis Task 3.3.2: Safety Evaluation Task 3.3.3: Safety and Multi Criteria Architecture Benchmarking Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: Functional changes (e.g. additional emergency brake mode) can result in necessary modifications to the Hazard and Risk Analysis (H+R) Logical changes (e.g. diverse implemented state observer) can result in modifications on the Functional Safety Concept (FSC), whereas technical changes (e.g. dual-core lockstep CPU or new compiler library) might result in a modified Technical Safety Concept (TSC).	
SCENARIO ACTIVITY:	
Output Artifacts Safety relevant context changes can basically result in modified (added, reduced) • faults (causes), • failure propagation possibilities (consequences), or • safety mechanisms or combinations of all three safety related change impacts with respect to the unchanged safety model elements. By simple Configuration Management and traceability mechanisms the affected elements of the safety case can be identified after the change has been modeled, but only if the model elements are modular enough in terms of configuration elements.	

Dynamic Checklists can be used based on the type of context change to ease the safety engineer with the change impact analysis.

SysML context diagram (use case or internal block diagram)

Requirements generated for SAFE

Requirements S19-035 – S19-037 in Requirements collection table. (see appendix)

5.1.30 Scenario 19-17: Safety Case incremental compilation/development

SCENARIO ID: S19-17	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Case incremental compilation/development	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1: Safety goals Modeling Task 3.1.1: Hazard analysis, safety goal and ASIL definition Task 3.1.2: Safety Requirement Expression Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: The development of a safety case can be treated as an incremental activity that is integrated with the rest of the development phases of the safety lifecycle. Such an approach allows intermediate versions of the safety case at given milestones of the product development. For example, a preliminary version of the safety case can be created after the verification of the technical safety requirements; an interim version of the safety case can be created after the verification of the system design; and the final version can be created just prior to the functional safety assessment. Traditionally, the Safety Plan specifies the safety activities (development phases, inputs, outputs, roles, etc.) in which artifacts are generated that build up the safety case.	
SCENARIO ACTIVITY: Model-based Safety Case refinement can be realized by incremental development.	
Output Artifacts	
Requirements generated for SAFE Requirements S19-038 – S19-039 in Requirements collection table. (see appendix)	

5.1.31 Scenario 19-18: Safety Case incremental assessment

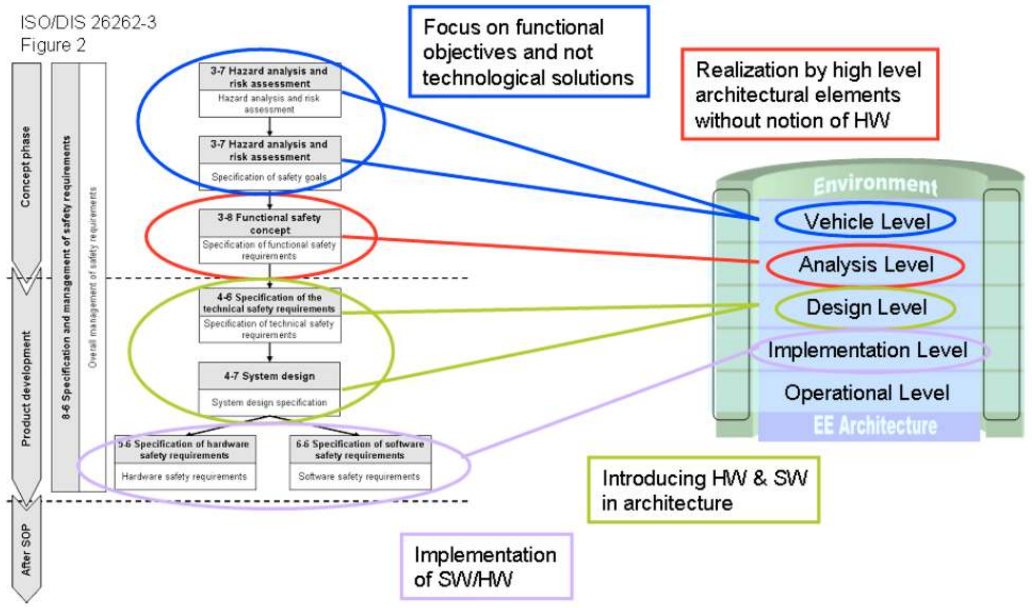
SCENARIO ID: S19-18	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Case incremental assessment	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition Task 3.3.2: Safety Evaluation	
SCENARIO JUSTIFICATION: It is vital for the safety manager to identify ambiguities and inconsistency concerning safety aspects as soon as possible, especially in a distributed development.	
SCENARIO ACTIVITY:	
Output Artifacts	
Requirements generated for SAFE Requirements S19-040 in Requirements collection table. (see appendix)	

5.1.32 Scenario 19-19: Safety Model Interoperability with various Modeling Tools (XML)

SCENARIO ID: S19-19	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Model Interoperability with various Modeling Tools (XML)	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 4.1: Meta Model Implementation Task 4.2.1: Plug-in for traceability and requirement import Task 4.2.2: Plug-in for behavioral translator Task 4.2.3: Plug-in for failure and cut-sets analysis	
SCENARIO JUSTIFICATION: The SAFE safety model should be based on a specific (TBD) subset of UML/SysML diagrams and stereotypes in order to be interoperable across tools which support this subset of basic diagram types. Try to avoid nice but proprietary tool features which are not standard UML/SysML.	
SCENARIO ACTIVITY:	
Output Artifacts	
Requirements generated for SAFE Requirements S19-041 – S19-046 in Requirements collection table. (see appendix)	

5.1.33 Scenario 19-20: Safety Model Abstraction Levels

SCENARIO ID: S19-20	CONTACT PERSON: Adaptation to SAFE: Jürgen Lucas
SCENARIO NAME: Safety Model Abstraction Levels	
Starting point of the process step that is under consideration: Overall safety lifecycle	
End point of the process step that is under consideration: Overall safety lifecycle	
LINK TO Validation Task 5.x: There is no validation task linked directly to this scenario. Validation will be performed based on concept document.	
SCENARIO RELEVANCE: Task 3.1.2: Safety Requirement Expression Task 3.2.1: System and software models enhancement Task 3.1.3: Safety Case Documentation Task 3.5 : Meta Model Definition	
SCENARIO JUSTIFICATION: The SPES method shall support safety modeling on different abstraction levels: • Vehicle level (Hazard and Risk analysis, H+R) • Analysis level (Functional Safety Concept, FSC) • Design Level (Technical Safety Concept, TSC) • Implementation Levels (SW and HW Safety Requirements) • Operational Level (Context)	
SCENARIO ACTIVITY:	
Output Artifacts SPES 2020 abstraction levels and viewpoints and EAST-ADL mapping	


ISO/DIS 26262-3
Figure 2
Concept phase
Product development
After SOP
8-6 Specification and management of safety requirements
Overall management of safety requirements
3-7 Hazard analysis and risk assessment
Hazard analysis and risk assessment
3-7 Hazard analysis and risk assessment
Specification of safety goals
3-8 Functional safety concept
Specification of functional safety requirements
4-6 Specification of the technical safety requirements
Specification of technical safety requirements
4-7 System design
System design specification
5-6 Specification of hardware safety requirements
Hardware safety requirements
6-6 Specification of software safety requirements
Software safety requirements
Focus on functional objectives and not technological solutions
Realization by high level architectural elements without notion of HW
Environment
Vehicle Level
Analysis Level
Design Level
Implementation Level
Operational Level
EE Architecture
Introducing HW & SW in architecture
Implementation of SW/HW

©, refer to MAENAD concept presentation, Dependability Analysis, 2011 Q3 research project

Requirements generated for SAFE

Requirements S19-047 – S19-048 in Requirements collection table. (see appendix)

5.1.34 Scenario 20: Verification of software behavior

SCENARIO ID: S20

CONTACT PERSON:

Alexander Griessing

SCENARIO NAME:

Verification of software behavior and the effectiveness of implemented safety measures in the presence of faults injected into the microcontroller hardware

Starting point of the process step that is under consideration:

Specification of software (safety) requirements

End point of the process step that is under consideration:

Verification of software (safety) requirements

LINK TO Validation Task 5.x:

WT 5.4, WT 5.3

SCENARIO RELEVANCE:

WT 3.3.2: Safety evaluation

WT 4.4: Implementation of software platform for mixed criticality

SCENARIO JUSTIFICATION:

The verification of software (and system) behavior in the presence of faults in the underlying hardware (e.g. microcontroller) is mandated by the ISO 26262. This includes the verification of safety requirements towards a deterministic behavior or graceful degradation in case of hardware faults, as well as the verification of the effectiveness of implemented safety measures to detect and mitigate such hardware faults. Fault injection, though an accepted methodology for this kind of verification is often hindered by the lack fault injection mechanisms and interfaces in the hardware or hardware models.

This scenario describes and demonstrates how fault injection verification can be done with the help of microcontroller C-models. The verification is carried out on the example of AUTOSAR MCAL driver software, but the same methodology can be applied similarly to more complex software and system designs.

SCENARIO ACTIVITY:

Specification and implementation of fault injection interfaces for the C-models:

- Determine relevant failure modes for individual microcontroller hardware parts (e.g. CPU core, memories, busses, peripherals)
- Specify and implement interfaces for fault injection to trigger the failure mode

Specification of software safety concept and software architectural design for the AUTOSAR MCAL driver software:

- Specify assumptions about scope, usage and safety requirements of the software, and derive conditions-of-use requirements

- Perform software partitioning, ASIL decomposition, safety requirement allocation and ASIL allocation
- Specify software safety mechanisms for error detection and error handling (errors due to random hardware failures as well as software faults), and to achieve freedom from interference for sufficient independence between software units
- Perform safety analysis to verify the effectiveness of the specified software safety mechanisms

Software unit design and implementation, unit testing, software integration and testing

- Design, implementation and testing according to requirements of ISO 26262 and the allocated ASIL

Verification of software safety requirements

- Integration of the AUTOSAR MCAL drivers with an example application software
- Execution of the integrated software on the microcontroller C-models
- Fault injection into the C-models
- Requirement based verification of the software behavior and the effectiveness of the implemented software safety mechanisms

Output Artifacts

- Software safety concept and software architecture design specification for AUTOSAR MCAL drivers
- Safety analysis report
- Verification report from fault injection verification
- AUTOSAR MCAL driver software

Requirements generated for SAFE

Requirements S20-001 – S20-002 in Requirements collection table. (see appendix)

5.2 Methods

Method Identification	Method Name	~~Class~~
M01	Conformance check	~~Method Description~~
M02	AltaRica	~~Method Description~~
M03	Graphical metric definition and evaluation.	~~Method Description~~
M08	Design Space Exploration	~~Method Description~~
M09	Model Transformations	~~Method Description~~
M10	Integrated System Architecture and Dependability Modeling with EAST-ADL2	~~Method Description~~
M12	Contract based evaluation of safety functions	~~Method Description~~
M13	Uniform Modeling of Variability in Automotive System Architectures	~~Method Description~~

5.2.1 Method Template

In order to describe and analyze the use methods, a template has been defined as follows:

Method ID: Mxx <Identification number (see overview table)>	CONTACT PERSON: <Name of contact person(s) (initiators)>
METHOD NAME: <Name of the method>	
METHOD RELEVANCE: Indicate in which Task of WP3 the activities of this methods is related:	
LINK TO Reference Tool Platform (RTP): Indicate to which tool development for WP4 this methods could attached	
METHOD JUSTIFICATION: Please explain progress being state of art (and existing tool) and quantitative measure of success	
METHOD ACTIVITY: Please explain the main activities and major steps of the method and techniques. Be sure to indicate potential interaction with others methods not targeted.	
REFERENCE META-MODEL IMPACT Describe briefly what would be the necessary information required in the reference meta-model from WT3.5 required to support this method	
LINK TO RTP RESULTS Describe how your methods could be implemented in RTP platform from WP4 and what tool development is planned (indicate tools you intent to perform/investigate/develop and identify gap if necessary)	
SKILLS REQUIREMENT: Please explain the skills you will require to perform your activities.	
Requirements generated for SAFE Collection of requirements that are elicited in this scenario (will be further handled in parallel to the requirements coming from WT2.1 and WT2.2) Numbering Scheme: <Scenario-ID(fixed)>_<local number> (Link in excel-Table)	

5.2.2 Method 01: Conformance check

Method ID: M01

CONTACT PERSON:

Adaptation to SAFE : Stefan Voget

METHOD NAME:

Conformance check

METHOD RELEVANCE:

The activities of this method are related to …

3.2.4 (guideline for conformity check)

3.3.3 (Methodology for conformity check)

3.3.4 (metrics for benchmarking)

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 – deleted with Change Request 2 of the FPP. Therefore, no longer a mapping possible

4.2.3 (Plug-in for conformance checker)

4.3 (Integrated platform)

METHOD JUSTIFICATION:

Preparation and evaluation of COTS to easily integrate and exchange safety COTS. The success will be measured by the use of the conformance checker by the participant before integration of COTS internally and before exchange of COTS between participants.

By using the same conformance methodology, industry will permit to be confident about the good implementation of the safety models.

Success will be also measure by proving the link between AUTOSAR conformance test and safety conformance test on a use case.

METHOD ACTIVITY:

1. Developing conformance check guideline to argument on the aim of the conformance and his perimeter. This guideline will also provide the interaction on the items of the other items of the project (methodology, meta-model, platform…). An analysis on AUTOSAR conformance test will permit to know what will be reuse.

2. Providing of a methodology of conformance check according to the guideline. This will permit to provide the step for the conformance test case creation (analysis, design, implementation validation…) and the methodology of test case execution.

3. Implementing conformance test engine

4. Integration of platform (IHM, plugin…)

5. Working on one or several use-case to validate the plugin and the methodology

6. Providing guideline to implement conformance test.

REFERENCE META-MODEL IMPACT

Meta-model will be enriched with information which will permit to evaluate the conformity of the COTS. This information will be for example dependence between COTS, extra information about the COTS.

LINK TO RTP RESULTS

- Integration & improvement of Meta-model parser into RTP
- Integration & customization of test language compiler (TTCN)
- Development and integration of Consistency checker

SKILLS REQUIREMENT:

- AUTOSAR Conformance test
- Conformance methodology
- Meta model development and parsing
- TTCN-3 language and architecture

Requirements generated for SAFE

The scenario does not lead to requirements for SAFE, due to:

1. TTCN-3 conformance test are no longer of relevance in AUTOSAR. The approach is a dead end.
2. Topic has been deleted from the FPP of SAFE.

5.2.3 Method 02: AltaRica

Method ID: M02

CONTACT PERSON:

Alain Griffault

METHOD NAME:

AltaRica

METHOD RELEVANCE:

AltaRica is a formal language designed to model both functional and dysfunctional behaviors of critical systems. AltaRica can be used to describe safety aspects of systems as well as system architectures at a very high abstract level. All AltaRica model can be checked to prove all kinds of properties related to behaviors.

AltaRica deals with all activities of WP3.

LINK TO RTP:

ARC is a model-checker developed by the LaBRI. ARC will be integrated as a plug-in in the platform. ARC is distributed under a BSD like license.

METHOD JUSTIFICATION:

AltaRica has been designed, ten years ago, both by academics and industrials. Academics to guarantee the feasibility (AltaRica is a formal language with a non-ambiguous semantic) and industrials to guarantee the applicability on real systems.

AltaRica is today the core of two commercial platforms dedicated to safety design: SIMFIA from EADS-APSYS and SD9 Safety Designer from Dassault Aviation.

AltaRica has been used with success in ESACS and ISAAC European projects dedicated to safety activities on aeronautical systems.

METHOD ACTIVITY:

AltaRica is a hierarchical language to describe abstractions of the system's behaviors. A system is represented by a hierarchical configuration and by all possible events that may occur. A particularity is the unification of functional and dysfunctional in the same concept of events. Due to the model-checking approach for validation and verification of the system (and so to the well-known problem of combinatory explosion of the number of states), it is better to use AltaRica during the first phases of the design of a system.

AltaRica can be used in a top-down approach starting from a blank page to obtain by iterative refinement the whole system; but also in a bottom-up approach, starting from components taken in a library and by applying architecture or safety patterns to build the whole system. Once the model is described, you can simulate it, check logical properties, compute min-cuts, generate fault trees, or translate it in another formalism like lustre.

REFERENCE META-MODEL IMPACT

Describe briefly what would be the necessary information require in the reference meta-model from Task 3.5 required to support this method

To make a system description with AltaRica, you need to describe a hierarchy (functional or architecture one) and its communication, and for each sub system of this hierarchy, you need to describe its behaviors as if it is an independent system.

For some analysis, you need to give probabilistic information, and for model-checking, you need to write requirements with a logical point of view.

LINK TO RTP RESULTS

AltaRica can be implemented as a plugin in the RTP platform. To communicate with the other tools, we may have to develop translators between formalisms. For example, translators to and from lustre to AltaRica have been developed during ESACS and ISAAC projects.

SKILLS REQUIREMENT:

To do integration in the RTP platform, we need to understand very well the meta model. To write translators, for each other formalism we need to know its syntax and semantic.

Requirements generated for SAFE

Requirements M02-001 – M02-004 in Requirements collection table. (see appendix)

5.2.4 Method 03: Graphical metric definition and evaluation

Method ID: M03

CONTACT PERSON:

Clemens Reichmann / Eduard Metzker

METHOD NAME:

Graphical metric definition and evaluation

METHOD RELEVANCE:

The activities of this method are related to …

3.3.4 (possible basis for benchmark definition)

3.5 (linking with (meta-)meta-objects)

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 …

4.2.4 (possible basis for benchmark definition)

4.3 (integration)

METHOD JUSTIFICATION:

For benchmarking architectures, a large number of textually or programmatically defined metrics exist. These are often specific to an organization (and part of its IP) or design decision at hand. Possible metrics include cost, weight, bus load, but also qualitative and quantitative measures for safety aspects and constraints.

Main drawbacks with existing approaches are:

- Textual description hides dependencies and requires programming background
- Metrics are dependent on each other
- Metrics are only weakly linked with the model / modeling domain

METHOD ACTIVITY:

Overall benefit of the method is to allow to graphically define metrics and benchmarks, link them to the benchmarked model instance and to calculate metrics. The method will be linked to the SAFE-meta-model, but is in itself meta-model-agnostic ("metric-as-a-model").

Main activities are:

1. Analysis of required metrics and their elements: Existing textual metrics and hardcoded benchmarks and their classes have been collected and systematized to evaluate required artefacts in a metric meta-model. Additional required model entities have to be defined for benchmarking safety attributes (once).
2. Definition of meta-model for metrics: Based on step 1 a data-model being able to hold all aspects needed for metric modelling and execution is defined (once).
3. Graphical definition of metrics: Safety benchmarks can be defined using a dataflow-oriented graphical notation. The notation is usable for automotive domain-experts and allows amongst other features the modelling of sources, sinks, algorithmic/iterative evaluation, control flow as well as extension blocks and embedding of existing textual metrics.
4. Graphical definition of metric relations: Dependencies between atomic

benchmarks can be modelled and analysed.

5. Selection of model-part to benchmark: To apply the metric to the model, the architecture developer can optionally select model parts that are subject to benchmarking.
6. Execution of metrics: The execution engine interprets the metric model and evaluates the defined benchmark. Relevant intermediate results can be logged.
7. Result presentation and evaluation: Results are presented in the model. Results can be quantitative (using numeric/graph presentation) as well as qualitative (with e.g. simple ok/not-ok-signalling).

REFERENCE META-MODEL IMPACT

Meta-class as an anchor for benchmarkable model artifacts, extension mechanism to annotate results

LINK TO RTP RESULTS

Extend the PREEvision framework with the capabilities to perform metrics for safety benchmarking of architectures-

SKILLS REQUIREMENT:

- Meta-model definition
- User interface design
- Metric evaluation
- Software Engineering

Requirements generated for SAFE

Requirements M03-001 – M03-005 in Requirements collection table. (see appendix)

5.2.5 Method 08: Design Space Exploration

Method ID: M08	CONTACT PERSON: Philipp Graf / Martin Hillenbrand
METHOD NAME: Design Space Exploration	
METHOD RELEVANCE: The activities of this method are related to … WT 3.3.3 (Safety and Multi-Criteria Architecture benchmarking)	
LINK TO Reference Tool Platform (RTP): This method can be attached to tool development in WP4 … WT 4.2.6 (Plug-in for safety and multi criteria architecture modeling and benchmarking)	
METHOD JUSTIFICATION: Automated enhancement of models w.r.t. safety. See also WT 3.3.3.	
METHOD ACTIVITY: Given the possibility to include and link safety artifacts with an overall architectural model (including requirements, functional level, software, hardware, electricity and topology), profound manual exploration of design space requires a large amount of human work and can only be performed for important design decision. The generation of alternatives has to take many constraints into account, esp. concerning safety. This approach is also very prone to errors. Thus (semi-)automatic generation and assessment of alternatives based on quantitative safety measures and checking of safety constraints can greatly enhance the speed of finding more optimal solutions in many aspects of the overall design. Main sub-activities include: 1. Definition and selection of suitable exploration strategies: Depending on the variable E/E architectural aspects different adequate exploration strategies have to be selected (e.g. graph optimization algorithms, genetic algorithms). → See statement *Requirements generated for SAFE*. 2. Meta-model instances and their relation to safety aspects: Safety relevant degrees of freedom in the overall model have to be collected and aggregated and set into relationship to the safety model. → Topic covered by several tasks in WP3. 3. Definition and formalization of architectural constraints based on ISO 26262 regulations: Constraints and rules for the degrees of freedom have to be evaluated. These are partly deductible from the relevant regulations, but have to include the possibility for the method user to define constraints. → Aspects of this activity covered by WT 3.3.3 and WT 3.5. 4. Generation of model alternatives: Along the variable axes of freedom model alternatives have to be generated. As design space often is far too large for systematic brute-force evaluation, meaningful strategies are necessary (see activity 1). This also includes random element, either by generating a set of random alternatives (genetic), by following promising heuristics or by a mixed	

strategy (e.g. simulated annealing). → Automatic generation / synthesis of model alternatives and algorithms for design space exploration not in the focus of SAFE.

5. Calculation of safety metrics and other model quality metrics (IP of tool user): To be able to benchmark generated alternatives metrics have to be defined and aggregated to a quality criterion. Again part of this overall metric can be set up from building blocks. However these have to be parameterized and combined by the method user as knowledge in this area largely touches IP-issues. → Activity formulated in WT 3.3.3 of SAFE.
6. Abort criterion: Decisions have to be take up to which point model-exploration is performed. → Strong relation to research on algorithms for model-exploration; not in the focus of SAFE.

REFERENCE META-MODEL IMPACT

Org. text: Overall system meta-model (ideally including for requirements, system, function, software, electronic hardware, harness, electrical distribution system), error and safety model, strong mapping between system aspects on different layers and with safety model.

Interpretation for SAFE: See statement *Requirements generated for SAFE*.

WT 3.3.3 focuses on multi-criteria benchmarking. This technique uses the concepts and relations of the overall system meta-model and the specific modeling aspects derived from this system meta-model. It is not intended nor considered necessary to complicate this meta model by additional concepts to facilitate the benchmarking.

LINK TO RTP RESULTS

SKILLS REQUIREMENT:

- Meta-modeling
- Safety models and regulations

Requirements generated for SAFE

A part of this method use case is not in the focus of SAFE, another is formulated in WT 3.3.3 of safe. The latter does not focus on model-exploration, as described in this method description, but benchmarking. WT 3.3.3 focuses on multi-criteria benchmarking. This technique uses the concepts and relations of the overall system meta-model and the specific modeling aspects derived from this system meta-model. It is not intended nor considered necessary to complicate this meta model by additional concepts to facilitate the benchmarking.

Except WT 3.3.3 itself, no further requirements are generated for SAFE.

5.2.6 Method 09: Model Transformations

Method ID: M09

CONTACT PERSON:

Adaptation to SAFE: Stefan Voget

Origin: lizzi@itemis.de, Itemis

METHOD NAME:

Model Transformations

METHOD RELEVANCE:

The activities of this method are related to …

WT 3.2.1 (System and Software models enhancement)

WT 3.5 (meta-model definition)

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 …

WT 4.1 (meta-model implementation)

WT 4.2 (specialized plug-in realization)

METHOD JUSTIFICATION:

Model transformations will be necessary to bridge the gap between existing tools and technologies. It includes Simulink to EAST-ADL/SysML transformation, as well as retrieving safety information from tools to exploit them in SAFE reference models.

METHOD ACTIVITY:

The purpose of the method is to provide interoperability between existing tools / formats that will be used in SAFE. The central point being the SAFE meta-model, transformations will allow import/export heterogeneous models and hence to exploit the best of breed of the different tools and technologies, in particular in regards with their safety capabilities.

Main activities are:

1. Develop or extend existing meta-models for each model to transform.
2. Implement the transformation. This can be done using several technologies such as QVT, ATL or openArchitectureWare.
3. Create tools or simple workflow to automate the different transformations.

Foreseen transformations include Simulink to EAST-ADL/SysML and/or to SAFE meta-model. Some tools in SAFE will also be used for their safety capabilities including failure and error modeling. Transformations will allow in particular benefiting of the safety elements provided by these tools into the reference meta-model.

REFERENCE META-MODEL IMPACT

The reference meta-model should contain the concepts that need to be imported from other tools.

LINK TO RTP RESULTS

- ATL or openArchitectureWare transformations
- EAST-ADL/SysML and/or EMF based editor

SKILLS REQUIREMENT:

- Meta-modeling
- Transformation knowledge

- Knowledge of the tools to bridge
Requirements generated for SAFE Requirements M09-001 – M09-004 in Requirements collection table. (see appendix)

5.2.7 Method 10: Seamless modeling of System Architecture incl. Dependability with EAST-ADL

Method ID: M10

CONTACT PERSON:

Adaptation to SAFE : S. Voget

METHOD NAME:

Seamless modeling of System Architecture including Dependability with EAST-ADL

METHOD RELEVANCE:

The activities of this method are related to …

WT 3.1.2, WT 3.1.3

WT 3.4

WT 3.5

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 …

WT 4.1

WT 4.2.1, WT4.2.3, WT4.2.5

METHOD JUSTIFICATION:

One main challenge with current development, deployment, maintenance and upgrade of automotive embedded systems is the lack of method and tool support for information traceability and transformation across lifecycle stages, quality aspects, multiple analysis and V&V technologies, product family and organisation boundaries. Traditional approaches with social and text-based information communication and documentation often failed in handling advanced embedded systems or meeting desired engineering efficiency and effectiveness.

While state-of-the-art modelling and analysis technologies provide many useful supports for behaviour and quality assessments, challenges remain in consolidating the related process- and information-flows. In engineering practices, difficulties have been shown in the areas of version and change management, reuse and integration of third-party solutions, traceability of requirements and V&V results, integration of system failure/error views, etc.

Currently, the emerging ISO26262 on Functional Safety for Road vehicles puts new demands on the structuring and traceability of requirements, functional and technical solutions, analysis and V&V results. It is also required that various information about system risks, design and realization solutions should be referenced and aggregated according to the Safety Case method in order to provide qualitative argumentations about why a system is safe enough.

METHOD ACTIVITY:

The technique provides supports for architecture modeling and traceability based on EAST-ADL, involving the following main activities.

- Description of system requirements, their refinement hierarchy, and the relations to design solutions, V&V cases and results.
- Description of system environment, system design, and variability at multiple

levels of abstraction, and system implementation based on AUTOSAR.

- Description of the malfunctions and scenarios of items, system hazards, safety goals, functional and technical safety concepts, and Safety Case in compliance with ISO26262.
- Description of an error view characterizing the component errors and propagations at different levels of abstraction for safety analysis.
- Description of nominal and erroneous behaviors with one or multiple formalisms.

REFERENCE META-MODEL IMPACT

(The EAST-ADL methodology and domain-model will constitute a useful basis for the SAFE tool-chain definition and meta-model where extensions in regards to hardware architecture, AUTOSAR platform, behavior formalisms (e.g., Altarica), analysis plug-ins will be supported.)

LINK TO RTP RESULTS

- An Eclipse Ecore realization of the EAST-ADL meta-model
- Plug-ins supporting for error and dependability modeling, safety analysis, design space exploitation and traceability.

SKILLS REQUIREMENT:

- Systems engineering and architecture development.
- EAST-ADL

Requirements generated for SAFE

Requirements M10-001 – M10-002 in Requirements collection table. (see appendix)

5.2.8 Method 12: Contract based evaluation of safety functions

Method ID: M12

CONTACT PERSON:

Thomas Peikenkamp/ Marion Suerken

METHOD NAME:

Contract based evaluation of safety functions

METHOD RELEVANCE:

The activities of this method are related to …

WT 3.2.2 (H/W failure descriptions)

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 …

WT 4.2.3 (modeling fault injection)

METHOD JUSTIFICATION:

Quantitative breakdown for safety functions.

METHOD ACTIVITY:

1. Developing a failure propagation model that allows the composition of quantified and un-quantified failures,
2. Providing a description of (the expected) failure behavior of architectural components based on requirements (contracts) formalized in the above model, and
3. Providing analysis methods allowing to quantify achievement of safety targets and to compare the safety characteristics of different architectures based on the above description of the failure behavior of its components.

The model developed under 1. is intended to cater for probabilistic failure data as those available for many H/W devices as well as for situations, where such data is not available, e.g. software errors or failures, where the effect can be mitigated by corresponding driver actions (controllability). Based on this model quantitative safety requirements can be described that allow, for instance, to require that a subsystem has a fault containment property under the *assumption* that the driver corrects all detected failures.

The main vehicle used for *localizing* quantified safety requirements are contracts consisting of assumptions and promises. The structure of these requirements allows to quantify, for instance, the correct execution of function provided by a component (promise) separately from the circumstances (assumption), under that the component is expected to provide this function. Based on the capabilities of the model developed under 1., the methods used for quantifying the achievement of safety targets are able to exploit characteristics of the environment or the controllability of failures to improve the result of the analysis.

The development of the method will be compliant to CESAR, in particular a development conformant to Reference Technology Platform is foreseen. For the description of the architecture and possible configurations, an integration in the CASE tool PREEvision tool from Vector Informatik / Aquintos is foreseen.

REFERENCE META-MODEL IMPACT

Representation of quantitative failures and dependencies. The definition of the corresponding figures should be allowed as well as the representation of the analysis results. Furthermore, the representation of architectural differences will be required.

LINK TO RTP RESULTS

- Adaptation of SPEEDS “dominance checking” to compare architectures w.r.t. their ability to meet given safety characteristics
- Stochastic model checking engines

SKILLS REQUIREMENT:

- Failure modeling
- Probabilistic models for failure dependencies
- Assume guarantee reasoning
- Symbolic requirement analysis methods

Requirements generated for SAFE

Requirements M12-001 – M12-007 in Requirements collection table. (see appendix)

5.2.9 Method 13: Uniform modeling of Variability in Automotive System Architectures

Method ID: M13	CONTACT PERSON: Michael Schulze

METHOD NAME:

Uniform modeling of Variability in Automotive System Architectures

METHOD RELEVANCE:

The activities of this method are related to …

WT 3.4 (Variant Management)

WT 3.2.1 (System and Software models enhancement)

WT 3.5 (Meta Model Definition)

LINK TO Reference Tool Platform (RTP):

This method can be attached to tool development in WP4 …

WT 4.2

WT 4.3

METHOD JUSTIFICATION:

Tools being used to develop automotive systems do not or do provide only limited support for variability modelling and variant management. For instance Simulink does not provide sufficient support here, only with help of 3rd party tools this is possible. Other tools provide their own variant modelling including tool specific semantics. Modelling standards such as SysML&UML also do not provide variability related modelling concepts out of the box. In the automotive domain have been initial efforts to work in this issue, EAST ADL and also upcoming release of AUTOSAR provide some support for variability modelling and variant management.

Part of the problem is the diversity of concepts for the different sub domains of automotive development (HW development, SW Development, Functional Modelling, Architecture Modelling). While all of these are strongly related, there is no uniform way to express variability related information (especially constraints between the domains).

The majority of these approaches currently provide only basic configuration validation (EAST-ADL) or even come without semantic validation (AUTOSAR).

Necessary for dealing with complex domains is not only proper validation of configured instance but also means to efficiently explore the variation space, restrict it variants /variant sets of interest. The exploration has to be supported by efficient reasoning to provide sufficient level of interaction with the tool users. Even advanced tools like pure::variants which provide good reasoning support do not cover the exploration of large variation spaces as targeted in SAFE.

Existing variability modelling concepts are not well suited to the needs of safety critical systems since they usually to not take into account the V&V including a safety related certification process. Basically each variant needs independent V&V but due to the high effort for this activity (compared to other development activities) limits the usefulness/effectiveness of reuse approaches.

METHOD ACTIVITY:

The method consists of the following main activities:

1. Express variability (variation points and constraints) in the different models used

to describe architectural and safety related concerns. This includes models as well as related documents in textual form including code & scripts. This shall be supported by easy to use UI integrated in the respective development environment. Addressed tools are the tool set defined to be part of the SAFE tool chain.

2. Based on provide variable models the definition of variants or variant spaces of interest is done. Selection of variants is based on requirements plus results from exploration of design space (exploration methods and tools are provided by other methods e.g. from FZI/EDALab/MTG management of variants and composition of systems from independent variant blocks. This will be repetitive process including more and more domain knowledge to finally lead to fully specified systems.
3. Changes to domain knowledge are made and previously defined variant descriptions are used to analyse change impact based on the changed domain knowledge input. For system variants not impacted, redo of V&V can be skipped, since change did not impact this particular set of instances.

REFERENCE META-MODEL IMPACT

The meta model must provide support efficient modeling of variability related information across all domain specific models by providing means to describe implicit and explicit *variation points* including related *constraints* describing the variation space. The meta model must be able to express variable systems (or parts of variable systems), information about partial and full configured instance of those systems (i.e. *variant descriptions* and *variant instances*). Partially instantiated means multiple variants can be derived but some of the variability space has been reduced by configuration. Fully instantiated variant descriptions are instance with no open variation left.

LINK TO RTP RESULTS

- Development of GUI for variability modeling in the relevant tools of SAFE tool chain
- Implementation of API to couple domain specific tool with variant management tool suites including a reference implementation using pure::variants framework plus SAFE specific components for domain specific variability related reasoning developed as part WP3 / WP 4.2 (e.g. FZI, MTG)
- Enhance variability reasoning w.r.t to SAFE complexity requirements with heuristic methods

SKILLS REQUIREMENT:

- Meta-model definition
- User interface design
- Reasoning
- Software Product Line Engineering

Requirements generated for SAFE

Requirements M13-001 – M12-005 in Requirements collection table. (see appendix)

6 Coverage of Use Cases and Methods in WP5

In this section it is analyzed, which use cases are covered in the validation scenarios in WP 5.

WT2.3 Scenario	WT 5.x Evaluation Task	WT5.x Task Title / Remarks
S01	WT 5.6	Model based analysis and code generation
S02	WT 5.2	Engine Management System
S05	WT 5.2	Engine Management System
S10	WT 5.5	Early validation of technical safety concept
S12	WT 5.6	Model based analysis and code generation. S12 is partially evaluated, only Code Generation part.
S17	WT 5.2	Electrical Brake System
S18	WT 5.3	Evaluation of mixed criticality SW layer
S20	WT 5.4	Development of an MCU model

7 Requirements template

In this section, the template to collect and allocate the SAFE requirements documents is described. The Sheet is very similar to the template used in D2.1. Deviations are specified here:

7.1 The Excel Sheet

Column	Description
A	Title: Use Case Scenario
B	Title of the Use Case or Method In case of product development the discipline is mentioned here
C	Project internal ID of requirement ID numeration: REQ [Use-Case ID]_[Sub Nr.]. For split requirements add alphanumerical info and WT info REQ [ISO Part]_[Sub Nr.]_[a-z]_[WT#] Examples: REQ M09_003; REQ S12_007
D - AO	(as specified in D2.1)

8 Conclusions and Discussion

In the first phase of the SAFE project the work in WT2.3 helped to create Requirements based on industrial experience (Use cases and Methods).

Scenario Identification	Scenario Title	Number of Requirements
S01	Model based analysis and code generation for safety aspects in safety relevant systems	**3**
S02	Safety assessment of engine management system based on models	**6**
S05	Preliminary Hazard Analysis and safety requirements definition	**2**
S06	Variant Management Function	**7**
S07	Optimization of Model Based Design with safety handling including re-use	**2**
S10	Integrated model based safety	**9**
S11a	Hazard and Risk Analysis	**4**
S11b	Generation of Safety Concepts	**8**
S11c	Safety Collaboration	**4**
S12	Connect safety analysis with a model-based development process, including requirements management and code generation.	**9**
S17	Functional and Technical Safety Concept including analysis and verification according ISO 26262 of a integrated brake system	**6**
S18	Integration of safety-related and none safety-related software	**2**
S19-1	Safety case contents	**4**
S19-2	Variability-aware Safety Case	**2**
S19-3	Safety Case in distributed development (OEM / Tier-1)	**3**
S19-4	Safety Case notation	**1**
S19-5	Model-based safety engineering and integration across system abstraction levels	**3**
S19-6	Model-based and compoundable Safety Concepts	**3**
S19-7	Combined Safety Analysis in one Safety Model	**1**
S19-8	Model-based Safety Patterns	**1**
S19-9	Modular Hazard Analysis on model-based function or system (item definition)	**8**
S19-10	Synchronization between Hazard and Risk Analysis (H+R) and Item Definition (System Definition)	**1**

S19-11	Consistency checks between Modular Hazard and Risk Analysis (H+R) and model-based Item Definition	**1**
S19-12	Comparability of Hazard and Risk Analysis (H+R)	**1**
S19-13	Guided Hazard and Risk Analysis (H+R)	**1**
S19-14	Safety Case properties	**1**
S19-15	Supported Analysis on Safety Case Contents	**2**
S19-16	Safety Case analysis due to context modifications (Change Impact Analysis)	**3**
S19-17	Safety Case incremental compilation/development	**2**
S19-18	Safety Case incremental assessment	**1**
S19-19	Safety Model Interoperability with various Modeling Tools (XML)	**6**
S19-20	Safety Model Abstraction Levels	**2**
S20	Verification of software behavior and the effectiveness of implemented safety measures in the presence of faults injected into the microcontroller hardware	**2**
	Total	**111**

Scenario Identification	**Method Name**	**Number of Requirements**
M01	Conformance check	**0**
M02	AltaRica	**4**
M03	Graphical metric definition and evaluation.	**5**
M08	Design Space Exploration	**0**
M09	Model Transformations	**4**
M10	Integrated System Architecture and Dependability Modeling with EAST-ADL2	**2**
M12	Contract based evaluation of safety functions	**7**
M13	Uniform Modeling of Variability in Automotive System Architectures	**5**
		27

So, all in all 138 requirements have been elicited based on the industrial use cases and methods. These Requirements have been classified (included/excluded) and (partially) allocated to the technical work packages WP3, 4, and 6.

9 References

[1] SAFE FPP

10 Acknowledgments

This document is based on the SAFE project in the framework of the ITEA2, EUREKA cluster program Σ! 3674. The work has been funded by the German Ministry for Education and Research (BMBF) under the funding ID 01IS11019, and by the French Ministry of the Economy and Finance (DGCIS). The responsibility for the content rests with the authors.