Contract number: ITEA2 – 10039

Contract number: Eurostars 6095 Safe-E

Safe Automotive soFtware architEcture (SAFE) & Safe Automotive soFtware architEcture – Extension (SAFE-E)

WP3.2.1 System and software models enhancement

Deliverable D3.2.1.d: 2nd Update proposal for extension of meta model for software and system modelling

Due date of deliverable: 30/11/2014

Actual submission date: 28/11/2014

Organization name of lead contractor for this deliverable: AVL

Editor: Elvira Biendl

Contributors: WT3.2.1 Participants

Reviewer: Stefan Voget, Jörg Kemmerich, Dirk Geyer, Alexander Rudolph, Maged Khalil

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

Revision chart and history log

Version	Date	Reason
0.1	24.03.2014	1st Draft based on D3.2.1.c
0.2	25.11.2014	Update of Figures Update of referenced documents
0.3	25.11.2014	Editorial changes
1.0	28.11.2014	Release of D3.2.1.d

1 Table of contents

The Eurostars Programme is powered by EUREKA and the European Community

EUREKA

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

FFG

The Eurostars Programme is powered by
EUREKA and the European Community

EUREKA

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

FFG

2 List of figures

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

FFG

3 Executive Summary

The automotive industry uses more and more electronically controlled equipment in passenger cars that covers safety critical functionality. This leads to an increase of systematic failures and random hardware failures. Many of those failures are able to cause harm to people. These safety relevant failures shall be reduced to a level of unreasonable risk.

ISO 26262 contains a guidance to avoid or mitigate the risks caused by safety relevant failures by providing appropriate requirements and processes.

Currently the automotive industry is applying the requirements and processes specified in the ISO 26262 to provide new systems that are able to avoid the increasing risks or at least mitigate them to an appropriate level.

The objective of this document is to analyze existing models like EAST ADL, SysML or AUTOSAR with the requirements given in the ISO 26262 part 4 and part 6. The result of this analysis shall provide input for creation of the SAFE meta-model that can be used to describe safety relevant systems in scope of ISO 26262.

The solution that is described in this document is the update of already provided document D3.2.1.c [9] and shall be used as a starting point for discussion with other users of EAST ADL, AUTOSAR and ISO 26262 to find an effective solution that is easy to use in future development projects.

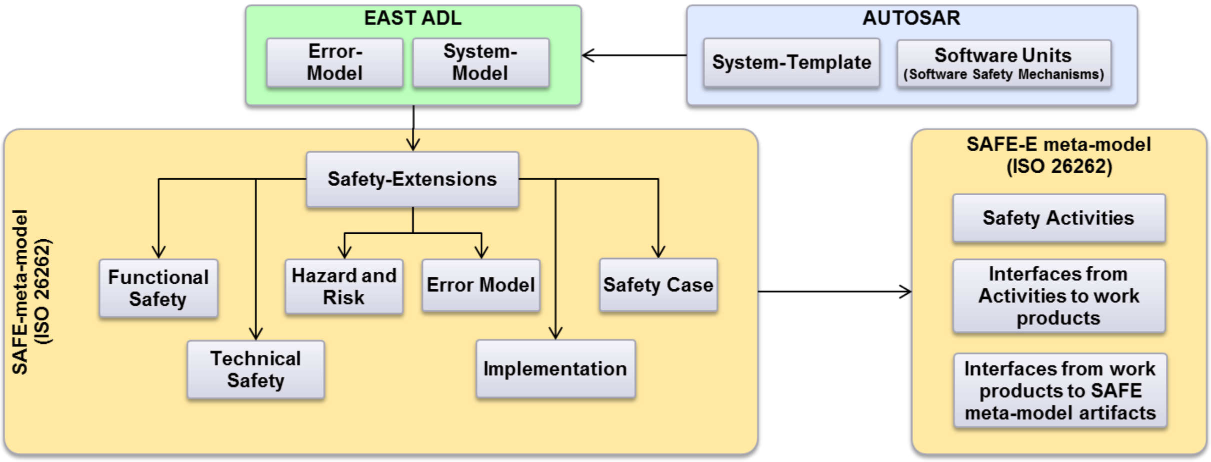

EAST ADL
Error-Model
System-Model
AUTOSAR
System-Template
Software Units (Software Safety Mechanisms)
SAFE-meta-model (ISO 26262)
Safety-Extensions
Functional Safety
Hazard and Risk
Error Model
Safety Case
Technical Safety
Implementation
SAFE-E meta-model (ISO 26262)
Safety Activities
Interfaces from Activities to work products
Interfaces from work products to SAFE meta-model artifacts

Figure 1: overview meta-models

4 Introduction

Up to now the automotive industry is already doing systematic failure analysis. But now the ISO 26262 defines the need to avoid unreasonable risk. Therefore this kind of analysis is getting more important for future automotive development projects.

The increasing use of electronically controlled equipment in the car leads to a changed behavior of the driver. Actions of the driver are guided by electronically controlled features, e.g. adaptive cruise control, electronic stability control, etc. All these features are able to help the driver to handle critical traffic situations. In a time of increasing number of cars on the road and increasing diversion for the driver during driving on the road, the driver trusts more and more in the new features of the car. All these topics lead to a changing of the common level of reasonable risk.

Based on the fact that unreasonable risk depends on a certain context according to valid societal moral concepts the automotive industry recognizes the challenge to handle the environmental context during development. The actual level of unreasonable risk in the target market of the vehicle in development is a new topic that shall be established in the already existing development process landscape.

Therefore this document describes the safety analyses as a central topic of the development of safety relevant products in scope of ISO 26262 to identify the safety relevant failures that are able to cause the hazardous events and to provide evidence of the effectiveness of the implemented safety measures in the item.

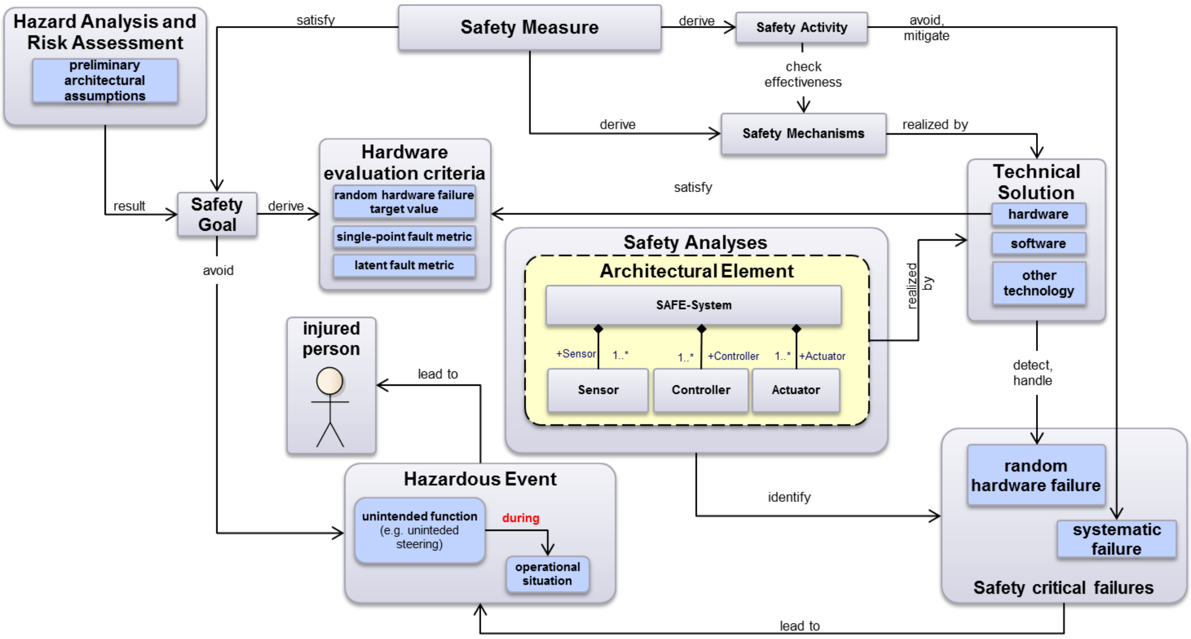


Figure 2: Safety Analyses as central topic during development of safety relevant items in scope of ISO 26262

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

4.1 Abbreviation, Special Terms, Acronyms

The following table describes the special terms used in this document.

Abbreviation/ Acronyms	Description
ASIL	**A**utomotive **S**afety **I**ntegrity **L**evel
AUTOSAR	**Aut**omotive **O**pen **S**ystem **Ar**chitecture
ASW	**A**pplication **S**oft**W**are
BSW	**B**asic **S**oft**W**are
Component	A **component** is an element of system that contains a single functionality (e.g. steering, break, powertrain, chassis ...). The component can consist of hardware elements, software elements, systems, sensors, actuators ... Therefore the component contains all elements to fulfill the specified function.
EAST-ADL	**E**lectronics **A**rchitecture and **S**oftware **T**echnology - **A**rchitecture **D**escription **L**anguage
Element	**Element** is a term that is used on each architectural level in a different way. At system level (e.g. system = vehicle) a system element is one part of the vehicle (e.g. wheel, window, mirror ...) At component level (e.g. component = powertrain) the element is one part of the powertrain (e.g. transmission. At part level (e.g. part = µC) the element is one part of the µC (e.g. a pin)
FAA	**F**unction **A**nalysis **A**rchitecture
FDA	**F**unction **D**esign **A**rchitecture
Hazard	A **hazard** is a potential source of physical injury or damage to the health of persons caused by malfunctioning behavior of the item
Hazardous Event	A hazardous event is a combination of a hazard and an operational situation.
Operational situation	An **operational situation** is a scenario that can occur during a vehicle's life.
preliminary	**Preliminary** is used to classify the maturity of an element. It means that the element is not finally verified or validated.
RTE	**R**eal **T**ime **E**nvironment
safety relevant failure	**Safety relevant failures** are failures that are identified during safety analyses to have the potential to lead to a violation of a safety goal

4.2 Scope of the document

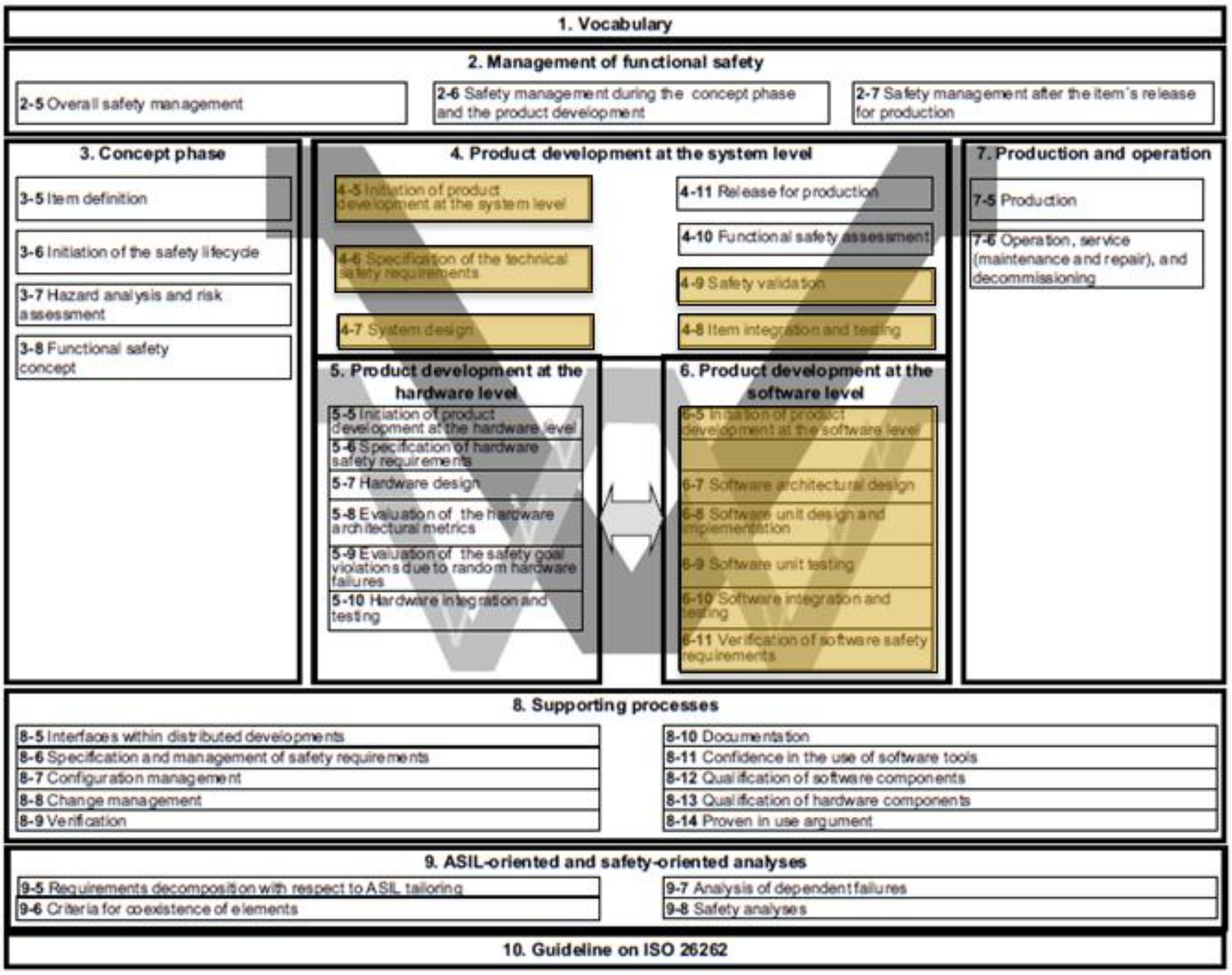

1. Vocabulary
2. Management of functional safety
2-5 Overall safety management
2-6 Safety management during the concept phase and the product development
2-7 Safety management after the item's release for production
3. Concept phase
3-5 Item definition
3-6 Initiation of the safety lifecycle
3-7 Hazard analysis and risk assessment
3-8 Functional safety concept
4. Product development at the system level
4-5 Initiation of product development at the system level
4-6 Specification of the technical safety requirements
4-7 System design
4-11 Release for production
4-10 Functional safety assessment
4-9 Safety validation
4-8 Item integration and testing
5. Product development at the hardware level
5-5 Initiation of product development at the hardware level
5-6 Specification of hardware safety requirements
5-7 Hardware design
5-8 Evaluation of the hardware architectural metrics
5-9 Evaluation of the safety goal violations due to random hardware failures
5-10 Hardware integration and testing
6. Product development at the software level
6-5 Initiation of product development at the software level
6-7 Software architectural design
6-8 Software unit design and implementation
6-9 Software unit testing
6-10 Software integration and testing
6-11 Verification of software safety requirements
7. Production and operation
7-5 Production
7-6 Operation, service (maintenance and repair), and decommissioning
8. Supporting processes
8-5 Interfaces within distributed developments
8-6 Specification and management of safety requirements
8-7 Configuration management
8-8 Change management
8-9 Verification
8-10 Documentation
8-11 Confidence in the use of software tools
8-12 Qualification of software components
8-13 Qualification of hardware components
8-14 Proven in use argument
9. ASIL-oriented and safety-oriented analyses
9-5 Requirements decomposition with respect to ASIL tailoring
9-6 Criteria for coexistence of elements
9-7 Analysis of dependent failures
9-8 Safety analyses
10. Guideline on ISO 26262

Figure 3: Scope of the document

This document is created based on the requirements given in ISO 26262 part 4 and part 6. The allocation of the requirements covered in this document is given in the referenced deliverables D2.1.c [5]. The scope of this document is to model a safety relevant item according to ISO 26262 by using already existing models like EAST-ADL and AUTOSAR.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

4.3 Architectural Overview

The following figure is showing the artifacts that are needed to model safety relevant items in addition to the already existing artifacts given in the referenced EAST-ADL-model.

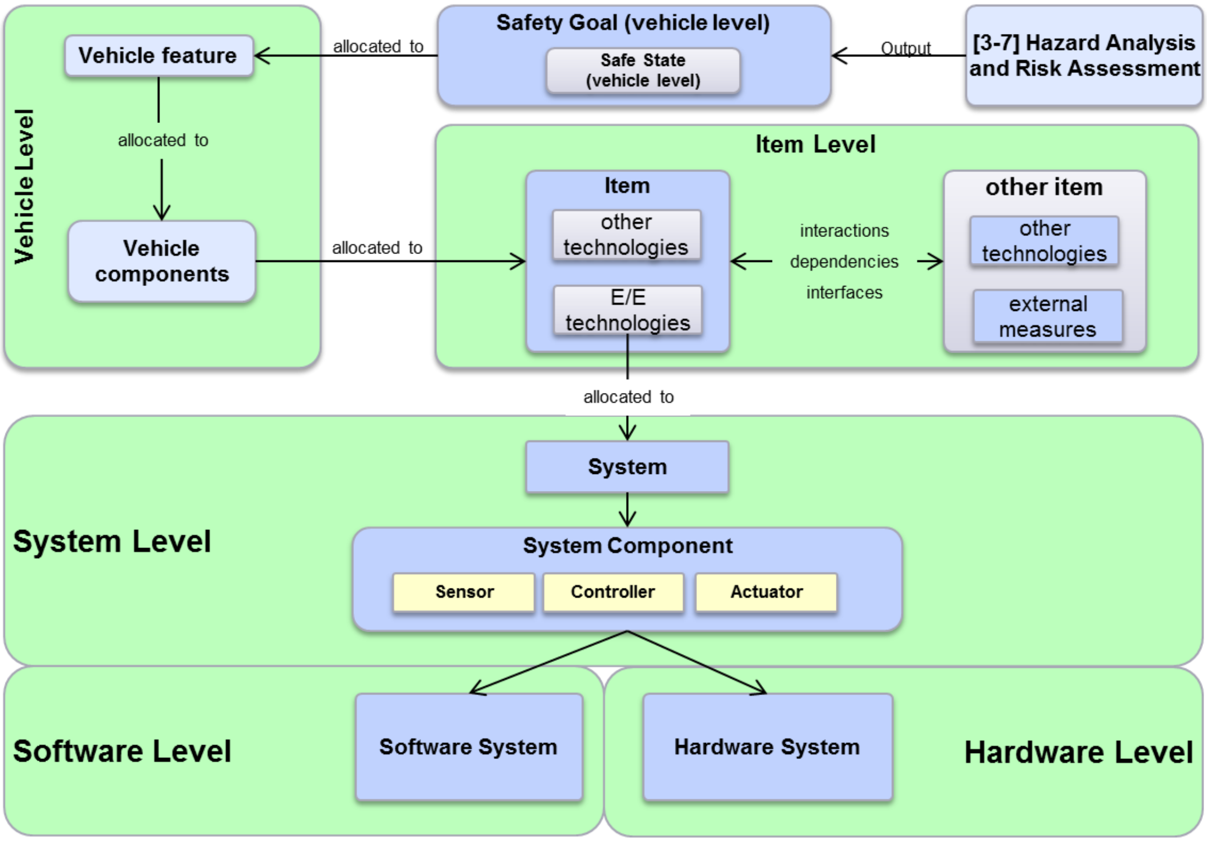


Figure 4: Item Architecture overview

The ISO 26262 is defined for safety-related systems that include E/E-systems that are installed in a series production passenger car with a maximum gross vehicle mass up to 3500 kg. Therefore the item is defined as a sub-system of a vehicle. The following architectural levels shall be specified for an item in scope of ISO 26262:

- Vehicle Level:

  The vehicle level is defined as the top level of the architecture. It describes the context of the item as well as the architectural splitting up to different items.

- Item Level:

  The item level describes the functionality of the item as well as the architectural splitting up to different systems.

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

- System Level:

  The system level describes the architectural elements of the system. A system contains at least one sensor, one controller and one actuator. The architectural splitting up of each sensor, controller, actuator to components is also part of this level. Another part of this level is the allocation of the different elements to software and hardware components. The architectural description of the interfaces between the Components is also part of this level.

- Software Level:

  The software level contains the architectural splitting up of the software system to software partitions, software component and software units. The architectural description of the interfaces between the Software Units is also part of this level.

- Hardware Level:

  The hardware level contains the architectural splitting up of the hardware system to hardware component and hardware parts. The architectural description of the interfaces between the Hardware Parts is also part of this level.

The SAFE meta-model shall provide the follwing safety extensions and packages as add-on for the EAST-ADL model.

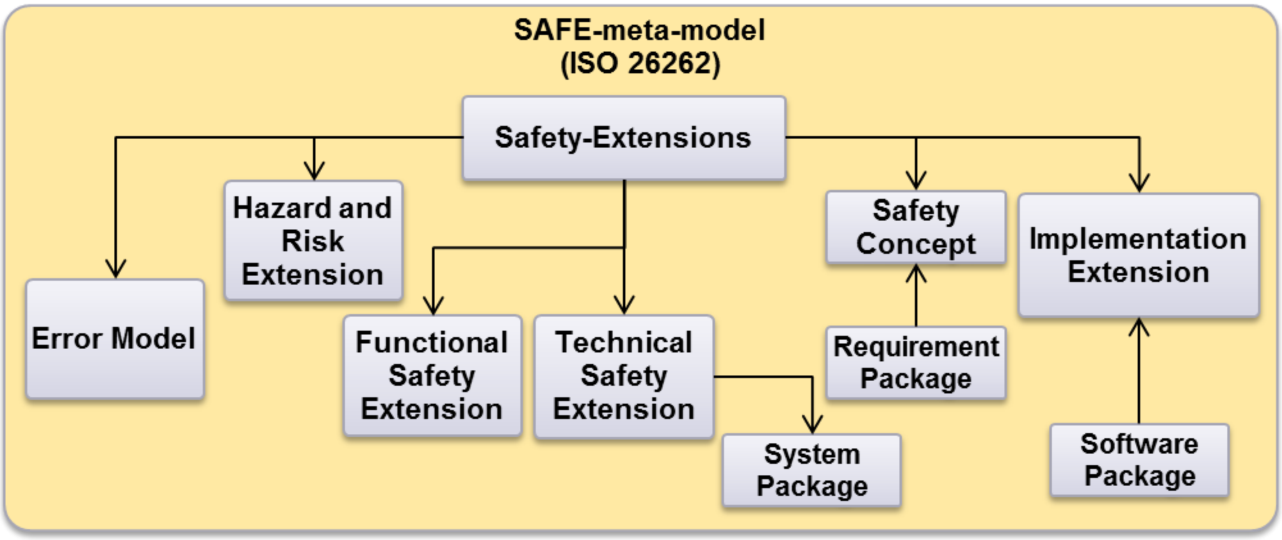


Figure 5: Safety extensions specified for the SAFE meta-model

4.3.1 Hazard and Risk Safety Extension

Hazard and Risk Safety Extension shall contain all artifacts to model the relevant item information to derive the hazardous events. Hazardous event is defined as a combination of a hazard with an operational situation. The item features shall be modeled by the feature model that is part of the vehicle level defined in EAST-ADL.

Safety Mechanisms that are already known during execution of Hazard Analysis and Risk Assessment shall not be regarded during classification of the hazardous event. They shall be provided as input to the functional safety concept.

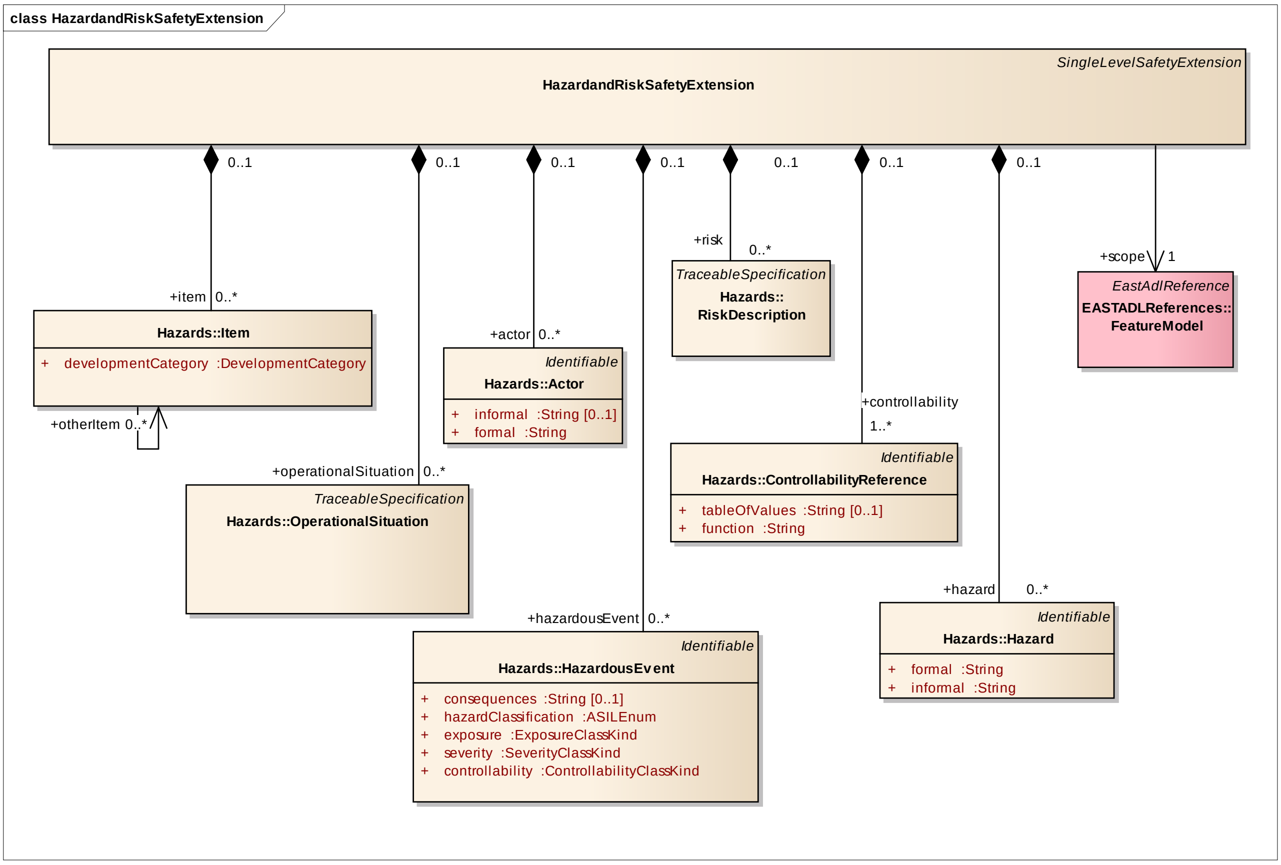

class HazardandRiskSafetyExtension
SingleLevelSafetyExtension
HazardandRiskSafetyExtension
0..1
0..1
0..1
0..1
0..1
0..1
0..1
+item 0..*
Hazards::Item
+ developmentCategory :DevelopmentCategory
+otherItem 0..*
+operationalSituation 0..*
TraceableSpecification
Hazards::OperationalSituation
+actor 0..*
Identifiable
Hazards::Actor
+ informal :String [0..1]
+ formal :String
+hazardousEvent 0..*
Identifiable
Hazards::HazardousEvent
+ consequences :String [0..1]
+ hazardClassification :ASILEnum
+ exposure :ExposureClassKind
+ severity :SeverityClassKind
+ controllability :ControllabilityClassKind
+risk 0..*
TraceableSpecification
Hazards::
RiskDescription
+controllability
1..*
Identifiable
Hazards::ControllabilityReference
+ tableOfValues :String [0..1]
+ function :String
+hazard 0..*
Identifiable
Hazards::Hazard
+ formal :String
+ informal :String
+scope 1
EastAdlReference
EASTADLReferences::
FeatureModel

Figure 6 SAFE meta-model - Hazard and Risk Safety Extension

Further details according to Hazard Analysis and Risk Assessment see D3.1.1.c [6]

4.3.2 Functional Safety Extension

Based on the Functional Analysis Architecture (FAA) that is part of the Analysis Level defined in EAST-ADL, the Functional Safety Extension shall be used to specify specifies the add-on needed to model the functional safety concept defined in the ISO 26262 part 3 chapter 8

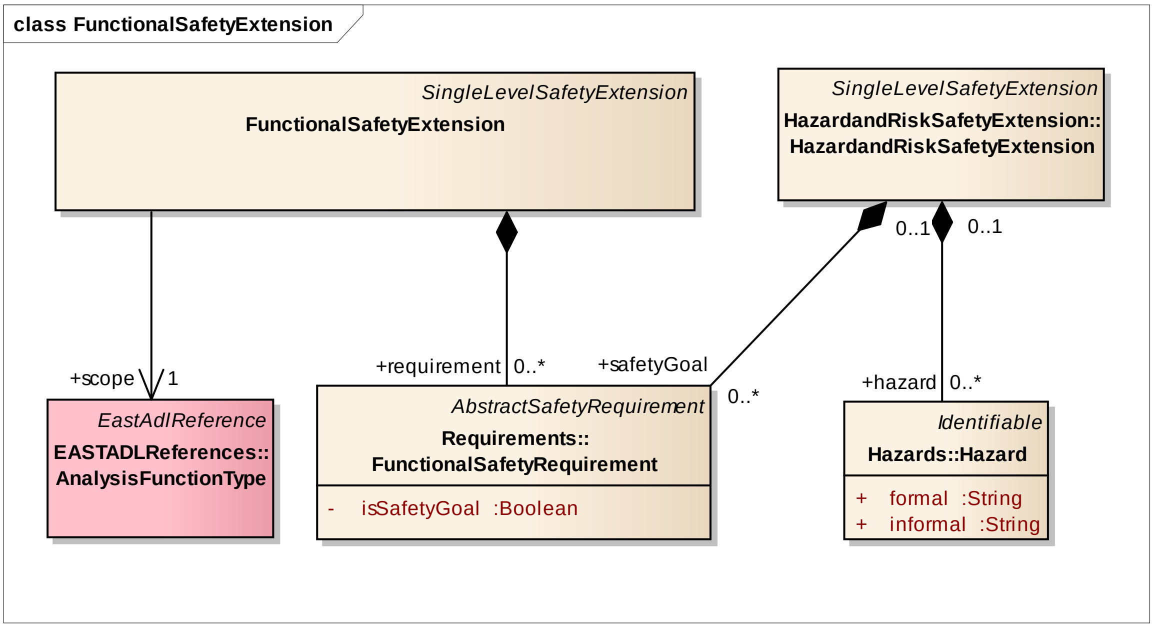

class FunctionalSafetyExtension
SingleLevelSafetyExtension
FunctionalSafetyExtension
SingleLevelSafetyExtension
HazardandRiskSafetyExtension::
HazardandRiskSafetyExtension
0..1
0..1
+scope
1
+requirement
0..*
+safetyGoal
0..*
+hazard
0..*
EastAdlReference
EASTADLReferences::
AnalysisFunctionType
AbstractSafetyRequirement
Requirements::
FunctionalSafetyRequirement
- isSafetyGoal :Boolean
Identifiable
Hazards::Hazard
+ formal :String
+ informal :String

Figure 7: Functional Safety Extension

Further Details according to the functional safety concept see chapter 5.1.3

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

4.3.3 Technical Safety Extension

The Technical Safety Extension is used as interface to the Design Level defined in EAST-ADL. This extension specifies the add-on needed to model a specific technical solution that is derived based on the functional safety concept. It contains the

- technical safety concept (ISO 26262 part 4 chapter 7)
- hardware software interface specification (ISO 26262 part 4 chapter 7.4.6)

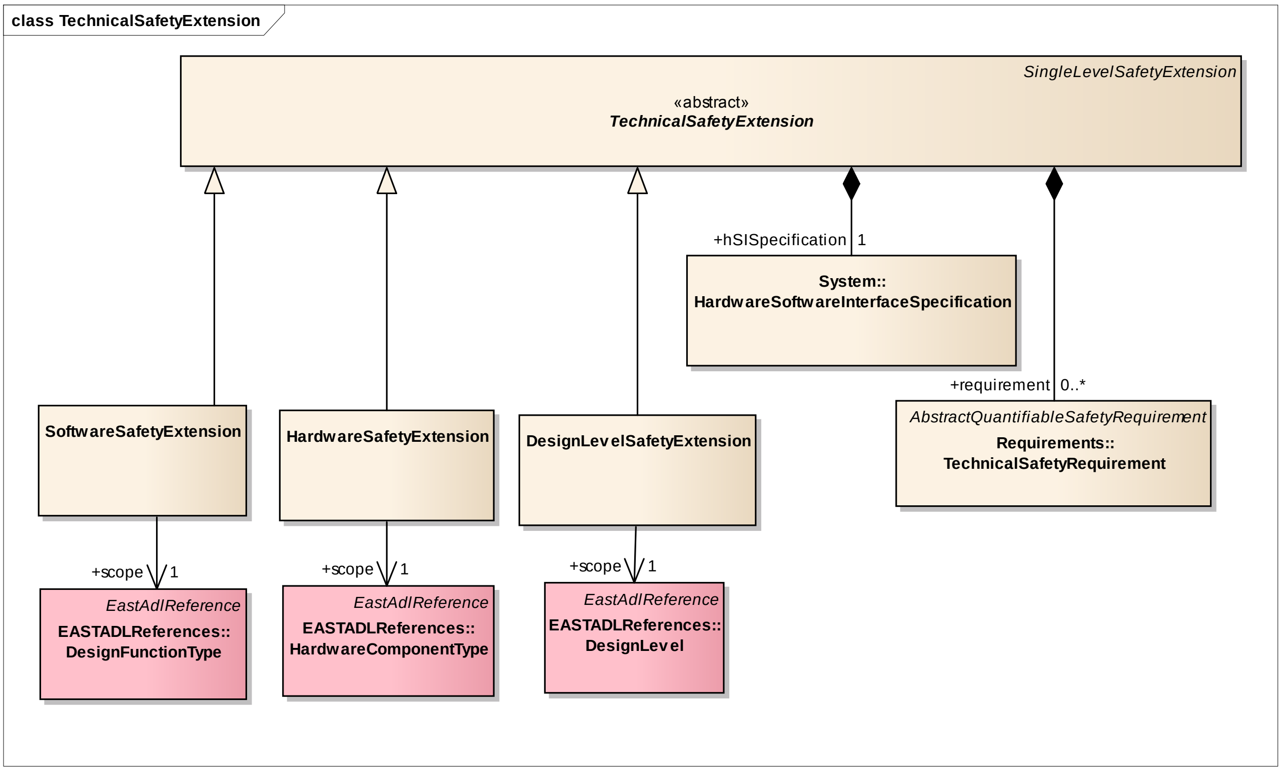


Figure 8: Technical Safety Extension

Further details according to the safety relevant content of the technical safety extension see chapter 5.4.5.3 and chapter 5.4.2

4.3.4 Requirements Package

The requirements package is defined as one part of the SAFE meta-model.

Safety Requirements shall be categorized into different groups:

- Functional Safety Requirement
- Technical Safety Requirement
- Software Safety Requirements
- Hardware Safety Requirements

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

Each safety requirement contains a sub-category that specifies the use case of the requirement:

- ***Quantitative***
  safety requirement describe for example the hardware architectural metrics
- ***Process***
  safety requirement describe for example safety relevant verification methods
- ***Product***
  safety requirements describe the technical solution specified to fulfill the safety goals
- ***Constraint***
  describe for example architectural assumptions or design constrains given from the higher level architecture

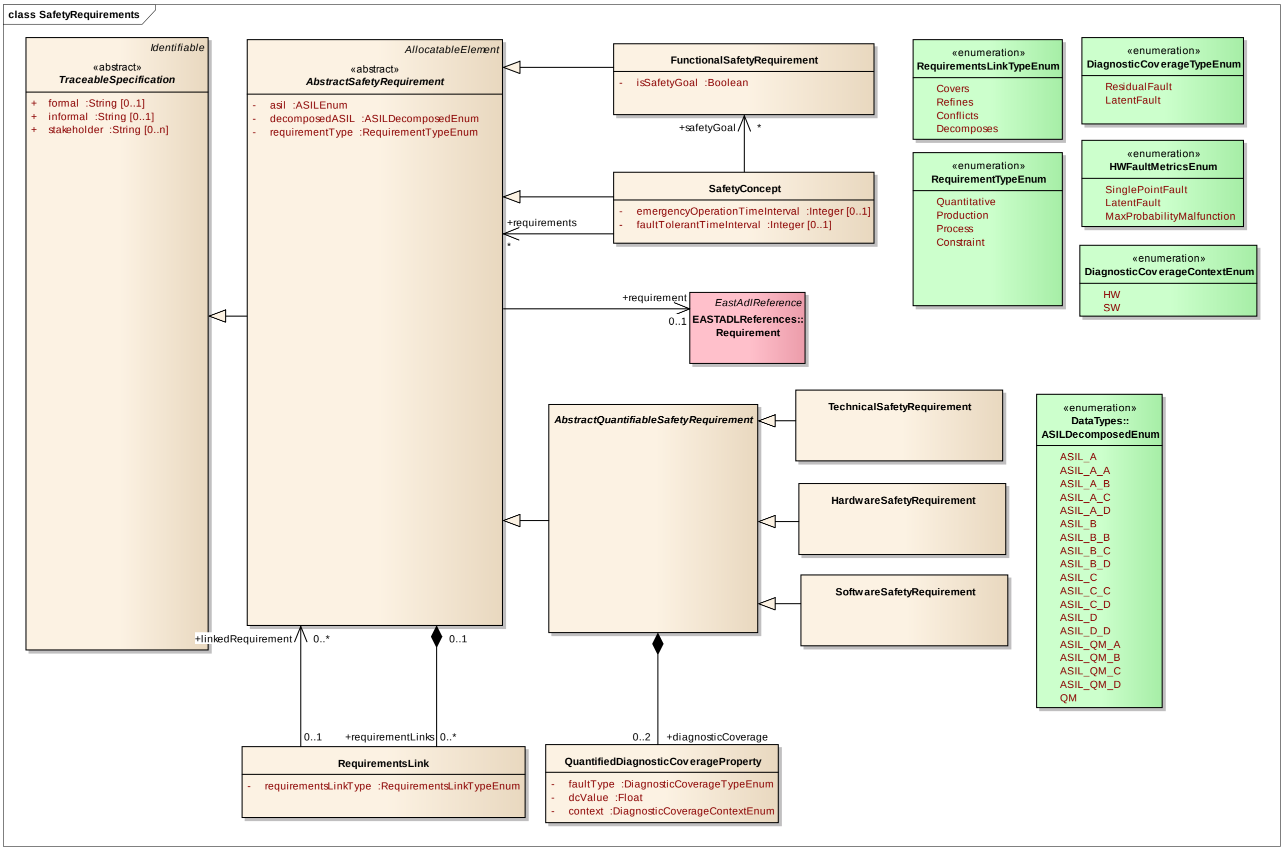


Figure 9: SAFE meta-model safety requirement diagram

Detailed description of the requirements package see D3.1.2.c [7]

Further details according to handling and management of safety requirements according to ISO 26262 part 8 chapter 6 see chapter 8.1.3.2 of this document.

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

4.3.5 Error Model

The SAFE meta-model shall implement an error model that contains the artifacts needed to cover a failure propagation of safety relevant failures identified during qualitative safety analyses according to ISO 26262. Elements that are needed in addition to the already existing artifacts of EAST-ADL are covered by the error model of SAFE meta-model.

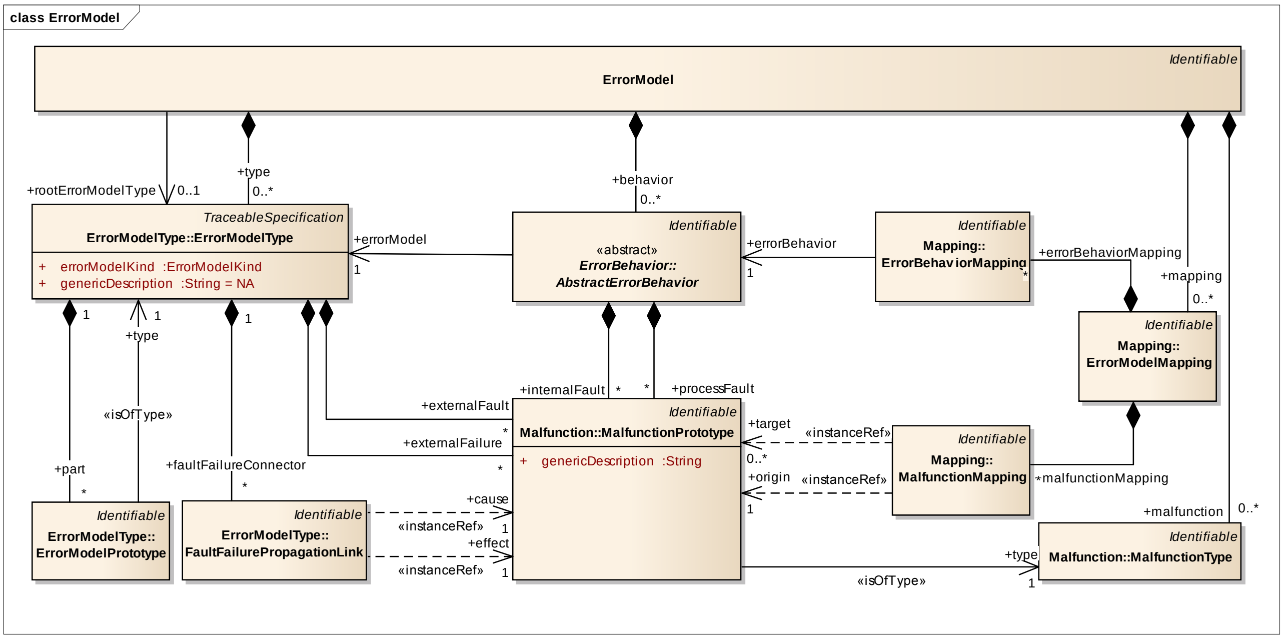


Figure 10: SAFE meta-model error model diagram

Further details according to failure propagation of safety relevant failures see D3.3.1.b [11].

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

5 System Package Specification

This chapter contains the specification of elements that are needed to cover a safety architecture on system level according to ISO 26262 part 4. Elements that are needed in addition to the already existing artifacts of EAST-ADL are allocated to the system package of the SAFE meta-model. This package contains

- description of Functional Safety Extension
- description of Technical Safety Extension
- safety measures and safety mechanisms to avoid, mitigate, detect or control safety relevant failures

The SAFE meta-model shall provide a solution that contains all relevant information about the safety relevant item in a consistent way. This can be reached by maintaining traceability between

- the safety goals analyzed in the Hazard Analysis and Risk Assessment
- the technical solution described in the safety relevant product documentation
- the verification and validation results

5.1 Input needed to start safety relevant product development at the system level

The following information shall be available to start the safety relevant product development at the system level

5.1.1 Item Definition

An Item is a system or array of systems to implement a function at the vehicle level that is able to cause harm to people inside or outside the vehicle.

It shall be possible to describe interfaces, interactions and dependencies to other items. The ISO 26262 is focused on E/E-technologies, therefore the technology used to realize an item shall be categorized into E/E technologies and other technologies.

The item as well as all external measures that are used as an argument for avoiding a violation of a safety goal shall be developed in accordance with ISO 26262.

It shall be ensured that the specified external measures are implemented. The evidence of that shall be part of the safety validation.

The Item Definition shall specify

- vehicle features and correlated vehicle components that are able to cause hazardous events during its lifecycle (e.g. braking, powertrain,…).
- item features and correlated item components that are used to realize safety relevant vehicle features.
- the interfaces of the safety relevant vehicle components to the environment
- the interfaces of the safety relevant item components to the environment
- planned use cases of the item

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

FFG

- safety mechanisms realized by safety relevant vehicle components used to realize safety relevant vehicle features
- safety measures to handle systematic failures caused by development team members

If an already existing item-architecture shall be reused, the existing interfaces, dependencies and interfaces shall be analyzed. If the analysis results changes of the existing item, these changes shall be handled as modification.

Evidence shall be provided that the modification cannot lead to a violation of an already identified safety goal.

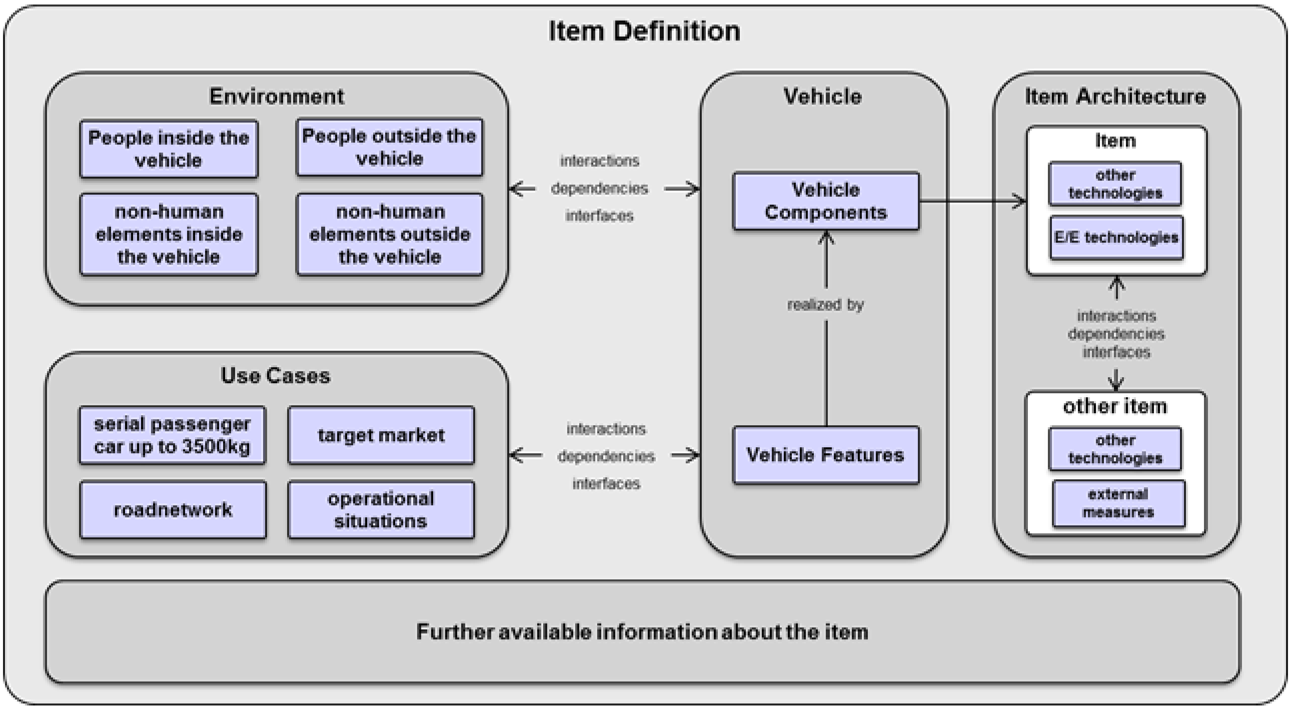


Figure 11: Item Definition

5.1.2 Safety Goals

The Item Definition shall be used as input for execution of a hazard analysis and risk assessment to identify the safety goals and its safe state.

The specified safety goals on vehicle level to avoid the identified hazardous events shall be provided as input to create the functional safety concept.

The safety goals shall be

- described as functional safety requirements and
- allocated to architectural elements of the item.

Further details according to Hazard Analysis and Risk Assessment are described in D3.1.1.c [6].

5.1.3 Functional Safety Concept

The functional safety concept shall be initially created during concept phase. Based on the information given in the item definition and the results of the Hazard Analysis and Risk Assessment the functional safety requirements shall be derived to fulfill the specified safety goals.

The functional safety concept describes

- safety measures
- fault tolerance mechanisms,
- necessary driver actions
- the allocation of the safety measures to the involved architectural elements.

The functional safety concept describes the safety measures that in terms of functional safety are needed to avoid violation of safety goals. It shall contain assumptions about necessary driver actions if needed. The safety measures shall be specified by functional safety requirements.

Traceability between the item feature that causes the safety relevant failure and the safety measures specified to handle the safety relevant failure shall also be part of the functional safety concept.

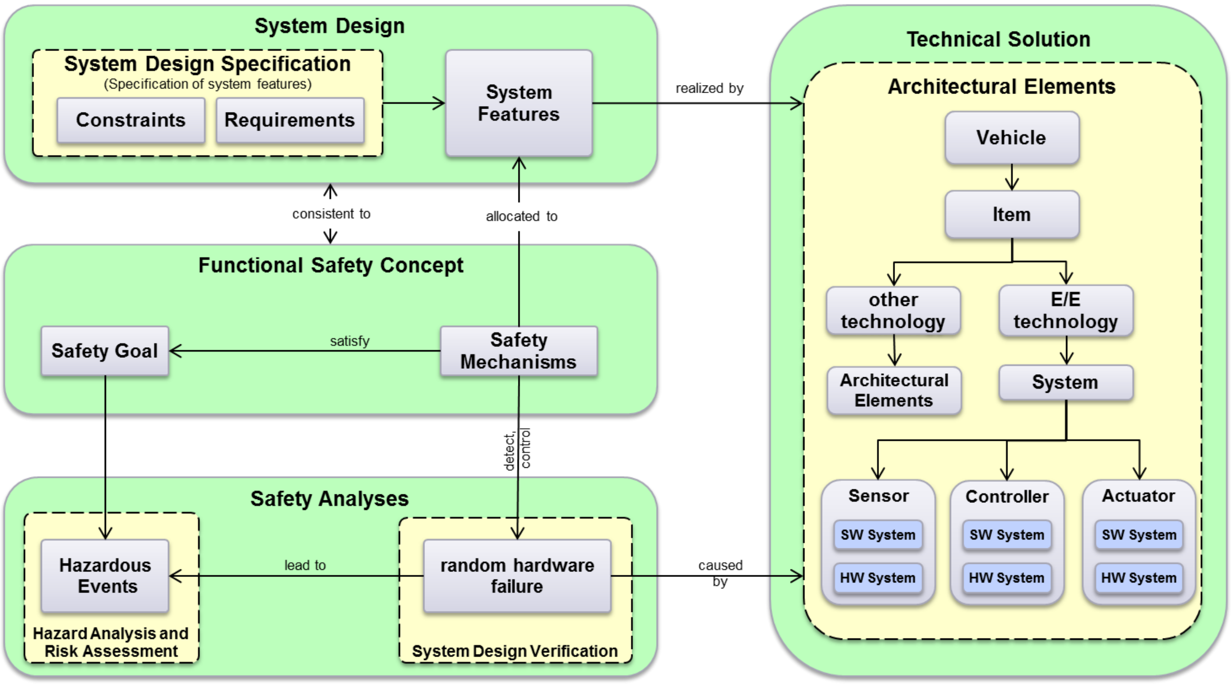


Figure 12: Functional Safety Concept

5.1.3.1 Safe State

A safe state in the scope of ISO 26262 is defined as operating mode of an item without an unreasonable level of risk. That means the item does not show any of the already identified unintended functions that are able to lead to an identified hazardous event.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

5.1.3.2 Fault Tolerant Time Interval

The functional safety concept shall describe the safe state for the specified safety goals. In addition to the safe state the time interval shall be specified starting with the occurrence of the safety relevant failure and ending with the transition to the safe state. This time interval is defined as fault tolerant time interval.

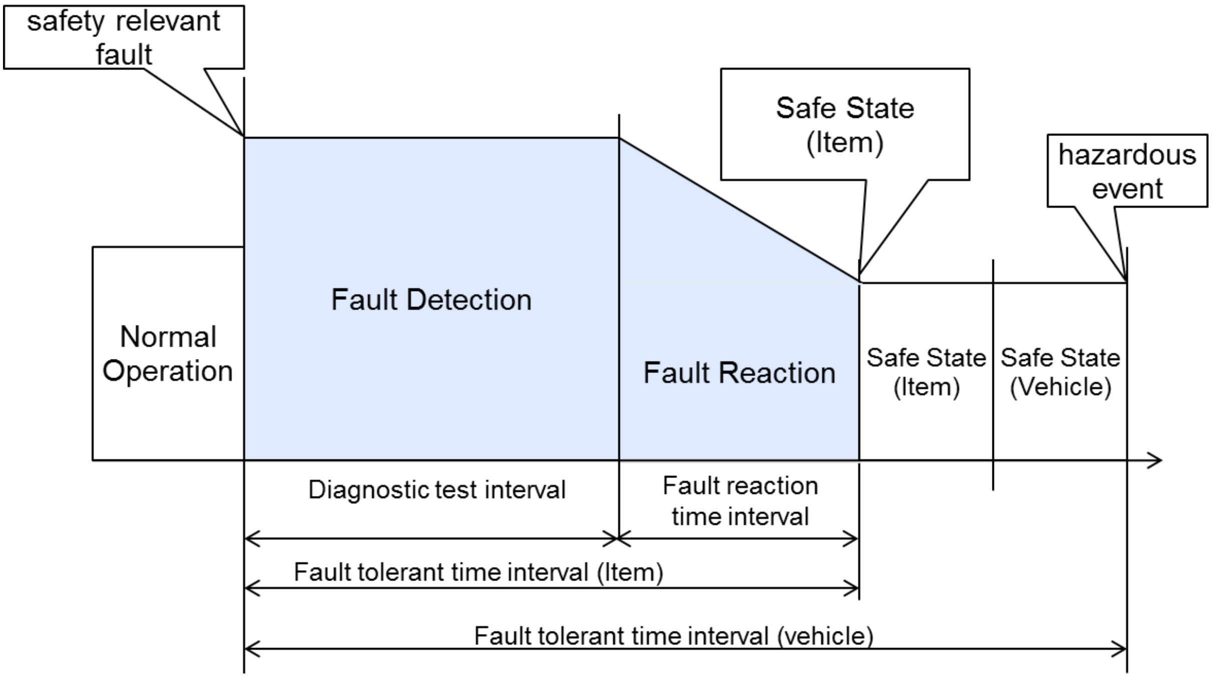


Figure 13: Fault Tolerant Time Interval

If it is not possible to reach the safe state within the defined fault tolerance time interval, a system reaction shall be specified, that is valid for a further time interval. The system behavior for this additional time interval is called warning and degradation concept. The system reaction that is allowed during the warning and degradation time interval shall be specified by safety requirements. These safety requirements shall be allocated to the architectural elements that are used to realize the system reaction specified by the safety requirements.

5.1.3.3 Fault tolerance mechanisms

A fault tolerance mechanism describes the item functionality in the case that a fault does not lead directly to the violation of one or more safety goals and which maintains the item in a safe state (with or without degradation);

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

5.1.3.4 Warning- and Degradation Concept

The warning- and degradation concept is the specification of how to alert the driver of potentially reduced functionality and of how to provide this reduced functionality to reach a safe state.

It is valid for the time interval that is needed to bring the system to the safe state with the defined restrictions of the system behavior. The warning and degradation concept shall be part of the functional safety concept, if needed.

The warning- and degradation concept shall contain:

- the transition to a safe state
- recovering from a safe state.
- fault detection and failure mitigation by switching to a safe state
- driver warning in order to reduce the risk exposure time to an acceptable interval

The specification of the warning and degradation concept and the necessary actions of the driver and other persons who are potentially at risk shall be used as input for the user manual of the item.

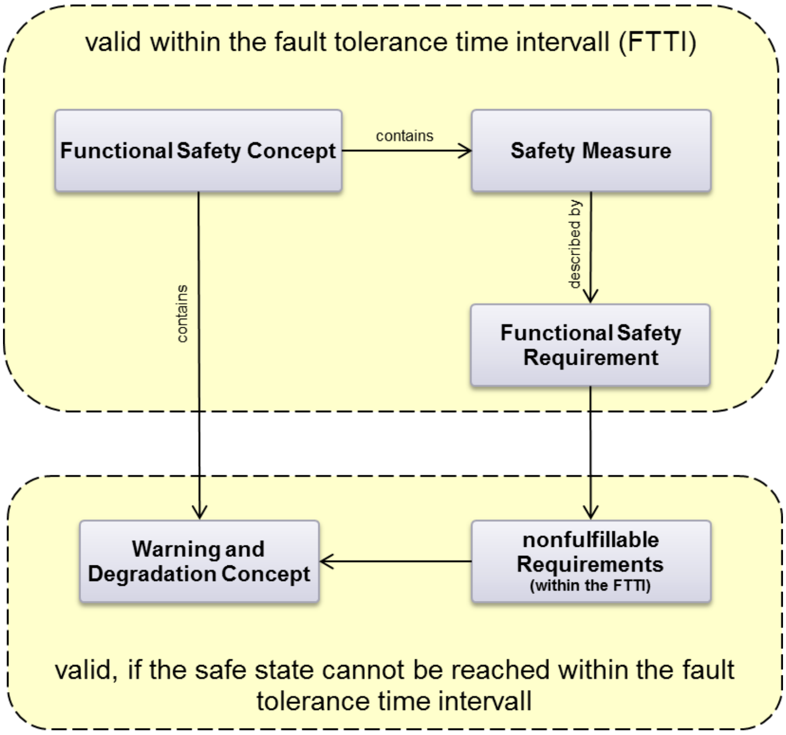

valid within the fault tolerance time intervall (FTTI)
Functional Safety Concept
contains
Safety Measure
described by
Functional Safety Requirement
contains
Warning and Degradation Concept
nonfulfillable Requirements (within the FTTI)
valid, if the safe state cannot be reached within the fault tolerance time intervall

Figure 14 : Warning- and Degradation Concept

5.1.3.5 Necessary Driver Actions

In the case that the driver or any other person at risk has to execute any action to reach the determined safety goal these actions shall be specified in the functional safety concept. To inform the driver or the person at risk a driver warning shall be specified by an adequate media (e.g. engine malfunction indicator lamp, ABS fault warning lamp).

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

The driver action shall be allocated to the corresponding safety goal and the architectural elements that are involved to ensure Traceability.

5.1.3.6 Safety Validation Criteria

The safety validation criteria shall be

- part of the safety concept.
- specified based on the functional safety requirements.
- refined based on the technical safety requirements.

5.1.3.7 Safety Measure

Safety Measures are specified to satisfy the derived Safety Goals. They shall be specified in the functional safety concept to reduce or mitigate the safety relevant failures to a reasonable level of risk.

Safety measures are defined as process activity or technical solution to handle safety relevant failures. Safety measures are described by

- functional safety requirements,
- quantitative safety requirements
- process safety requirements.
- requirements according to production, operation, service and decommissioning instructions, if needed to satisfy at least one allocated safety goal

Safety measure used to handle random hardware failures shall contain

- Specification of transitioning to a safe state;
- arbitration logic to select the most appropriate control request from multiple requests generated simultaneously by different functions
- reference to the safe state that is defined for this safety measure
- reference to the operating modes that are considered during specification of the safety measure
- emergency operation if applicable
- considered functional redundancies

Safety measure used to handle systematic failures shall contain the reference to the safety activity (e.g. verification of Functional Safety Concept)

Further details according to safety activities see chapter 8.1.

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

The following figure is showing the two different kinds of safety measures and their meta-model allocation:

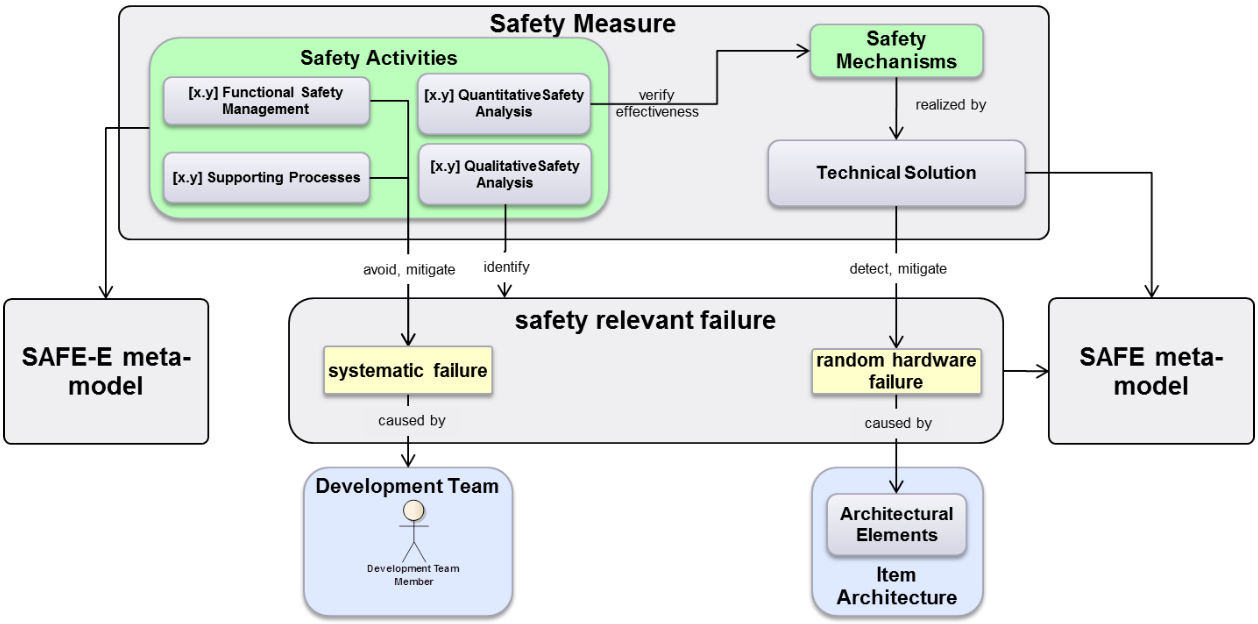


Figure 15: Safety Measures

5.1.3.7.1 *Safety Activities*

Safety Activities shall be specified by process safety requirements or quantitative safety requirements to avoid or mitigate systematic failures. They shall be integrated in the safety-relevant work product safety plan.

The specification of the sequence flow of the safety activities throughout the safety lifecycle is part of the process model. The first proposal is defined in D3.7.a [15] provided in the SAFE-E project.

5.1.3.7.2 *Safety Mechanism*

Safety mechanisms shall be specified as add-on to the technical solution defined in the system design.

Safety mechanisms shall be specified by technical safety requirements derived from the functional safety requirements to fulfill the safety goals identified during hazard analysis and risk assessment.

Safety Mechanisms can be used to achieve different targets. These targets shall be defined for each safety mechanism by selecting one of the following categories:

- for detection, indication and control of faults caused inside the system/Item.
- for detection, indication and control of faults caused by external devices that have influence in the system/Item’s behavior.
- to enable and achieve or maintain the defined safe state
- to implement the warn- and degradation concept

The safety mechanism shall be allocated to the corresponding architectural element in the item architecture.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

Safety mechanisms that are already implemented in the item or planned to be implemented in the item shall not be considered during categorization of hazardous events caused by the item. These safety mechanisms that are already known during execution of hazard analysis and risk assessment shall be described by safety requirements. The safety requirements shall be part of an initial version of a functional safety concept.

The ISO 26262 describes different kinds of safety measures:

- a safety activity to avoid or control systematic failures
- a technical solution to detect or control random hardware failures
- a technical solution to mitigate the harmful effects of random hardware failures

During the derivation of functional safety requirements the preliminary architectural assumptions shall be taken into account.

It shall contain assumptions about necessary driver actions if needed to comply with at least one of the specified safety goals. It shall be available to start derivation of Technical Safety Requirements.

Safety mechanisms to achieve or maintain the safe state

Safety mechanisms that are specified for achieving or maintaining the safe state shall have the following attributes:

- Transition to safe state
- Fault tolerant time interval
- Emergency operation interval, if the safe state cannot be reached immediately
- Measures to maintain the safe state
- behavioral description to achieve or maintain the safe state.
  - operation modes
  - functional redundancies
  - safe state
  - transition from the hazardous event to the safe state
  - allocation to the corresponding warning and degradation concept, if needed

Safety mechanisms to avoid latent faults

Safety mechanisms, that are able to prevent identified multiple-point faults from being latent, shall be specified.

A latent fault in the scope of ISO 26262 is defined as multiple-point fault whose presence is not detected by a safety mechanism nor perceived by the driver within the multiple-point fault detection interval.

5.2 Item Level

In the scope of ISO 26262 an Item is defined as a system or array of systems that contains E/E technology. It is used to implement features at vehicle level that is able to cause harm to people inside or outside the vehicle.

5.2.1 Item Views

The item level shall contain different views:

- Item Element View
- Item Failure View
- Item Feature View

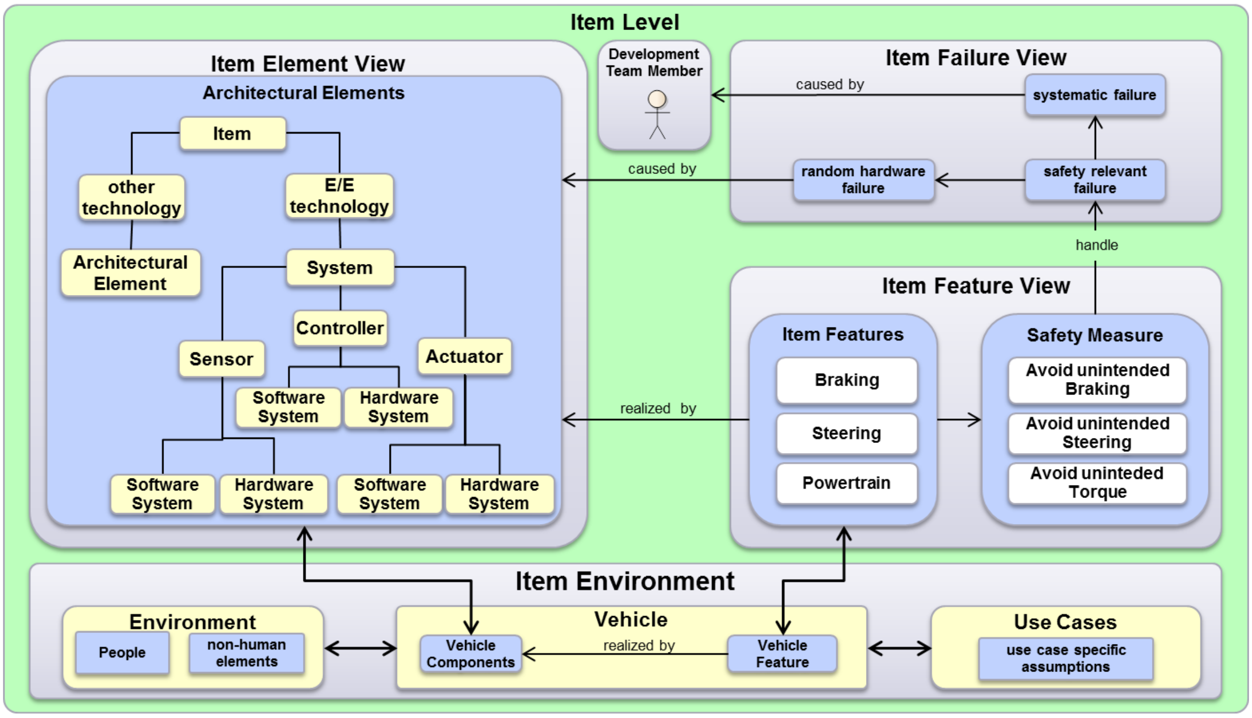


Figure 16: Item Views

5.2.1.1 Item Element view

The item element view shall contain all architectural elements that are used to realize the identified safety relevant item features.

The item element view shall contain the interfaces between the architectural elements of the item.

The Item element view shall contain the allocation between the architectural elements of the item and item features.

The item element view shall contain the interfaces between the item and its environment.

5.2.1.2 Item Features view

The item feature view shall contain all identified safety relevant features of the item.

The item feature view shall contain interfaces between the safety relevant features of the item.

The item feature view shall contain the allocation between safety relevant item features and the architectural elements used to realize the safety relevant item features.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

The item feature view shall contain the interfaces between the safety relevant item features and vehicle features.

5.2.1.3 Item Failure view

The item failure view shall contain all identified safety relevant failures that are caused by

- architectural elements of the item or
- development team members.

Each safety relevant failure shall contain the allocation to the architectural element that has caused the safety relevant failure.

The safety relevant failure shall be identified during qualitative safety analyses at system level (see chapter 5.5).

5.2.2 Item Environment

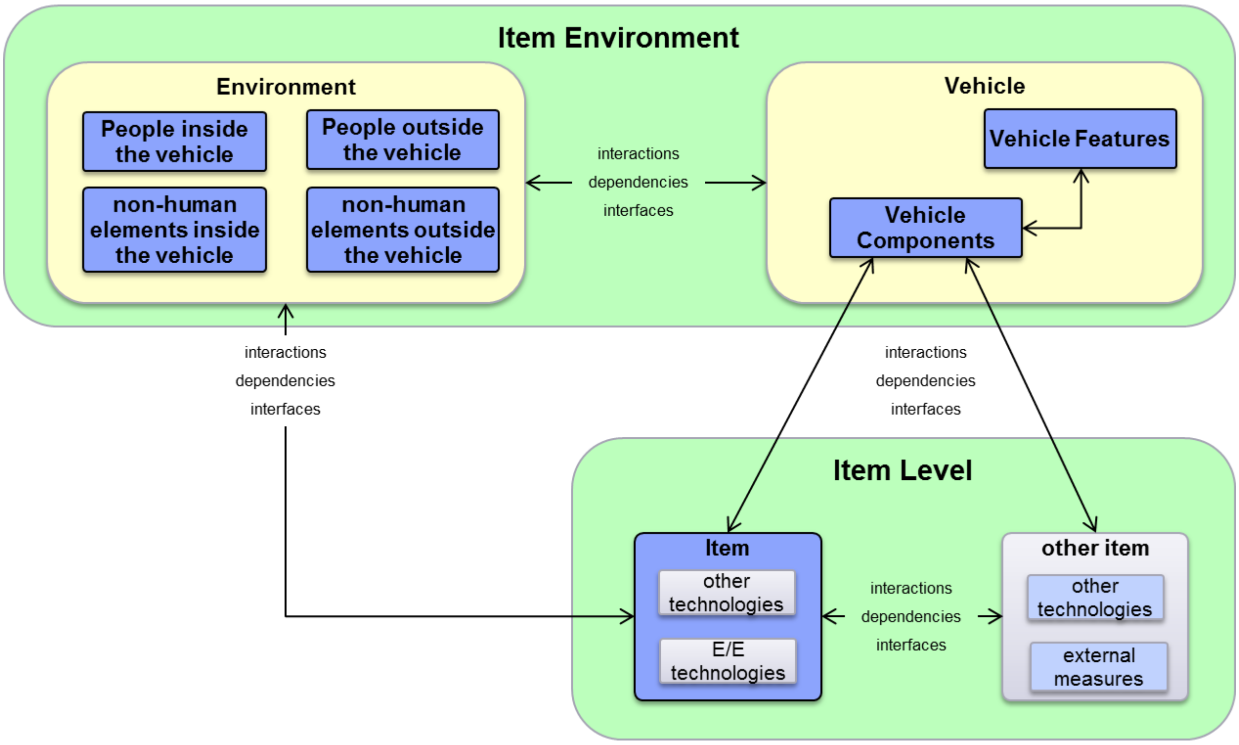


Figure 17: Item Environment

The environment of the item under development contains all elements that can influence the behavior of the item. In case of ISO 26262 the item environment contains all elements of the item under development that can lead to harm of people inside and outside of the vehicle.

All interactions of the item to its environment, that have the potential to influence the safety relevant functionality, shall be specified..If a vehicle contains more than one item, the model shall also contain the interactions between the items and the interfaces between each item and the environment.

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

FFG

Behavioral interactions shall be captured by describing operational scenarios and the corresponding effects on the item or the items.

5.2.2.1 Vehicle Feature

A vehicle feature is realized by vehicle components. Each vehicle component consists of one or more items realized by E/E-technology or other technology.

5.2.2.2 Other item

In most cases the defined item has interfaces to other items that are not in scope of development, but they are also items in the scope of ISO 26262. That means they are also systems or arrays of systems that contain E/E-technology. These interfaces shall also be described in the item definition.

Therefore the SAFE-meta model shall contain architectural elements for the item and for other items to enable the specification of the item boundaries.

5.2.2.3 Other technologies

If items contain elements realized by other technologies, the implementation of those elements shall be ensured through measures outside the scope of ISO 26262. No ASIL shall be allocated to the elements allocated to other technologies.

5.2.2.4 External Measures

External Measures are safety measures implemented with E/E-technologies. They are applied in a system or a system-array allocated to other items. Other items shall also be developed in accordance with ISO 26262.

5.2.2.5 Item Element

Item elements are sub-elements of the item that are used to realize item features, e.g. HW-parts, Software Units, Connectors,

5.2.3 Item Boundary

Based on the fact that a vehicle is developed in different organizations or different development teams the vehicle components are split up into separate items. The boundaries of each item shall be defined as detailed as possible to reduce the probability of systematic failures during integration of the different items to one vehicle component.

The environment of the item under development contains all elements that can influence the behavior of the item in a way that can lead to harm of people inside and outside of the vehicle. All interactions of the item to its environment that have the potential to influence the safety relevant functionality, shall be specified.

If a vehicle contains more than one item, the model shall also contain the interactions between the items and the interfaces between each item and the environment.Behavioral interactions shall be captured by describing operational scenarios and the corresponding effects on the item or the items.

The item as well as all external measures that are used as an argument for avoiding a violation of a safety goal shall be developed in accordance with ISO 26262. It shall be ensured that the specified external measures are implemented. The evidence of that shall be part of the safety validation.

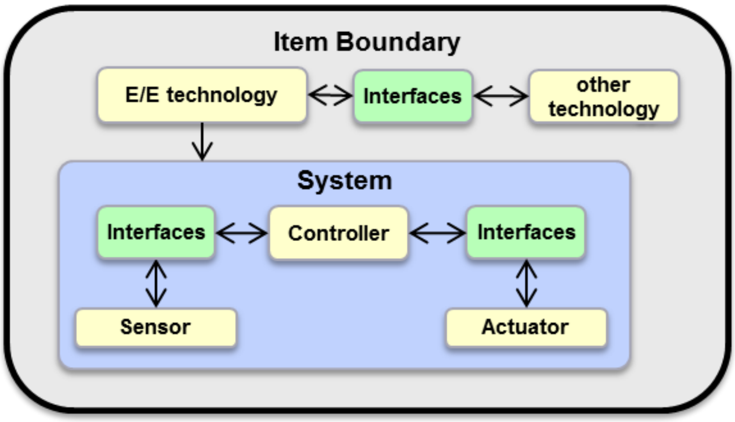


Figure 18: Item Boundary

5.2.4 Item Architecture

This section describes the identified architectural elements to specify a safety relevant item in scope of ISO 26262.

An item consists of one or more systems to implement a vehicle function. The safety-relevant system shall contain at least three system components, one sensor, one actuator and one controller.

Each system component can be decomposed from software system and/or hardware system.

A software system consists of one or more software partitions.

A software partition consists of one or more software units.

A hardware system consists of one or more hardware components.

A hardware component consists of one or more hardware parts.

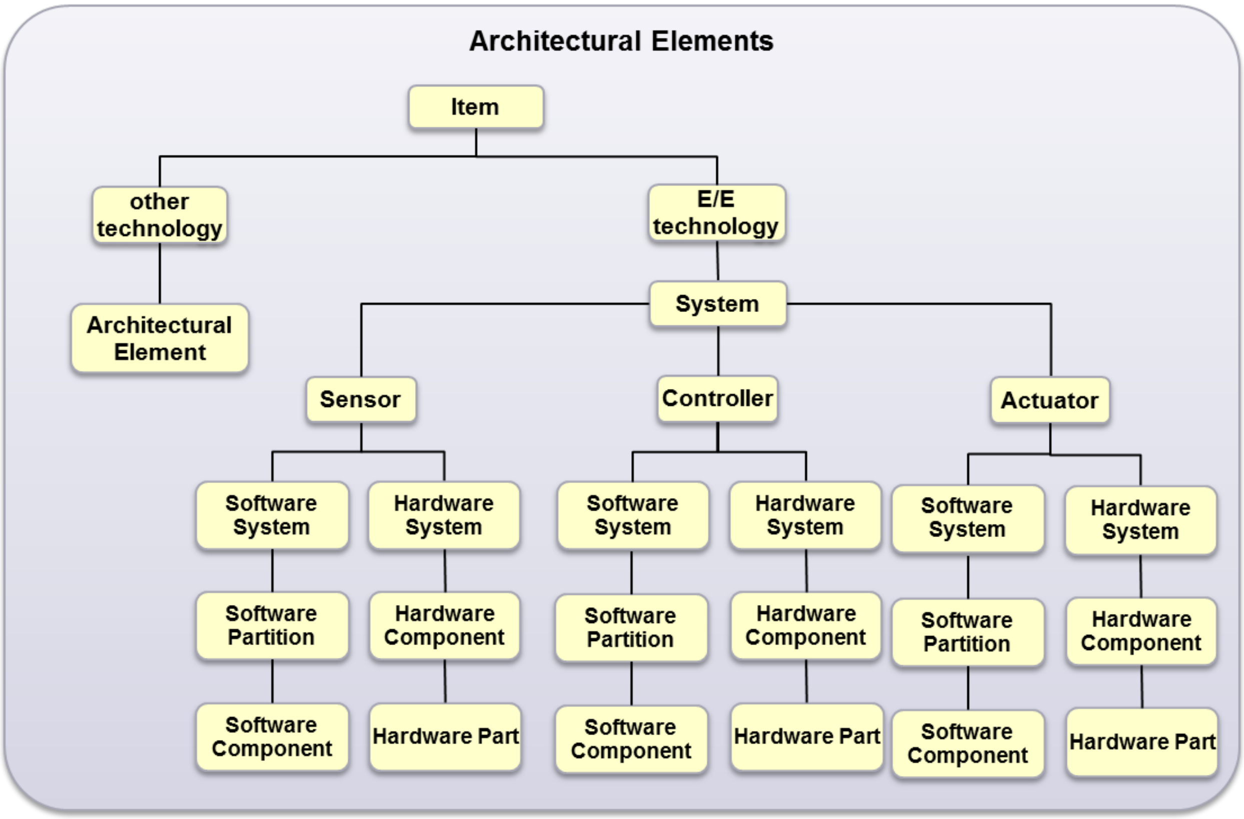


Figure 19: Architectural elements on item level

It shall be possible to

- add a preliminary description to the preliminary architectural elements
- allocate functional safety requirements to preliminary architectural elements
- create architectural elements as well as preliminary architectural elements
- allocate other technologies to an architectural element
- allocate external measures to an architectural element

It shall be ensured that the preliminary architectural assumptions defined in the concept phase are consistent with the preliminary architectural assumptions in the sub-phases. Therefore traceability shall be established between the architectural assumptions and the derived requirements.

The maturity of the architectural elements shall also be part of the SAFE meta-model (see 8.1.3.3)

5.2.4.1 Item Feature

Item features are functionalities that shall be implemented in the item, e.g. braking, steering. In the scope of ISO 26262 the main topic is to avoid or at least mitigate safety relevant malfunctions of the item. Therefore the item feature view shall also contain features that are implemented to avoid safety relevant malfunctions, e.g. avoid unintended braking.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

5.2.4.2 Item Interfaces

The item architecture shall contain the elements of the item under development and the specification of the item boundary. The item boundary shall contain the interfaces of the item to other items and its environment.

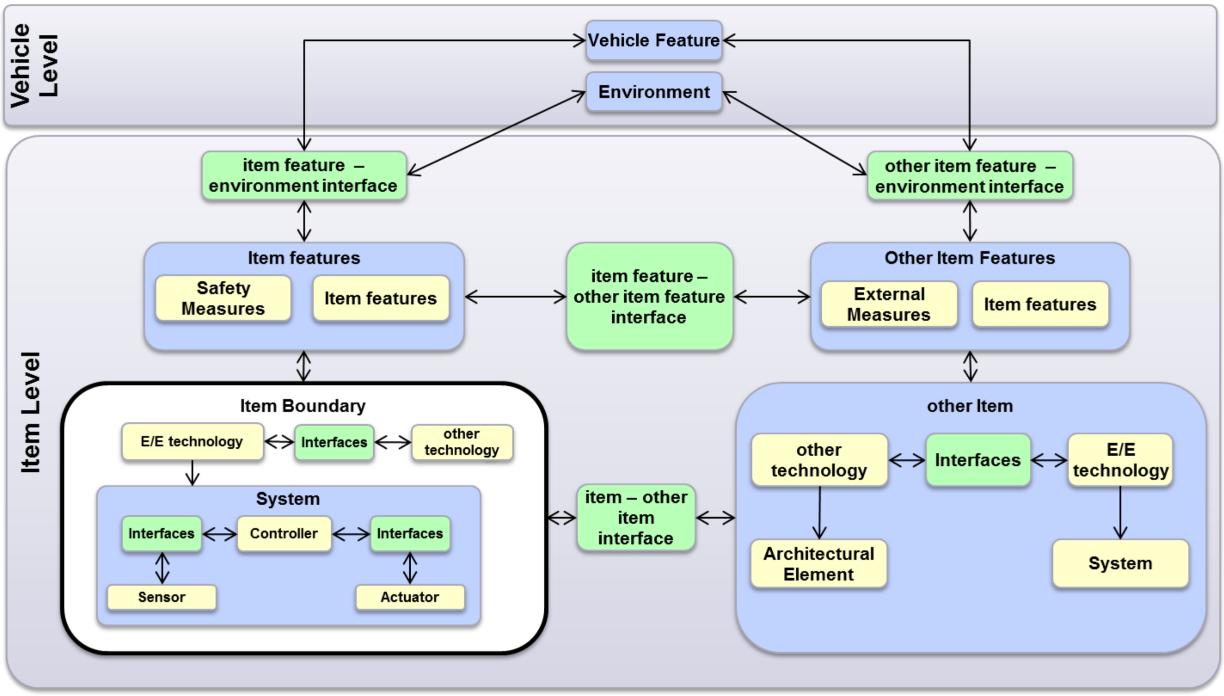


Figure 20: Item Interfaces

The item boundary shall contain:

- ***Interfaces to other technologies***
  The interfaces of the item to the elements realized by other technology shall be specified.
- ***Interfaces to the environment***
  The interfaces of the item to its environment shall be specified
- ***Interfaces to other items***
  The interface of the item to other items shall be specified

5.2.5 Development Category

It shall be possible to specify the category of the item under development in the first development phase. There are two different categories defined in the ISO 26262:

- ***New***
  This development category shall be selected for items that are developed completely new.
- ***Modification***
  This development category shall be used, if an already existing safety relevant item shall be modified for a new use case.

The development category shall be selected once for an item development.

5.2.6 Safety Element out of Context (SEooC)

A safety element out of context (SEooC) is a system or an element of a system that is not developed for a particular vehicle. The safety element out of context shall be developed based on assumptions, that describe the context of the element for that it is developed. This could enable the development of generic elements.

The SEooC shall be integrated to a new application by execution of an impact analysis of the assumptions specified for the SEooC and the requirements of the new application that have impact to the functionality of the SEooC.

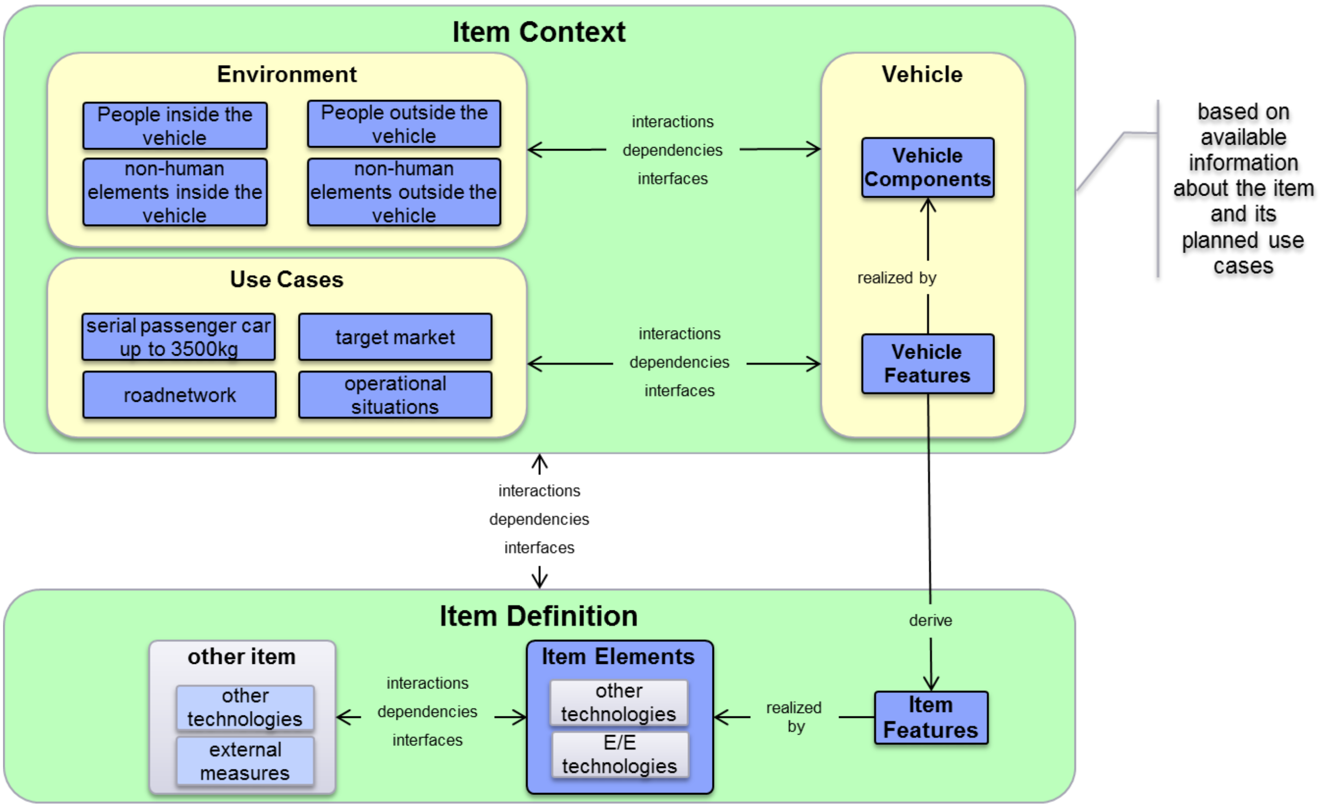


Figure 21: Safety Element out of Context (SEooC)

5.3 System Level

The System Level contains the description and the architecture of the safety relevant system components.

5.3.1 System Architecture

The following system architecture shall be specified during safety relevant product development at system level.

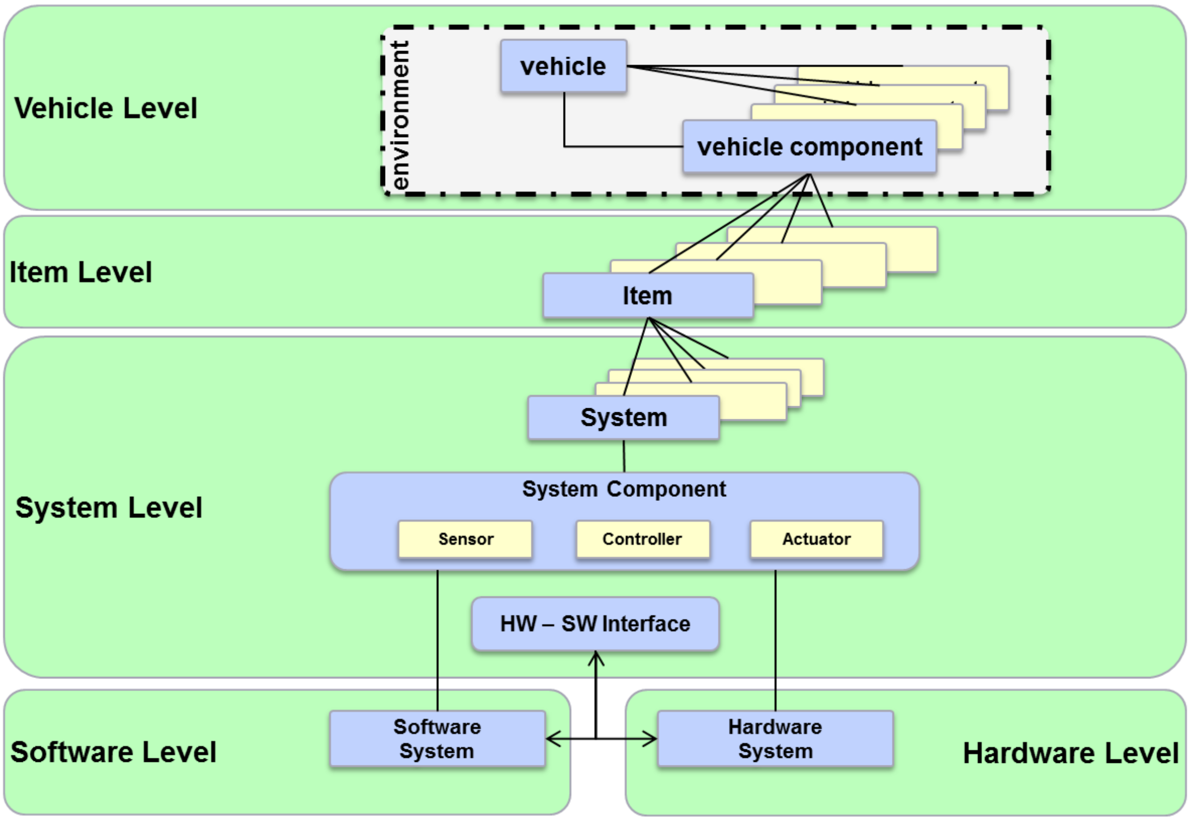


Figure 22: System Architecture

5.3.2 System Array

In the case that the item contains more than one system the item architecture on system level shall contain the architectural elements of all systems that are part of the item.

Preliminary architectural assumptions that are already available during creation of the item architecture of the item shall be considered.

Architectural Elements that are not finally verified or validated are called preliminary architectural elements.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

5.4 System Design

The objective of the safety activity System Design in scope of the ISO 26262 is, to develop and verify the system design and the technical safety concept that comply with the functional requirements and the technical safety requirements specification of the item

5.4.1 System Design Specification

The System Design specification shall contain the system requirements allocated to the architectural elements on system level and the specification of the system functionality.

The following system properties shall be described in the meta-model:

- external interfaces (e.g. communication interfaces, user interfaces,…)
- system constraints (e.g. environmental conditions, functional constraints,…)
- system configuration (e.g. calibration data,…)
- design constraint

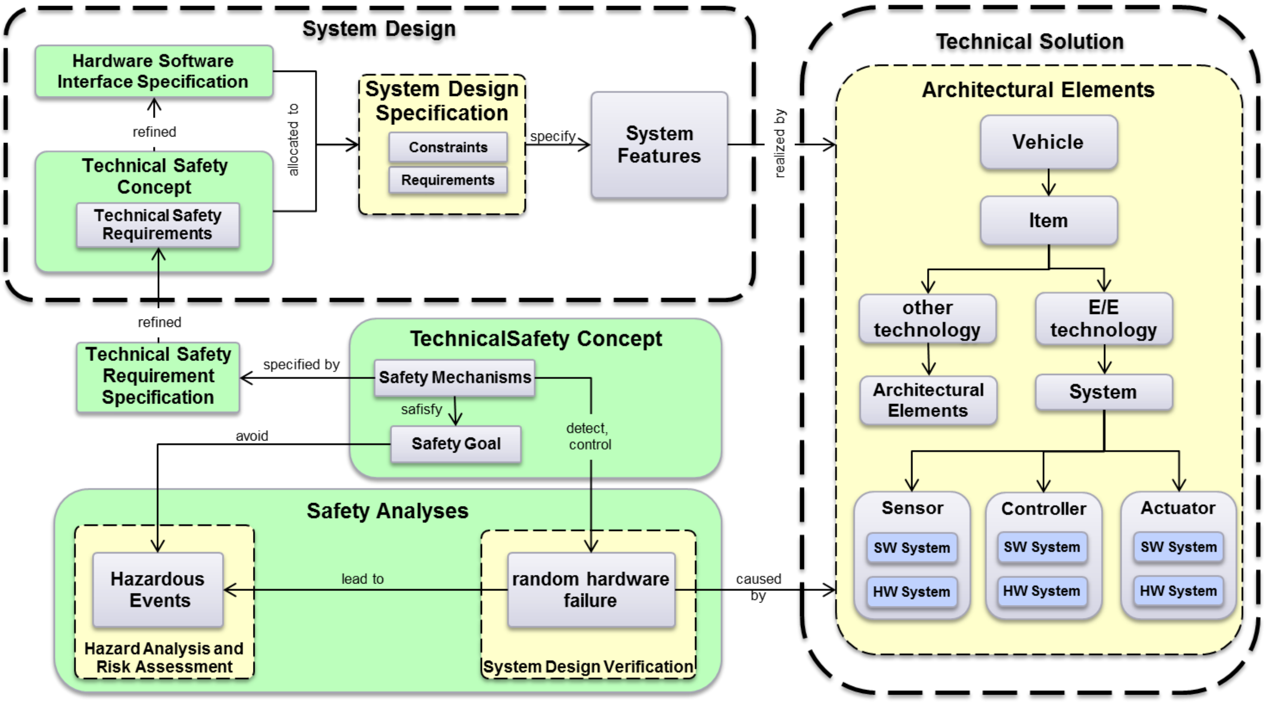


Figure 23: System Design

The System Design Specification shall contain the specification of all system features. The system features that are identified as safety relevant as well as those that are not safety relevant shall be specified in the system design.

The features that are planned to detect and/or handle safety relevant failure shall be specified in the functional safety concept. The technical solution of those features shall be specified in the technical safety concept. These three documents shall be consistent throughout the entire safety lifecycle.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

To reduce systematic failures, well-trusted automotive systems design principles should be applied. A well-trusted design principle is the way to get a good design for the planned system that is already gone several times successfully.

A decision not to re-use well-trusted design principles should be justified for requirements classified with ASIL D. This decision is not needed for requirements classified with QM, ASIL A, ASIL B, ASIL C.

The technical solution described in the system design specification shall contain safety mechanisms to avoid, control or mitigate safety relevant failures that are identified during the safety analyses.

5.4.2 Hardware-Software Interface

The Hardware Software Interface Specification (HSI) shall describe the safety relevant interfaces between hardware- and software-system for each E/E-System component (see Figure 22: System Architecture)

The Hardware Software Interface Specification (HSI) shall be provided as input for the product development on hardware level and on software level.

The HSI shall

- describe the diagnostic capabilities of the hardware elements and their use by software
The relevant diagnostic capabilities consist of
- the hardware diagnostic features; and
- the diagnostic features concerning the hardware, to be implemented in software
- describe all safety-relevant dependencies between hardware and software
- be specified during the system design and will be refined during the hardware- and the software development
- be consistent with the technical safety concept and shall specify the interaction between the hardware elements and the software elements
- contain relevant operating modes of hardware elements/parts and the relevant configuration parameters
- contain the hardware features that ensure the independence between elements and that support software partitioning
- contain shared and exclusive use of hardware resources
- contain the access mechanism to hardware devices
- contain the timing constraints defined for each service involved in the technical safety concept
- be consistent with the Technical Safety Concept.
- be specified during system design.
- be refined during software- and hardware development.

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

FFG

The following figure is showing the proposal for the implementation of the Hardware Software Interface Specification as part of the Technical Safety Extension of the SAFE meta-model:

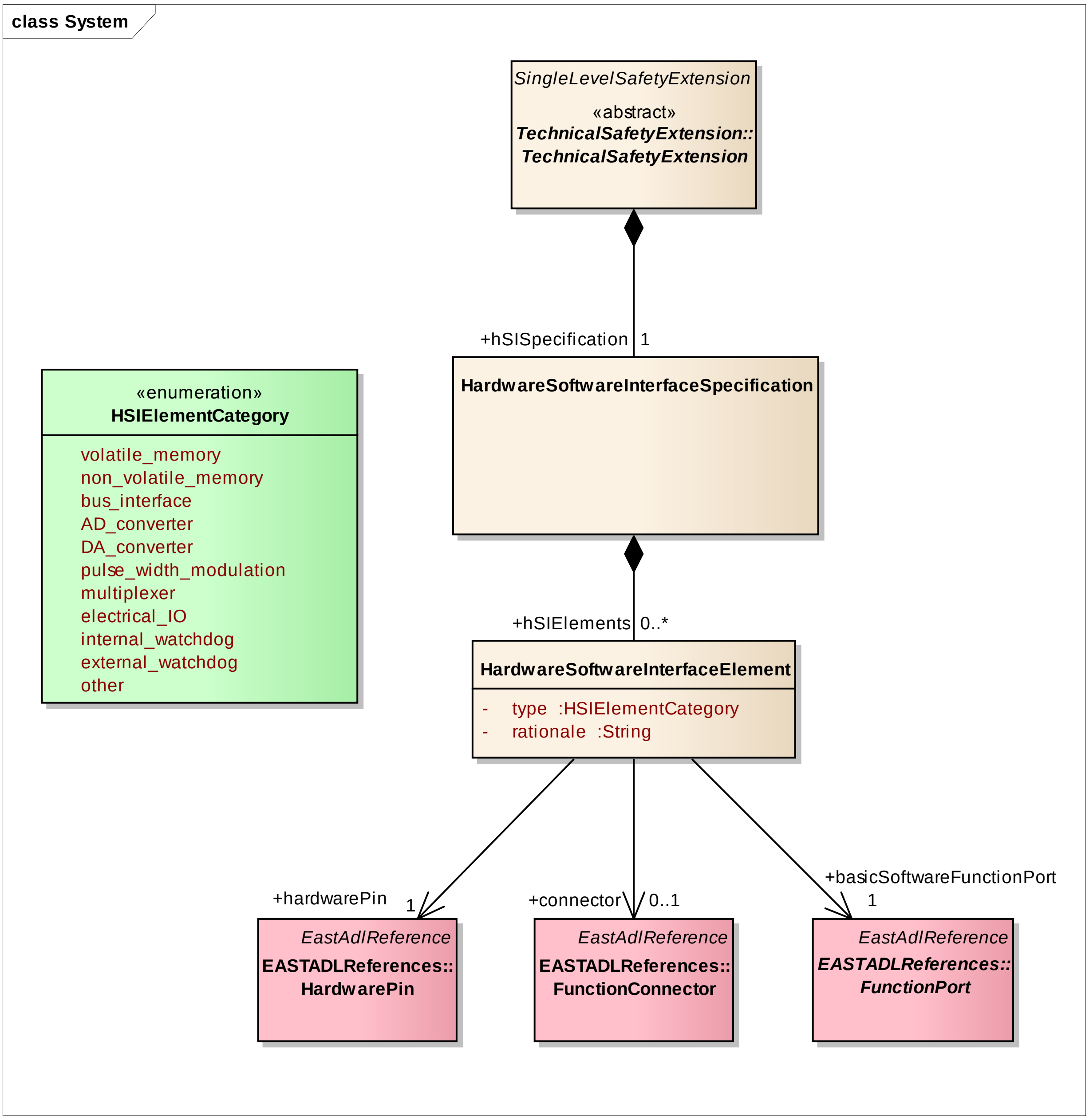
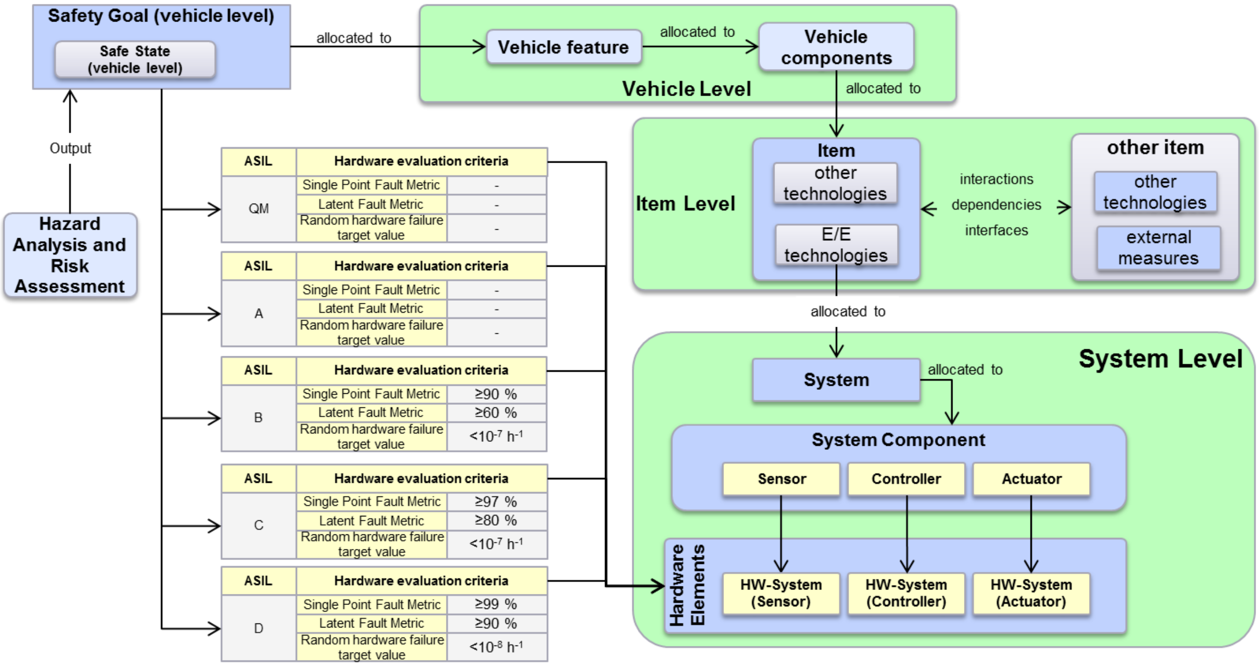


Figure 24: SAFE meta-model: Hardware Software Interface Specification

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

5.4.3 Allocation of Hardware evaluation criteria

Based on the ASIL allocated to the safety goals defined on vehicle level the target values for hardware evaluation criteria for the affected hardware system shall be derived.



ASIL	Hardware evaluation criteria	
QM	Single Point Fault Metric	-
	Latent Fault Metric	-
	Random hardware failure target value	-

ASIL	Hardware evaluation criteria	
A	Single Point Fault Metric	-
	Latent Fault Metric	-
	Random hardware failure target value	-

ASIL	Hardware evaluation criteria	
B	Single Point Fault Metric	≥90 %
	Latent Fault Metric	≥60 %
	Random hardware failure target value	$<10^{-7}$ h^{-1}

ASIL	Hardware evaluation criteria	
C	Single Point Fault Metric	≥97 %
	Latent Fault Metric	≥80 %
	Random hardware failure target value	$<10^{-7}$ h^{-1}

ASIL	Hardware evaluation criteria	
D	Single Point Fault Metric	≥99 %
	Latent Fault Metric	≥90 %
	Random hardware failure target value	$<10^{-8}$ h^{-1}

Figure 25: Hardware Evaluation Criteria

The hardware evaluation criteria consist of the following quantitative values for each safety goal:

- Single-point fault metric
- Latent fault metric
- Random hardware failure target value

The target value for single-point fault metric, latent-point fault metric and random hardware failure target value for the affected hardware-systems shall be specified in the System Design.

Two alternative methods are proposed to provide evidence that the residual risk of a safety goal violation due to random hardware failures of the item is sufficiently low:

- Probabilistic Metric for random hardware failures (PMHF)
- Individual evaluation of each random hardware failure, that leads to a violation of the considered safety goal.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

5.4.3.1 Single point fault metric

For each safety goal a quantitative target value for the single point fault metric shall be specified.

In the scope of ISO 26262 the following sources are defined:

- Hardware architectural metric calculation applied on similar well-trusted design principles.
- Single-point fault metric based on ISO 26262-5 Table 4

Single-point fault metric	
ASIL B	≥90 %
ASIL C	≥97 %
ASIL D	≥99 %

5.4.3.2 Latent fault metric

For each safety goal a quantitative target value for the latent fault metric shall be specified.

In the scope of ISO 26262 the following sources are defined:

- Hardware architectural metric calculation applied on similar well-trusted design principles.
- Latent fault metric based on ISO 26262-5 Table 5

Latent fault metric	
ASIL B	≥60 %
ASIL C	≥80 %
ASIL D	≥90 %

5.4.3.3 Random hardware failure target value

For each safety goal a quantitative target value for the probability of safety relevant random hardware failure shall be specified.

In the scope of ISO 26262 the following sources are defined:

- Field data from similar well-trusted design principles
- Quantitative analysis techniques applied to similar well-trusted design principles using single-point fault and latent fault metric defined.
- Random hardware failure target values based on ISO 26262-5 Table 6

Random hardware failure target values	
ASIL B	$<10^{-7}\ h^{-1}$
ASIL C	$<10^{-7}\ h^{-1}$
ASIL D	$<10^{-8}\ h^{-1}$

5.4.4 Allocation of ASIL to System Design Elements

Each system design element shall inherit the highest ASIL from the technical safety requirements that specify safety mechanisms realized in the element.

Technical Safety requirements contained in the Technical Safety Concept shall be allocated to the architectural elements in the item architecture on system level. Each architectural element inherits the highest ASIL of all allocated technical safety requirements. This classification shall be done automatically.

If one of the architectural elements is divided into Sub-Elements, the classification of the ASIL of the Sub-Elements shall be done by regarding the criteria for coexistence defined in ISO 26262 part 9 Chapter 6. If any of the defined criteria for coexistence is met, it shall be documented.

Technical Safety requirements contained in the Technical Safety Concept shall be allocated to the architectural elements in the item architecture on system level. Each architectural element inherits the highest ASIL of all allocated technical safety requirements. This ASIL inheritance shall be done automatically.

If one of the architectural elements is divided into Sub-Elements the classification of the ASIL of the Sub-Elements shall be done by regarding the criteria for coexistence defined in ISO 26262 part 9 Chapter 6. If any of the defined criteria for coexistence is met, it shall be documented.

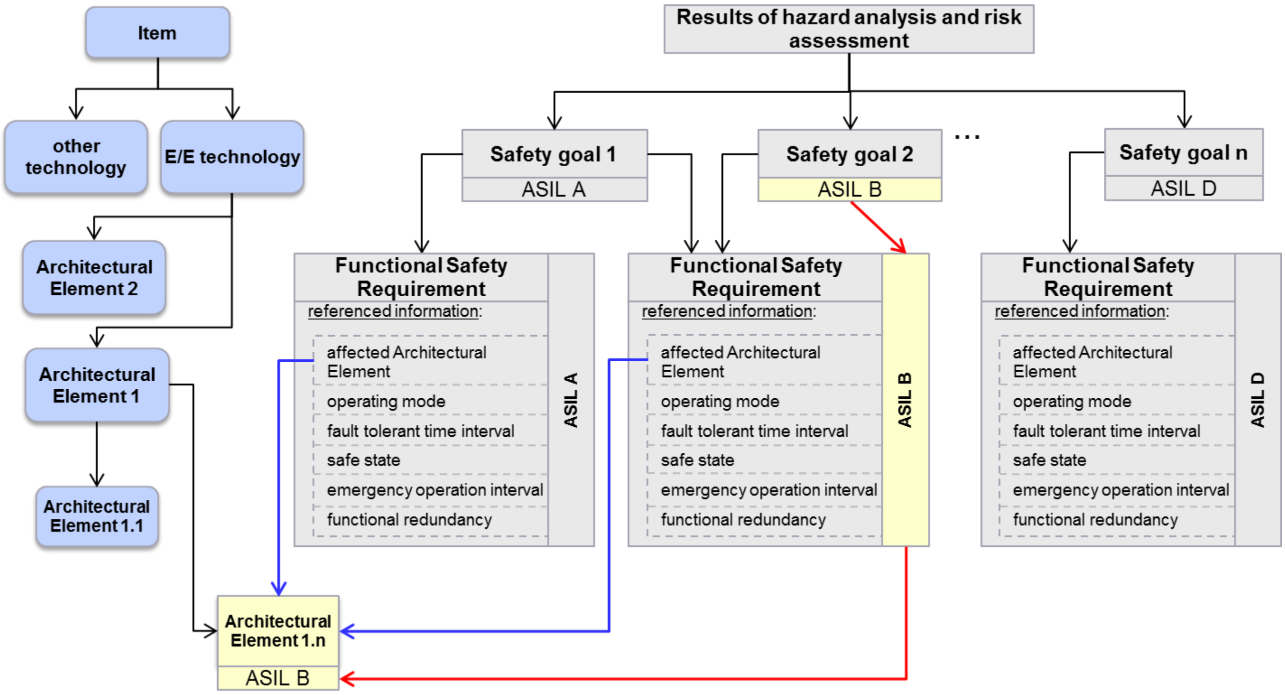


Figure 26: Allocation of ASIL to system design elements

5.4.5 Safety Concept on System Level

The functional safety concept initially created during concept phase shall be refined during product development at the system level in scope of ISO 26262.

5.4.5.1 Safety Requirements at system level

Safety mechanisms shall be defined as add-on to the already existing system design to

- detect or control random hardware failures that have the potential to lead to a violation of the allocated safety goal and/or
- avoid or mitigate systematic failures that have the potential to lead to a violation of the allocated safety goal.

The safety mechanisms shall be specified by technical safety requirements. These requirements shall be specified in accordance to the following information provided as input for the safety activity ***specification of technical safety requirement***:

- System-Interfaces
- Item-interfaces
- environmental conditions or functional constraints
- system configuration requirements
- system design specification
- safety goals allocated to the system
- functional safety requirements allocated to the system

The specified technical safety requirements shall be allocated to

- the architectural elements of the system architecture
- the preliminary architectural assumptions
- the functional safety requirements

to ensure the bidirectional traceability.

Technical Safety Requirements shall be

- compliant to the requirements defined in the Functional Safety Concept
- consistent to the requirements defined in the Functional Safety Concept
- compliant to the preliminary architectural design
- consistent to the preliminary architectural design

System Interfaces

Internal and external interfaces of safety-related elements shall be defined, in order to avoid other elements having adverse safety-related effects on the safety-related elements.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

System configuration

In the case that a safety-relevant system uses configuration and/or calibration data that are able to influence the safety-relevant system behavior the calibration/configuration data shall be verified during item integration and testing.

Evidence shall be provided on system- and/or item-level that the calibration data/configuration data are compliant with the corresponding safety requirements.

Each configuration at implementation level that is intended for serial production shall be verified during item integration and testing.

Every configuration at implementation level that is intended for serial production shall be verified during item integration and testing.

Further details according to item integration and testing see 8.1.1

5.4.5.2 ASIL Decomposition

The ASIL tailoring during the design process is called ASIL decomposition. ASIL decomposition shall be done in accordance with one of the decomposition schemes given in the ISO 26262 part 9 chapter 5.

The objective of ASIL decomposition is to provide rules for decomposing safety requirements into redundant safety requirements to allow ASIL tailoring at the next level of detail.

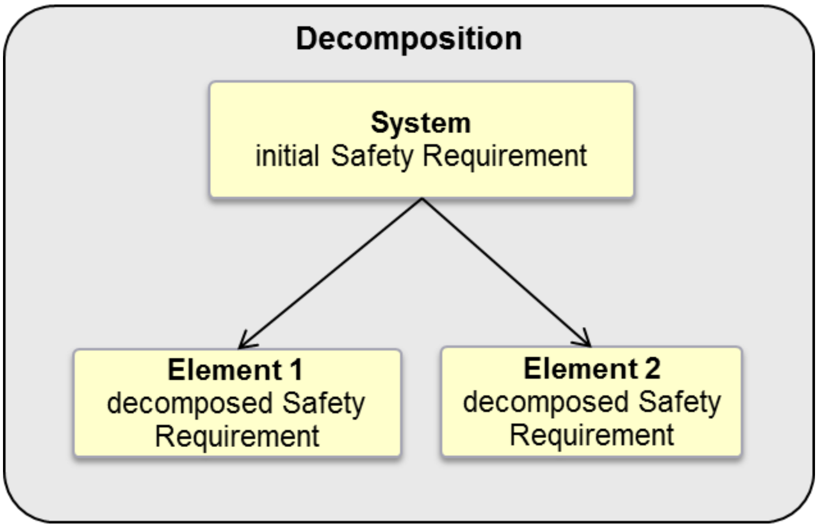

Decomposition
System
initial Safety Requirement
Element 1
decomposed Safety Requirement
Element 2
decomposed Safety Requirement

Figure 27: Decomposition

ASIL decomposition can obtain the benefit

- to implement safety requirements redundantly by sufficiently independent architectural elements
- to assign a potentially lower ASIL to the decomposed safety requirements.

If the architectural elements are not sufficiently independent, the redundant requirements inherit the same ASIL as the safety requirement allocated from the higher development level (previous safety activity).

Evidence of sufficiently independent architectural elements can be provided by analysis of dependent failures caused by the correlated architectural elements. More details about analysis of dependent failures will be part of D3.2.2.b [9]

ASIL decomposition allows the measuring of the ASIL of safety requirements that are allocated to several architectural elements and the same safety goal.

Safety requirement, that are used to specify the evaluation of the hardware architectural metric and the evaluation of safety goal violations due to random hardware failures will not be changed by ASIL decomposition

If ASIL-Decomposition results in the allocation to the initial functionality and an associated safety mechanism then the safety mechanism inherits the highest ASIL of the decomposition. The following figure is showing an example:

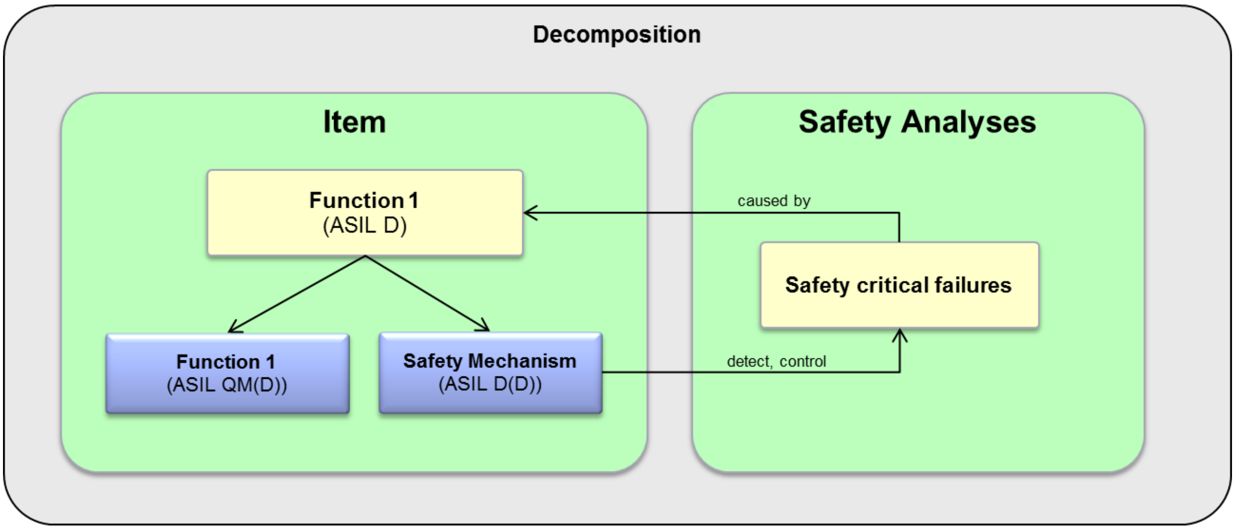


Figure 28: Decomposition Function + Safety Mechanism

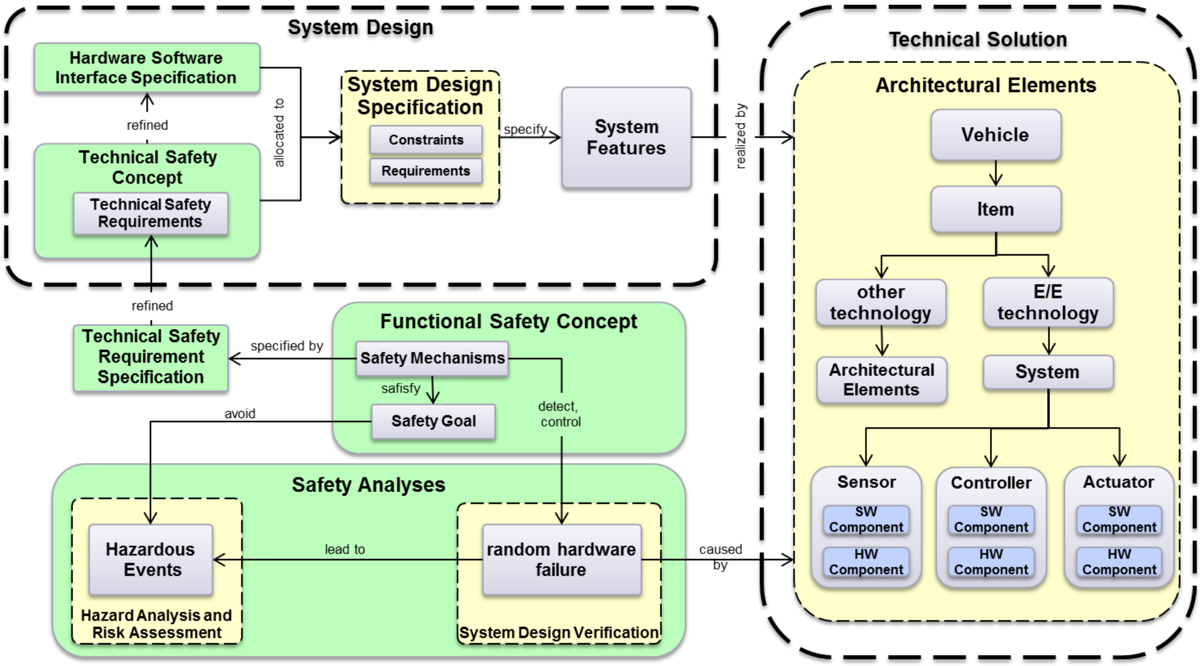
SAFE meta-model shall provide a solution to show the different kinds of safety requirements:

- safety requirements without decomposition
- safety requirements that are part of a decomposition

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

5.4.5.3 Technical Safety Concept

The Technical Safety Concept is derived from the Functional Safety Concept. It contains the specification of the technical solution planned to realize for the specified safety mechanisms.



.

Figure 29: Technical Safety Concept

The technical solution shall contain safety mechanisms

a) to the detection, indication and control of faults in the system itself;

b) to the detection, indication and control of faults in external devices that interact with the system;

c) to achieve or maintain a safe state;

d) to detail and implement the warning and degradation concept; and

e) to avoid latent faults

The Technical Safety Concept shall contain requirements according to production, operation, service and decommissioning like described in the following list, if they are needed to fulfill at least one of the safety goals allocated to the item:

- Assembly instructions
- Safety-related special characteristics
- Requirements for insurance of proper identification of safety-relevant systems or system elements (e.g. labels)
- Verification methods/measures for production
- Service Requirements for diagnostic data or service notes
- Decommissioning requirements

The allocation of the safety mechanisms to the system elements (e.g. hardware parts, software units,…) shall be described in the Technical Safety Concept.

5.5 Safety Analyses at the system level

Safety Analyses shall be executed as central topic of the product development to identify safety relevant failures caused by any element of the item under development, that are able to cause harm to people.

Two different kinds of safety analyses are defined in scope of ISO 26262

- ***qualitative safety analysis***
  identification of safety relevant failures based on the already existing item documentation
- ***quantitative safety analyses***
  verification of the effectiveness of the safety mechanisms specified as add-on for the system-design of the item.

Safety relevant failures identified during qualitative safety analyses shall be provided as input to the safety activity System Design according to ISO 26262 part 4 chapter 7.

5.5.1 System Design Analysis

Qualitative safety analyses shall be executed in parallel to the system development to identify the safety relevant failures caused by the architectural elements of the item under development.

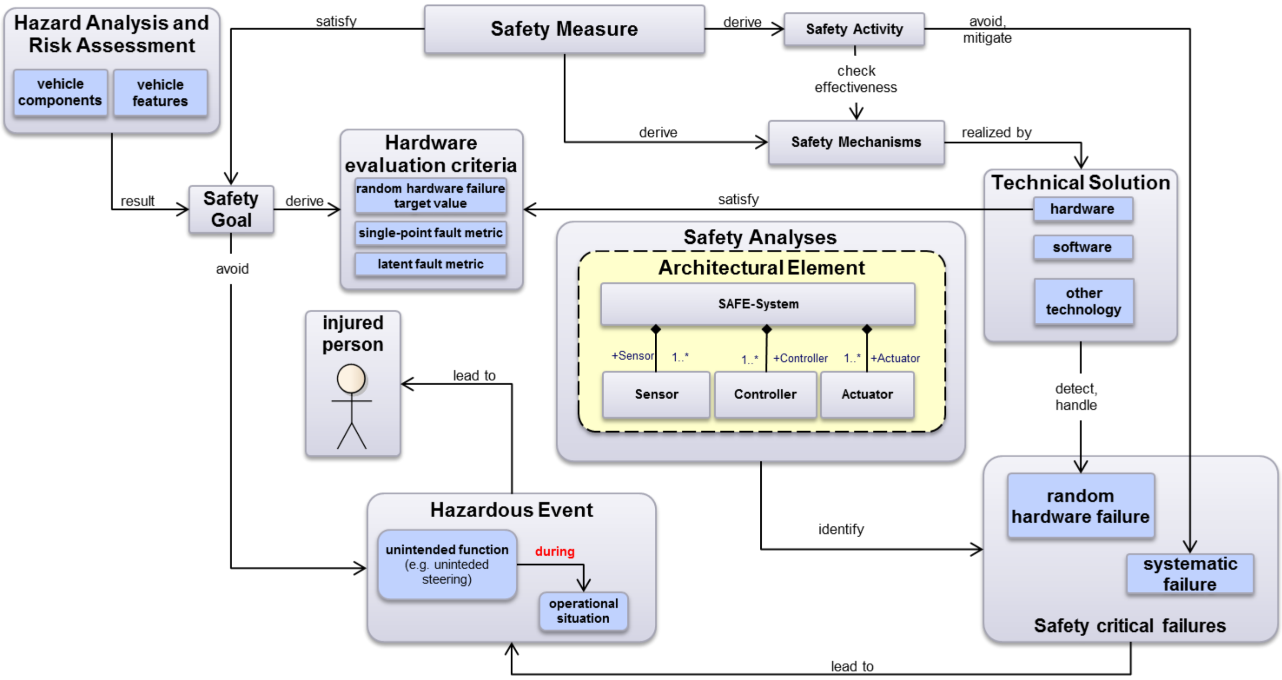


Figure 30: Safety analysis as central topic during system development

In the case that the safety analyses, executed during product development at the system level, identifies safety relevant failures that are able to cause new hazards, that are not already regarded during Hazard Analysis and Risk Assessment, the new hazard shall be analyzed by an impact analysis to identify the impact to the already existing safety relevant work products.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

Safety relevant failures identified during the safety analyses shall be part of the error model.

The following methods are defined according to the ASIL allocated to the safety goals allocated to the item:

Methods		ASIL			
		A	B	C	D
1	Deductive analysis[a]	o	+	++	++
2	Inductive analysis[b]	++	++	++	++

a Deductive analysis methods include FTA, reliability block diagrams, Ishikawa diagram.

b Inductive analysis methods include FMEA, ETA, Markov modelling.

Table 1: Methods to avoid systematic failures during product development at the system level

Further details according to these methods see ISO 26262 part 4 Table 1 and ISO 26262 part 9 chapter 8.

5.5.2 Criteria for coexistence of elements

In the case that the system architecture of the item contains sub-elements that have different ASIL the ISO 26262 specifies criteria for coexistence of elements.

Internal and external interfaces of each safety-related architectural element shall be defined to avoid safety-related effects of other elements.

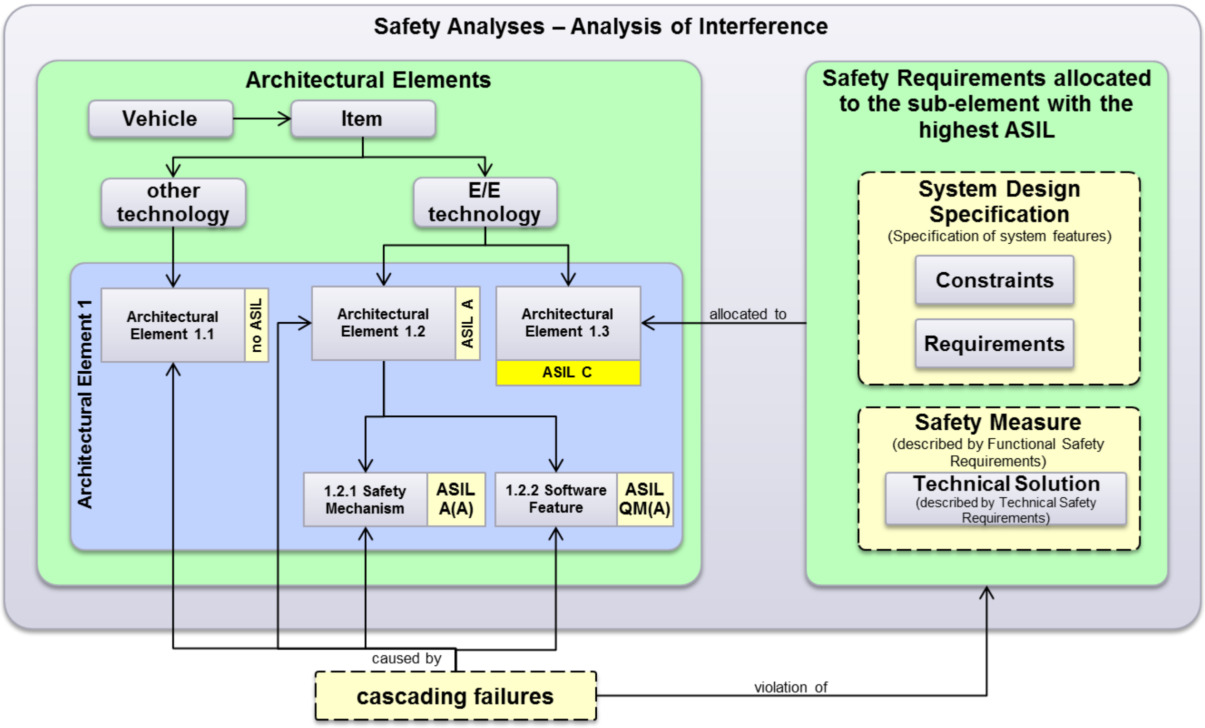

Safety Analyses – Analysis of Interference
Architectural Elements
Vehicle
Item
other technology
E/E technology
Architectural Element 1
Architectural Element 1.1
no ASIL
Architectural Element 1.2
ASIL A
Architectural Element 1.3
ASIL C
1.2.1 Safety Mechanism
ASIL A(A)
1.2.2 Software Feature
ASIL QM(A)
allocated to
Safety Requirements allocated to the sub-element with the highest ASIL
System Design Specification
(Specification of system features)
Constraints
Requirements
Safety Measure
(described by Functional Safety Requirements)
Technical Solution
(described by Technical Safety Requirements)
caused by
cascading failures
violation of

Figure 31: Item with sub-elements that have different ASIL

Analysis of interference of a sub-element with the other sub-element of architectural element 1 shall be executed.

Interference in the scope of ISO 26262 is defined as the presence of cascading failures from a sub-element with no ASIL or a lower ASIL is assigned to a sub-element with a higher ASIL that is able to violate a safety requirement that is allocated to one of the sub-elements of architectural element 1.

5.5.3 Impact Analysis

In case of any safety relevant modification or change an impact analysis shall be executed to identify the activities needed to implement the planned change without avoiding an already identified safety goal. The SAFE meta-model shall represent the results of the impact analysis.

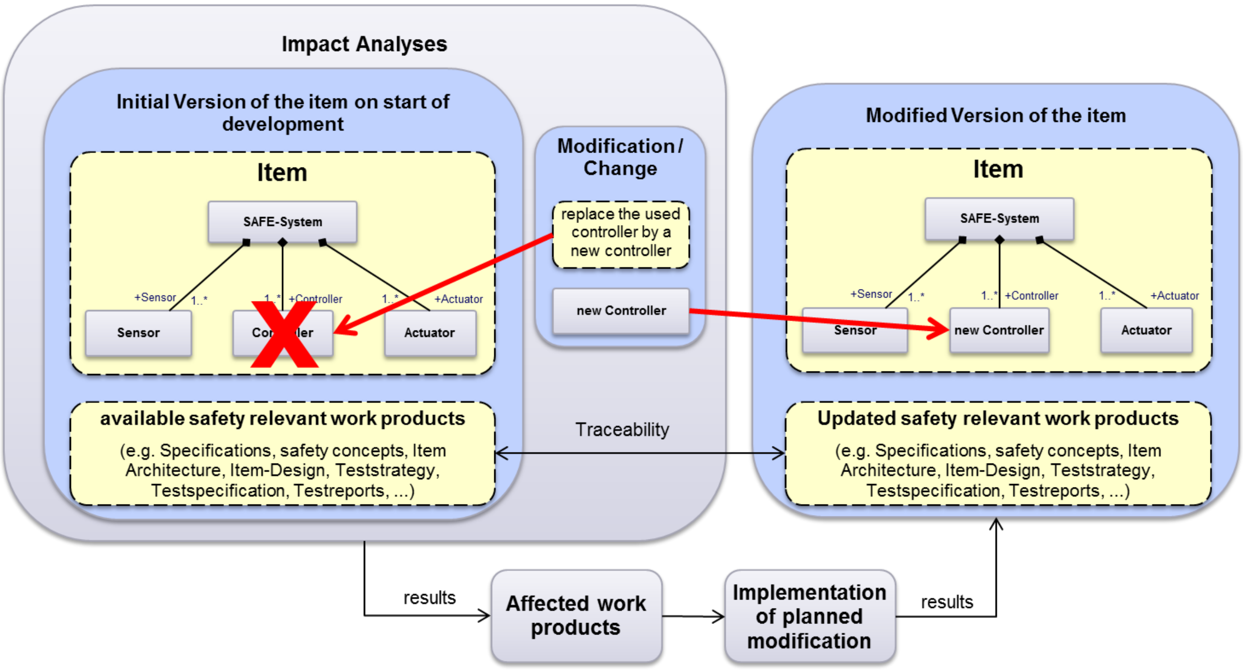

Impact Analyses
Initial Version of the item on start of development
Item
SAFE-System
+Sensor
1..*
1..*
+Controller
1..*
+Actuator
Sensor
Controller
Actuator
available safety relevant work products
(e.g. Specifications, safety concepts, Item Architecture, Item-Design, Teststrategy, Testspecification, Testreports, ...)
Modification / Change
replace the used controller by a new controller
new Controller
Traceability
Modified Version of the item
Item
SAFE-System
+Sensor
1..*
1..*
+Controller
1..*
+Actuator
Sensor
new Controller
Actuator
Updated safety relevant work products
(e.g. Specifications, safety concepts, Item Architecture, Item-Design, Teststrategy, Testspecification, Testreports, ...)
results
Affected work products
Implementation of planned modification
results

Figure 32: Impact Analysis

The impact analysis shall identify the

- architectural elements affected by the change request/modification
- already planned process activities affected by the change request/modification
- environmental interfaces already specified for the item affected by the change/modification
- affected item features affected by the change/modification
- effect of the modification regarding functional safety
- missing work products of the already existing item

The identified changes needed to implement the planned modification/change provided as a result of the impact analysis shall be stored and described in an effective manner. This should be realized based on the organization-specific supporting processes

- Requirement Management
- Change Management
- Configuration Management

Further details according to supporting processes see chapter 8.1.3

5.5.4 System Failure Propagation

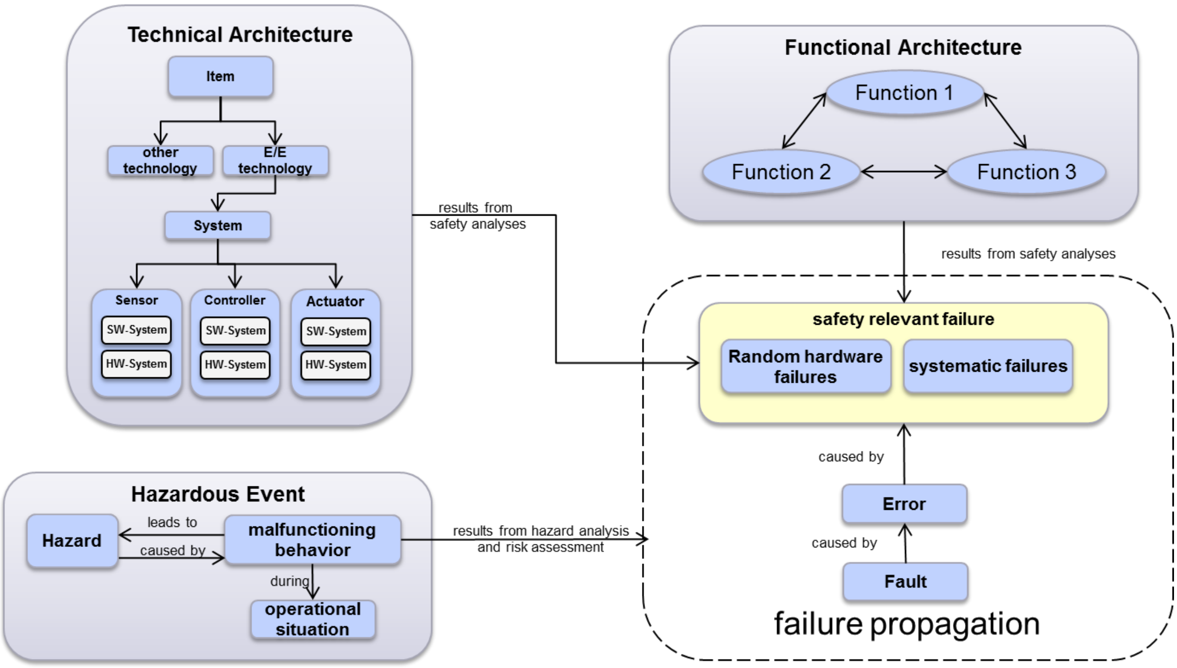


Figure 33: Failure Propagation on system level

Safety relevant failures that are given as results from the safety analysis shall be represented in the SAFE meta-model.

These failures on system-level are described in the ISO 26262 as malfunctioning behavior. SAFE meta-model shall contain a view for failure propagation. It shall be possible to create an error model for all identified safety relevant errors that are able to cause a safety relevant failure. The fault models used to analyze the safety relevant failures shall be consistent to

- hardware design
- evaluation of the hardware architectural metrics
- evaluation of safety goal violations due to random hardware failures

Further details according to failure propagation see D3.3.1.b [11]

5.6 Safety Validation

During safety validation evidence shall be provided that

- the planned external measures are implemented as specified in the safety requirement documentation
- the technical solution satisfies the allocated safety goals

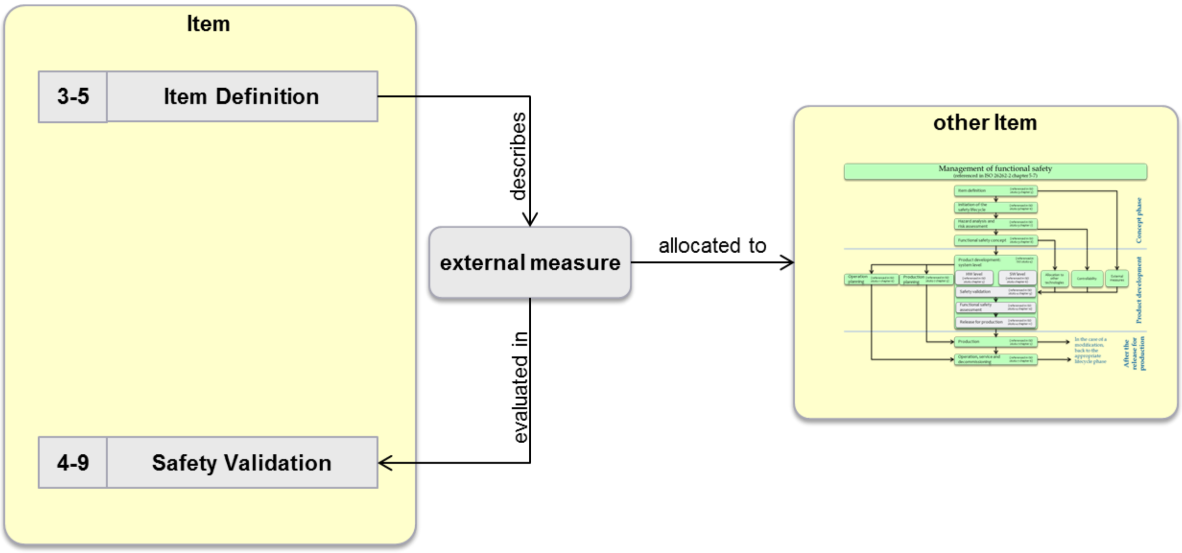


Figure 34: Safety Validation

Further details according to safety validation see D3.7.a [15] provided in SAFE-E.

6 Software Package Specification

This chapter contains the specification of elements that are needed to cover a safety architecture on software level according to ISO 26262 part 6. Elements that are needed in addition to the already existing artifacts of EAST-ADL and AUTOSAR are allocated to the software package of the SAFE meta-model.

6.1 Software Level

The following figure is showing the interface between System- and Software-Package specified for the SAFE meta-model.

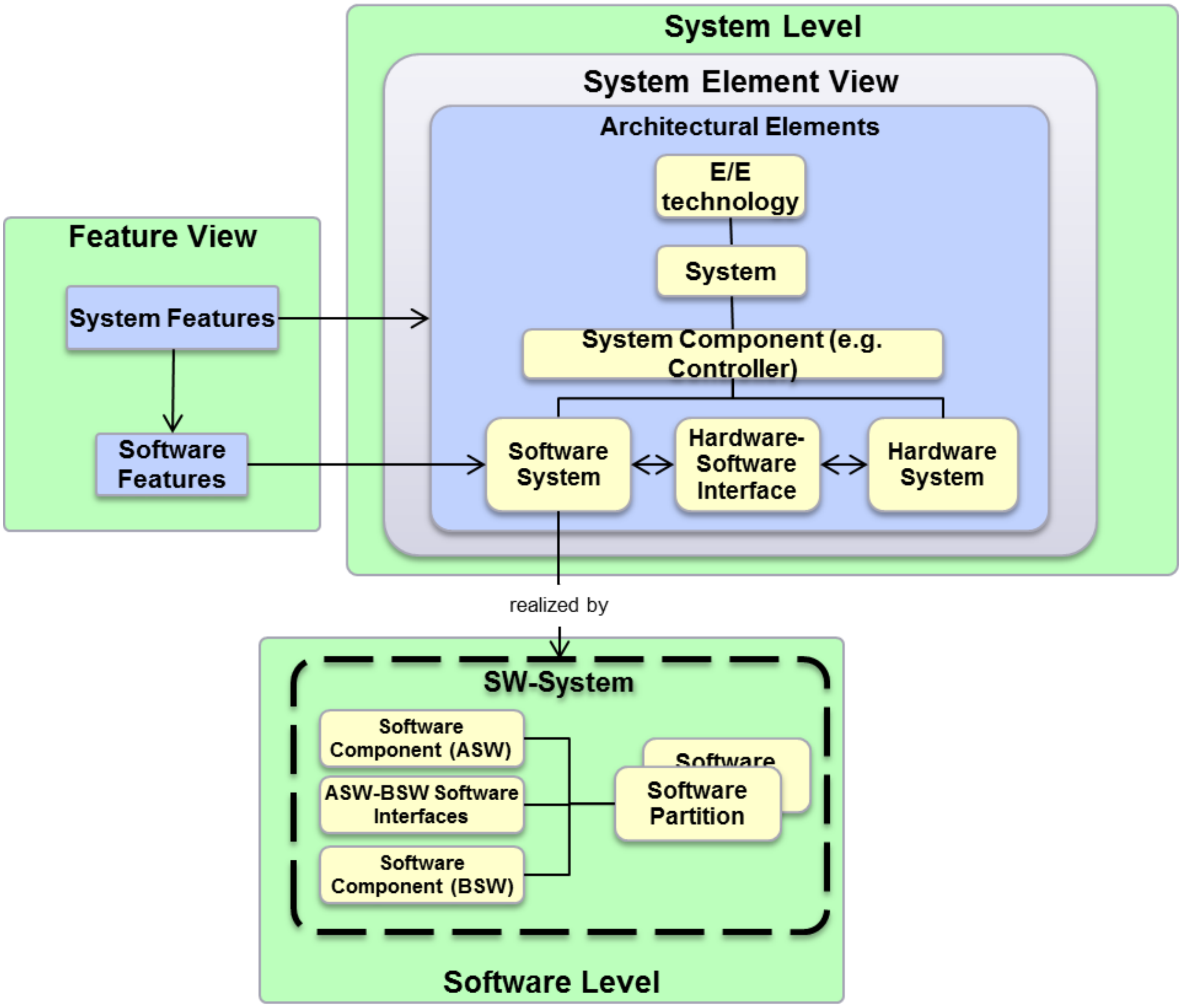


Figure 35: Interface System-Level <-> Software Level

A safety relevant software system shall contain at least the following elements:

- Software Partitions
- Software Components
- Software Units

The software level specifies

- the architectural splitting up of each software system to software components and software units.
- the architectural description of the interfaces between the software units
- safety mechanisms realized by software features to handle random hardware failures caused by hardware components/-parts used to realize safety relevant software features.
- safety measures realized by safety activities to handle systematic failures caused by development team members

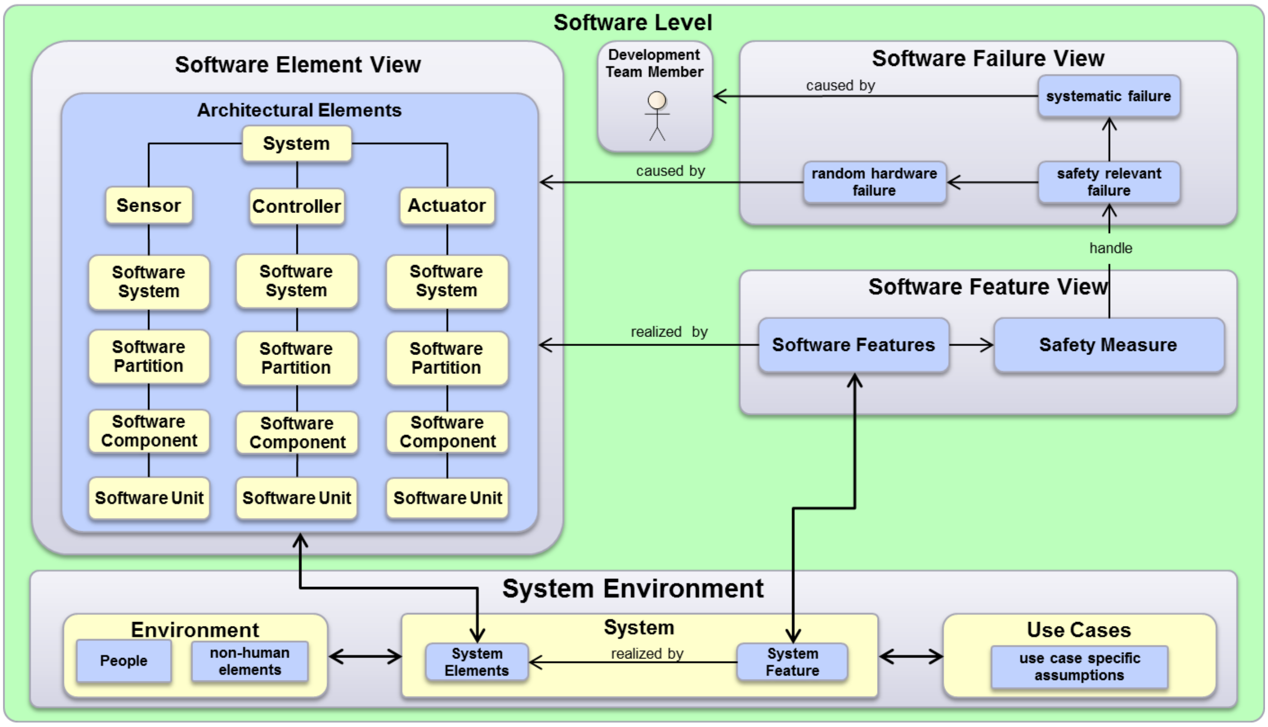


Figure 36: Software Level

6.1.1 Software Views

The software level shall contain different views:

- Software Element View
- Software Failure View
- Software Feature View

6.1.1.1 Software Element View

The software element view shall contain all architectural elements that are used to realize the identified safety relevant software features.

The software element view shall contain the interfaces between the architectural elements of the software.

The software element view shall contain the allocation between the architectural elements of the software and software features.

The software element view shall contain the interfaces between the software component and the hardware components that are used to realize the software functions.

The software element view shall contain the interfaces between the software component and its environment.

6.1.1.2 Software Features View

The software feature view shall contain all identified safety relevant features of the software.

The software feature view shall contain interfaces between the safety relevant features of the software.

The software feature view shall contain the allocation between safety relevant software features and the architectural elements used to realize the safety relevant software features.

The software feature view shall contain the interfaces between the safety relevant system features and item features.

6.1.1.3 Software Failure View

The software failure view shall contain all identified safety relevant failures that are caused by

- architectural elements of the software or
- development team members.

Each safety relevant failure shall contain the allocation to the architectural element that has caused the safety relevant failure.

The safety relevant failure shall be identified during safety analyses on software level.

6.2 Software Safety Requirement Specification

The software safety requirements shall be part of the software safety requirement specification that is derived from the Technical Safety Concept. The software safety requirements describe the software safety mechanism realized by architectural elements on software level defined as add on to the technical solution to fulfill the functional safety requirements referred in the technical safety concept.

The software safety requirement specification is defined as an input document for starting the safety relevant software development.

6.3 Software Architecture and Design

Based on the requirements given in the ISO 26262 part 6 the following generic safety relevant architectural elements on software level are defined.

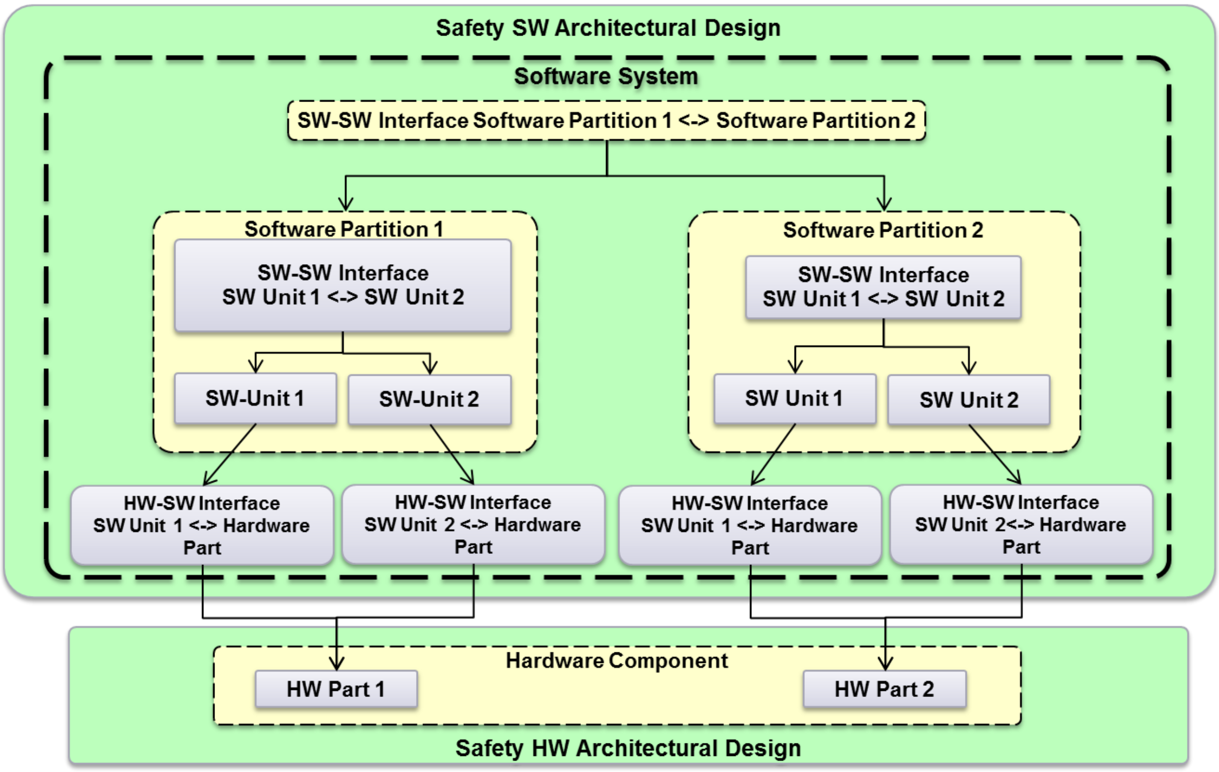

Safety SW Architectural Design
Software System
SW-SW Interface Software Partition 1 <-> Software Partition 2
Software Partition 1
SW-SW Interface SW Unit 1 <-> SW Unit 2
SW-Unit 1
SW-Unit 2
Software Partition 2
SW-SW Interface SW Unit 1 <-> SW Unit 2
SW Unit 1
SW Unit 2
HW-SW Interface SW Unit 1 <-> Hardware Part
HW-SW Interface SW Unit 2 <-> Hardware Part
HW-SW Interface SW Unit 1 <-> Hardware Part
HW-SW Interface SW Unit 2<-> Hardware Part
Hardware Component
HW Part 1
HW Part 2
Safety HW Architectural Design

Figure 37: Safety relevant SW-Architecture Elements

During safety relevant software development several safety activities shall be planned to avoid/mitigate systematic failures during software development. Further details according to this safety activity see D3.7.a [15].

Each software component used in the software architectural design shall be categorized with the following categories:

- newly developed
- reused with modifications
- reused without modifications

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

6.3.1 Software-Partitions

A safety relevant SW-System Architecture shall contain an overview about the SW-Partitions planned to realize the software safety requirements.

In the case that more than one SW-Partition is used, the SW-Partitions shall be designed sufficiently independent to ensure that a SW-Feature realized in one SW-Partition is not able to violate a software safety requirement allocated to any other SW-Partition.

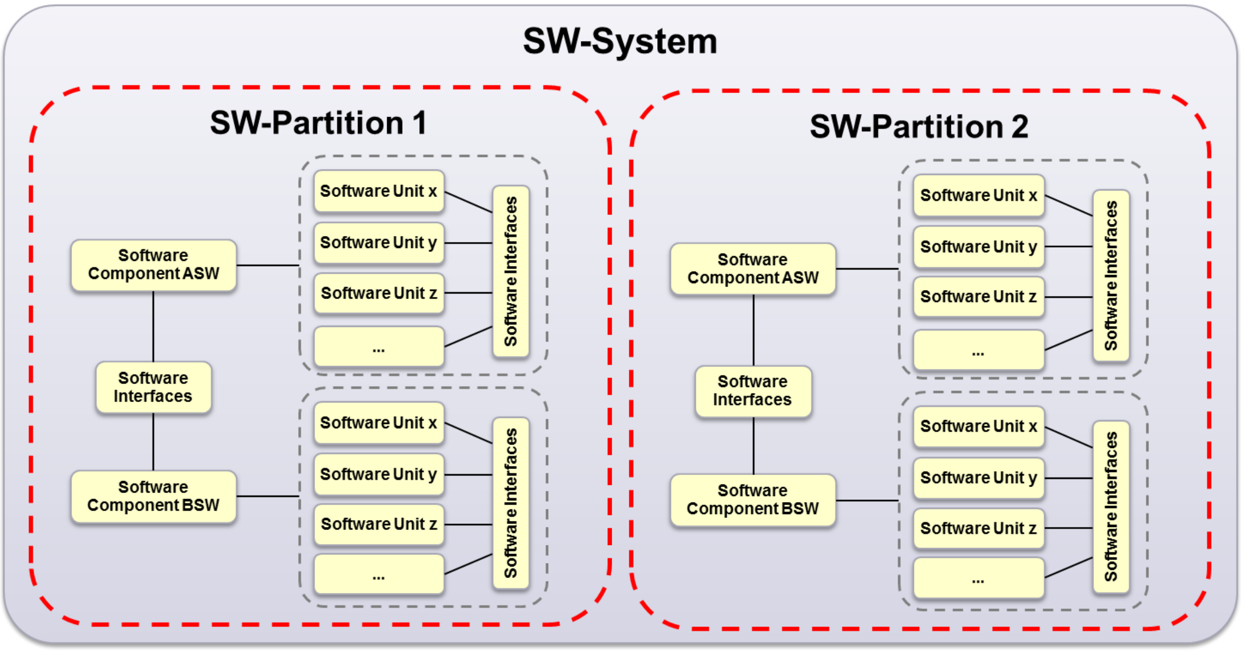


Figure 38: Safety relevant SW-Partitions

6.3.2 Software Safety Mechanisms

Software safety mechanisms shall be allocated and/or specified as add-on to the already existing architectural elements on software level. They shall be specified by software safety requirements derived by technical safety requirements.

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

6.4 Integration of AUTOSAR-Elements to the SW-Architecture of SAFE-Meta-Model

The following figure is showing the methodology to integrate an already existing AUTOSAR-element to safety-relevant software architecture.

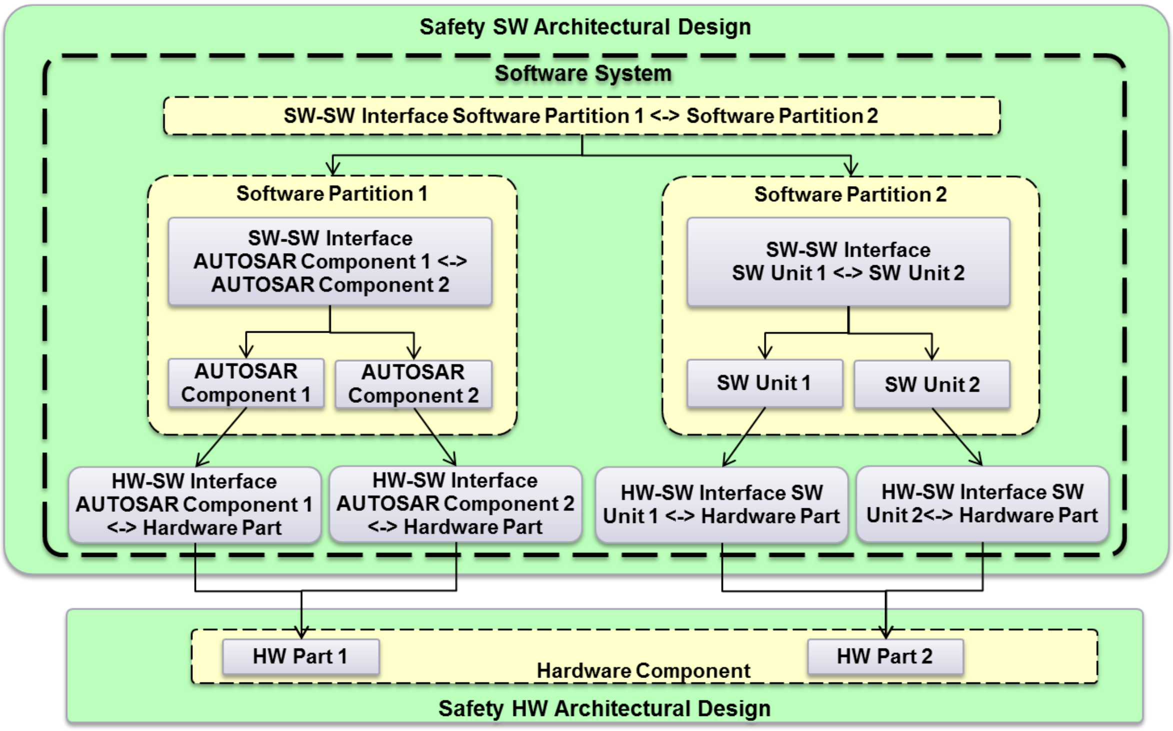

Safety SW Architectural Design
Software System
SW-SW Interface Software Partition 1 <-> Software Partition 2
Software Partition 1
SW-SW Interface AUTOSAR Component 1 <-> AUTOSAR Component 2
AUTOSAR Component 1
AUTOSAR Component 2
Software Partition 2
SW-SW Interface SW Unit 1 <-> SW Unit 2
SW Unit 1
SW Unit 2
HW-SW Interface AUTOSAR Component 1 <-> Hardware Part
HW-SW Interface AUTOSAR Component 2 <-> Hardware Part
HW-SW Interface SW Unit 1 <-> Hardware Part
HW-SW Interface SW Unit 2<-> Hardware Part
HW Part 1
Hardware Component
HW Part 2
Safety HW Architectural Design

Figure 39: SW-Architecture and Design

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

FFG

6.4.1 AUTOSAR - Architecture

Therefore the following figure is showing the Technical Overview of AUTOSAR:

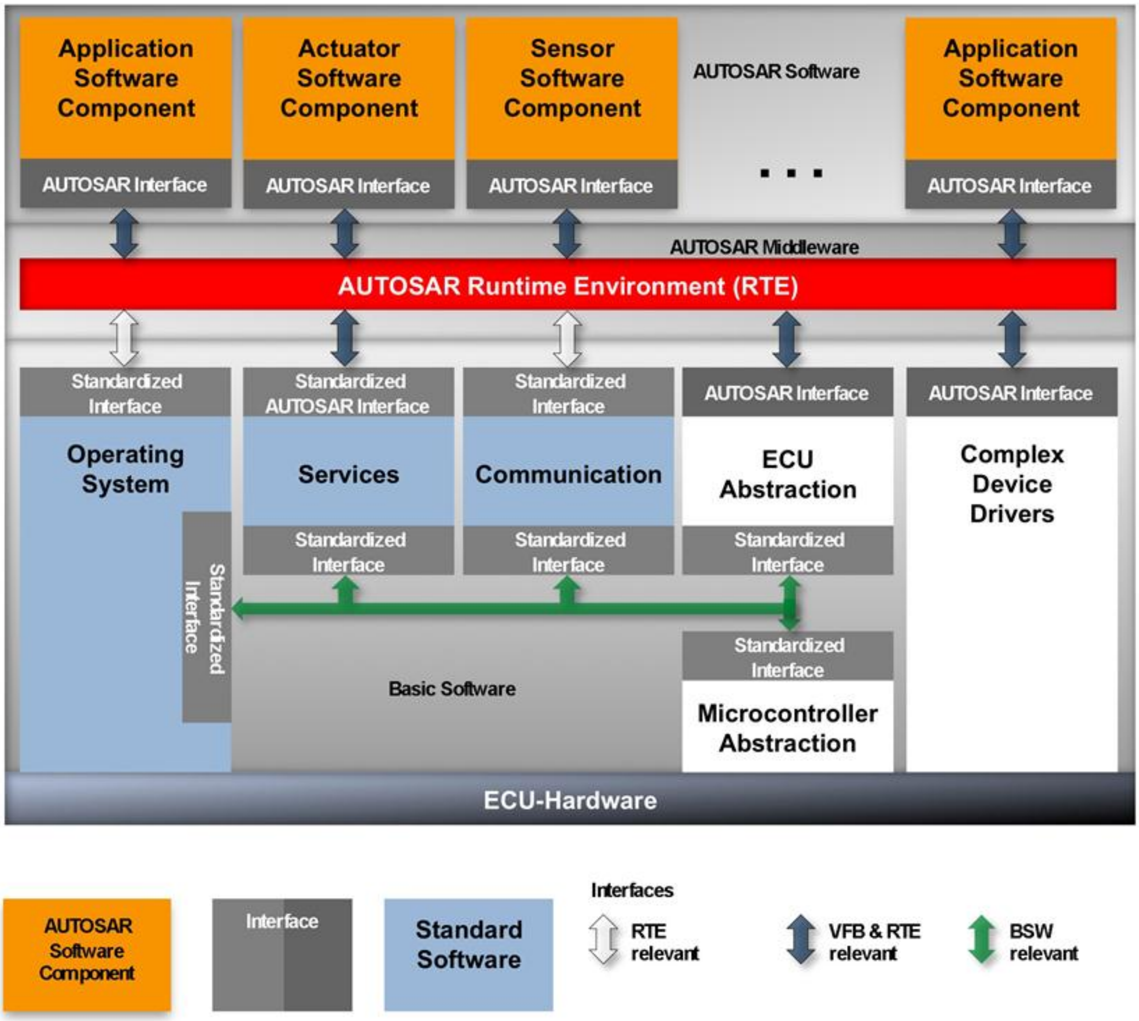


Figure 40: AUTOSAR - Technical Overview[1]

[1] www.autosar.org

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

6.4.2 Evaluation of the AUTOSAR SW-Tools

During integration of AUTOSAR-SW-Component different software tools have to be used, like Component Generator, AUTOSAR system configuration generator … To provide evidence that the software tool is suitable for the planned activities a validation of the used software tools shall be executed according to ISO 26262 part 8 chapter 11. The validation of used software tools is one of the safety activities described in the chapter safety relevant supporting processes (see chapter 8.1.3).

6.4.3 Qualification of the AUTOSAR SW-Component

In addition to that the integrated AUTOSAR SW-component shall be qualified to provide evidence for their suitability for re-use in the defined item under development according to ISO 26262 part 8 chapter 12. The qualification of software components is one of the safety activities described in the chapter safety relevant supporting processes (see chapter 8.1.3).

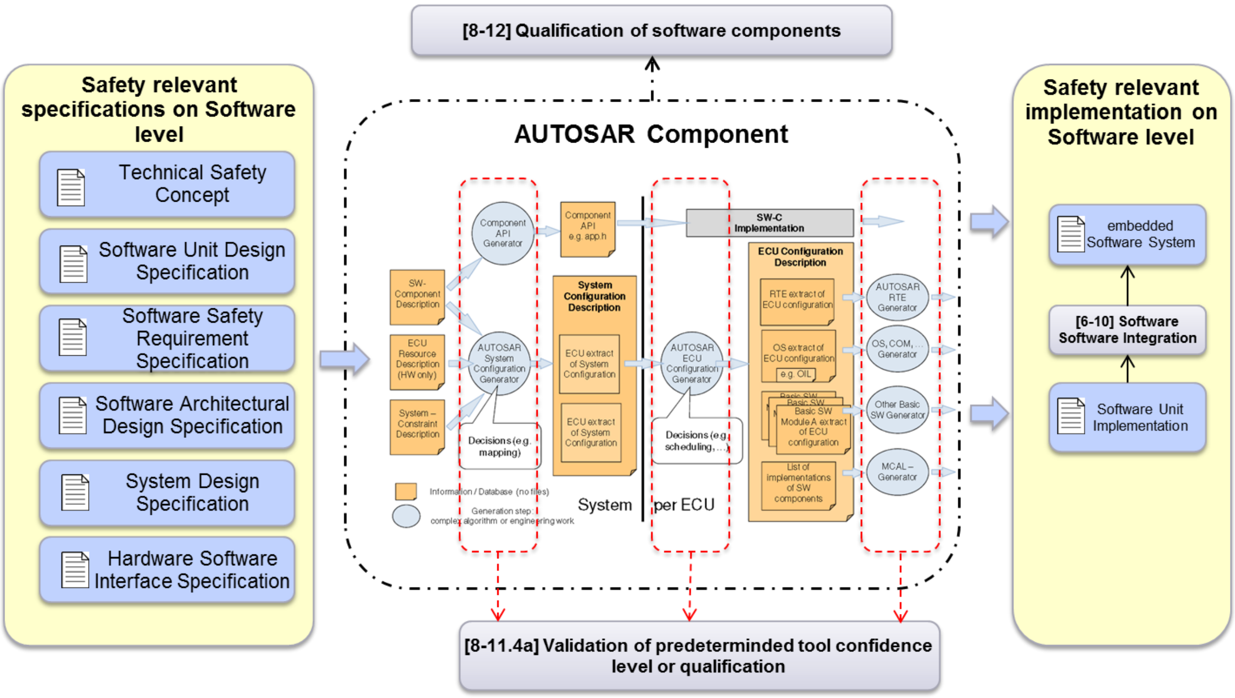

[8-12] Qualification of software components
Safety relevant specifications on Software level
Technical Safety Concept
Software Unit Design Specification
Software Safety Requirement Specification
Software Architectural Design Specification
System Design Specification
Hardware Software Interface Specification
AUTOSAR Component
Component API Generator
Component API e.g. app.h
SW-C Implementation
SW-Component Description
ECU Resource Description (HW only)
System – Constraint Description
AUTOSAR System Configuration Generator
Decisions (e.g. mapping)
System Configuration Description
ECU extract of System Configuration
ECU extract of System Configuration
AUTOSAR ECU Configuration Generator
Decisions (e.g. scheduling, …)
ECU Configuration Description
RTE extract of ECU configuration
OS extract of ECU configuration
e.g. OIL
Basic SW Module A extract of ECU configuration
List of implementations of SW components
AUTOSAR RTE Generator
OS, COM, … Generator
Other Basic SW Generator
MCAL – Generator
Information / Database (no files)
Generation step: complex algorithm or engineering work
System
per ECU
Safety relevant implementation on Software level
embedded Software System
[6-10] Software Software Integration
Software Unit Implementation
[8-11.4a] Validation of predetermninded tool confidence level or qualification

Figure 41: Reuse of AUTOSAR-Components

After successful qualification of the AUTOSAR SW-Component a Software-Software Integration shall be executed based on the requirements given in ISO 26262 part 6 chapter 10.

6.5 Software Configuration

To enable controlled changes in the behavior of the software for different applications software configuration shall be applied according to ISO 26262 part 6 Annex C.

Configuration- and Calibration parameter shall be specified by Software safety requirements.

The SAFE meta-model shall provide the needed artifacts to cover configuration and calibration parameter. The traceability to the software safety requirements shall be ensured.

Verification and Validation activities shall be defined as safety activities to ensure

- the use of values within their specified range; and
- the compatibility with the other configuration data

Further details according to safety activities see chapter 8.1

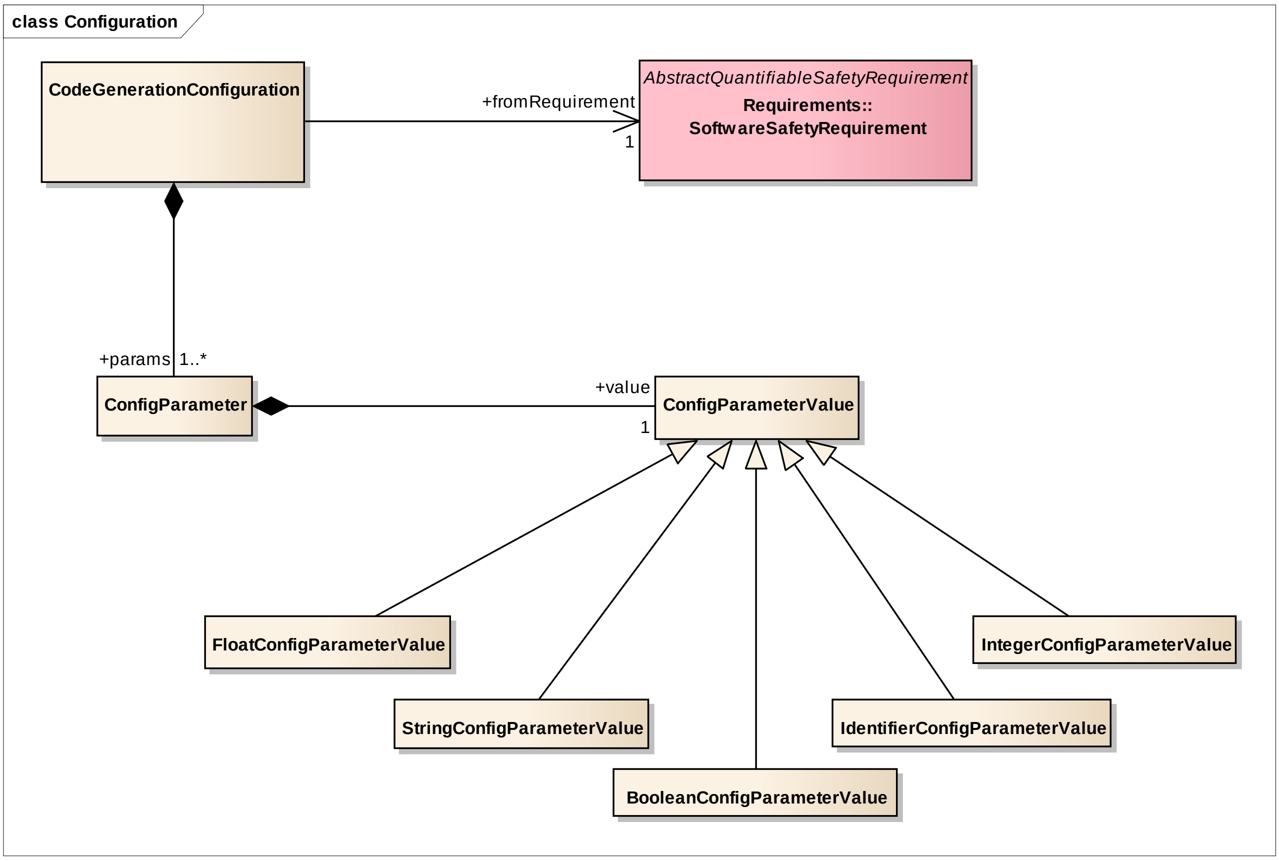

class Configuration
CodeGenerationConfiguration
+fromRequirement
1
AbstractQuantifiableSafetyRequirement
Requirements::
SoftwareSafetyRequirement
+params
1..*
ConfigParameter
+value
1
ConfigParameterValue
FloatConfigParameterValue
IntegerConfigParameterValue
StringConfigParameterValue
IdentifierConfigParameterValue
BooleanConfigParameterValue

Figure 42: SAFE meta-model software configuration

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

6.6 Software Verification Activities

During development of safety relevant software systems the following safety activities shall be planned to verify the software:

- Software Unit Testing
- Software Integration and Testing
- Verification of Software Safety Requirements
- Verification of Configuration Data
- Verification of Calibration Data

Further details according to safety activities see chapter 8.1.2

6.6.1 Freedom from Interference

SAFE meta-model shall contain a field to classify the independence of two elements. It shall be possible to mark elements that are free from interference. The evidence of freedom from interference shall be documented in the SAFE meta-model.

SAFE meta-model shall allow defining software partitions, which guarantee freedom from interference for the software components allocated to different software partitions.

7 Implementation of the SAFE meta-model

The SAFE meta-model is defined to create a traceable view of a safety relevant item in the meaning of ISO 26262. The starting point of this meta-model is the definition of the item which is the element under development called SAFE-Object.

The SAFE-Object shall contain all SAFE-Elements that are able to influence the safety relevant behavior of the item under development.

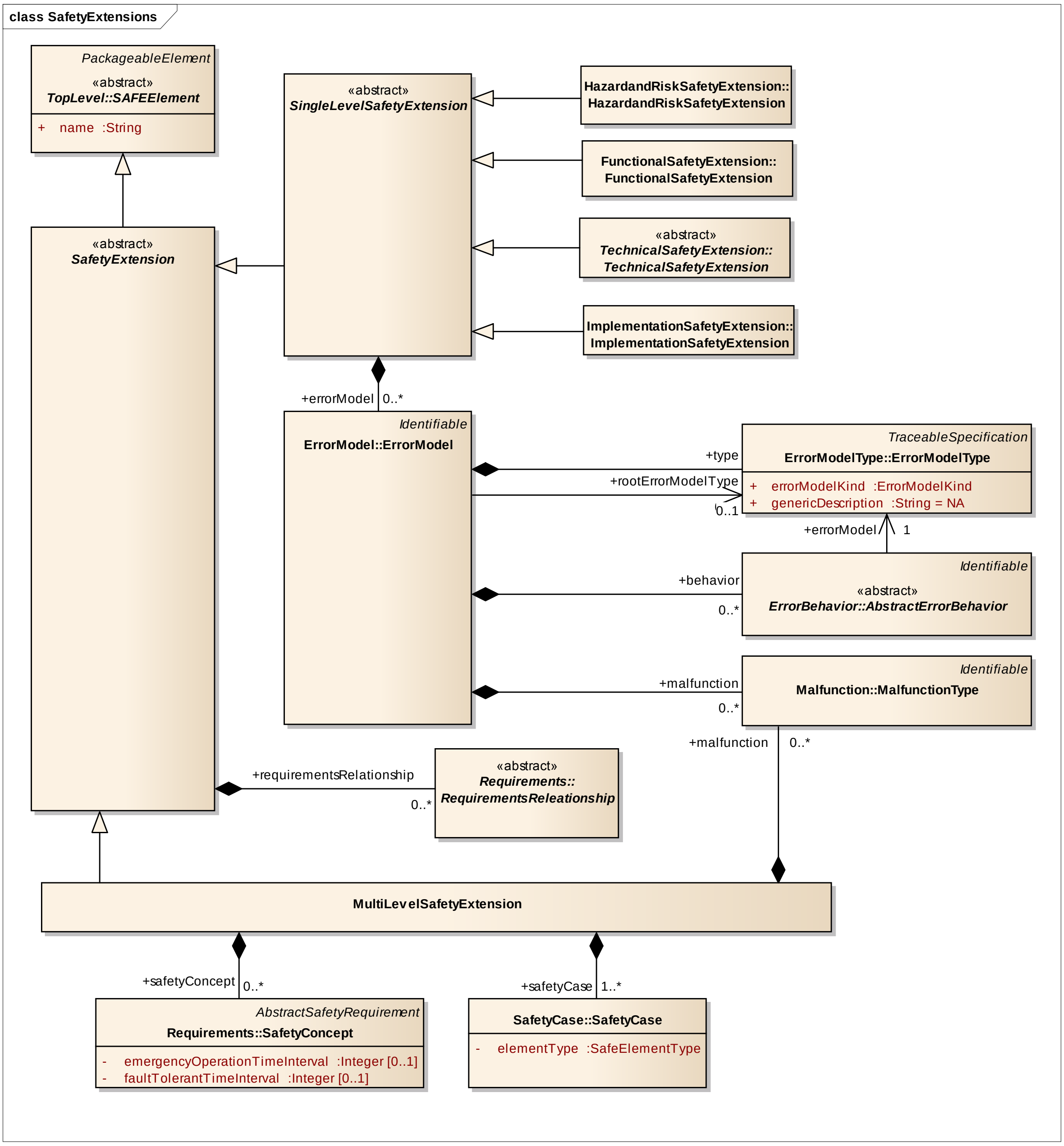

class SafetyExtensions
PackageableElement
«abstract»
TopLevel::SAFEElement
+ name :String
«abstract»
SafetyExtension
«abstract»
SingleLevelSafetyExtension
HazardandRiskSafetyExtension::
HazardandRiskSafetyExtension
FunctionalSafetyExtension::
FunctionalSafetyExtension
«abstract»
TechnicalSafetyExtension::
TechnicalSafetyExtension
ImplementationSafetyExtension::
ImplementationSafetyExtension
+errorModel 0..*
Identifiable
ErrorModel::ErrorModel
+type
+rootErrorModelType
0..1
TraceableSpecification
ErrorModelType::ErrorModelType
+ errorModelKind :ErrorModelKind
+ genericDescription :String = NA
+errorModel 1
+behavior
0..*
Identifiable
«abstract»
ErrorBehavior::AbstractErrorBehavior
+malfunction
0..*
Identifiable
Malfunction::MalfunctionType
+malfunction 0..*
+requirementsRelationship
0..*
«abstract»
Requirements::
RequirementsReleationship
MultiLevelSafetyExtension
+safetyConcept 0..*
+safetyCase 1..*
AbstractSafetyRequirement
Requirements::SafetyConcept
- emergencyOperationTimeInterval :Integer [0..1]
- faultTolerantTimeInterval :Integer [0..1]
SafetyCase::SafetyCase
- elementType :SafeElementType

Figure 43: SAFE meta-model safety extensions

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

7.1 Description of the SAFE meta-model

The description of all the artifacts contained in the SAFE meta-model is part of D3.5.c [12]

This document contains a detailed explanation of the artifacts used in the SAFE meta-model.

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

8 Further Topics

This chapter contains topics that are not covered by this document in detail.

8.1 Safety Activities within the development of safety relevant products

The ISO 26262 specifies safety activities on different development levels to avoid or mitigate systematic failures during product development. These safety activities are specified as a proposal for a process model in the documents D3.7.a [15], D3.7.b [16] and D3.7.c [17] provided as work products of the SAFE-E project. These documents were also used as input for the Methodology and application rules described in D6.b [13].

8.1.1 Safety Activity on system level

The proposed process model contains the following safety activities on system level:

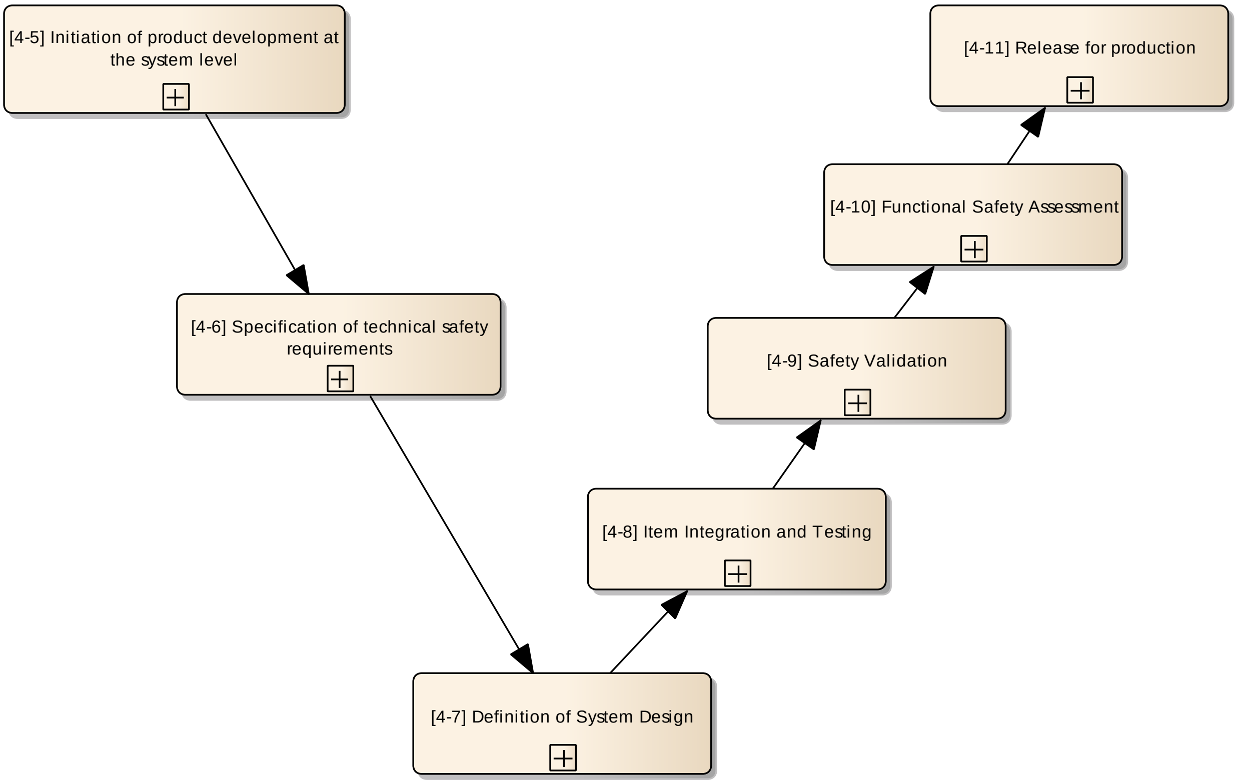

[4-5] Initiation of product development at the system level
[4-6] Specification of technical safety requirements
[4-7] Definition of System Design
[4-8] Item Integration and Testing
[4-9] Safety Validation
[4-10] Functional Safety Assessment
[4-11] Release for production

Figure 44: Safety Activities during product development on system level

Further details regarding Safety Activities on System level see D6.b [13].

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

8.1.2 Safety Activity on software level

The proposed process model contains the following safety activities on software level:

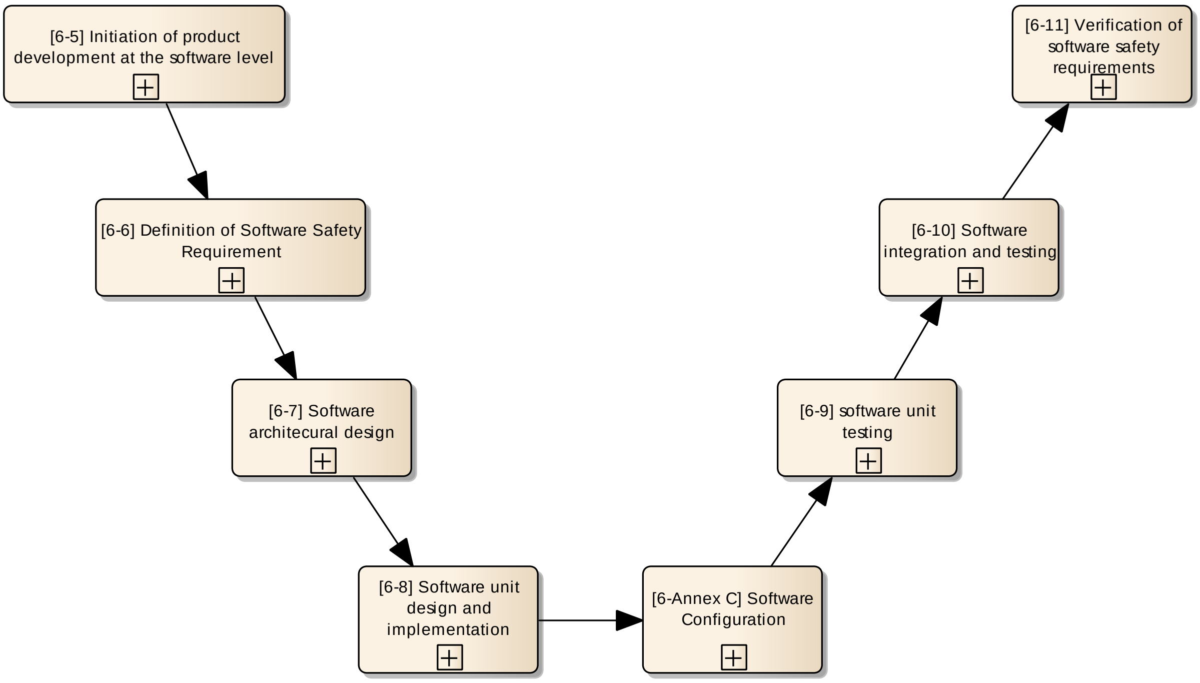

[6-5] Initiation of product development at the software level
[6-6] Definition of Software Safety Requirement
[6-7] Software architecural design
[6-8] Software unit design and implementation
[6-Annex C] Software Configuration
[6-9] software unit testing
[6-10] Software integration and testing
[6-11] Verification of software safety requirements

Figure 45 Safety Activities during product development on software level

In addition to that the following chapters give some examples of safety relevant supporting activities that shall be also part of the SAFE-E process model.

8.1.3 Safety relevant supporting process

The proposed process model shall contain the following safety relevant supporting processes

8.1.3.1 [8-5] Interfaces within distributes development

During this safety activity the procedures within distributed developments for items and elements shall be described in the Development Interface Agreement. The associated responsibilities shall be associated to the described procedures.

Further details see ISO 26262 part 8 chapter 5.

8.1.3.2 [8-6] Specification and management of safety requirements

This supporting process shall be applied to ensure

- the correct specification of safety requirements with respect to their attributes and characteristics.

- consistent management of safety requirements throughout the entire safety lifecycle.

Management of functional safety requirements means:

- obtaining agreement on the requirements
- obtaining commitments from those person(s) who are responsible for implementing the requirements
- maintaining traceability
- managing of requirements (usage of suitable requirements management tool)

Further details see ISO 26262 part 8 chapter 5.

8.1.3.3 [8-7] Configuration Management

This supporting process shall be applied to ensure

- that the work products, and the principles and general conditions of their creation, can be uniquely identified and reproduced in a controlled manner at any time.
- that the relations and differences between earlier and current versions can be traced.

Based on the two different use cases of the item development (modification / new development) the input on start of development can contain different maturity levels.

In the case of a new development the input described in the item description contains a product idea.

In the case of modification the input described in the item description contains already validated requirements of the item that is planned to be modified.

In both cases the used input shall be validated for the new application.

Therefore the initial maturity level of a requirement or architectural element is defined as preliminary.

Depending on the organization specific development process further maturity levels shall be defined to show the maturity level of each safety relevant work product during the development lifecycle. The following figure is showing an example of maturity levels for a safety relevant work product.

The Eurostars Programme is powered by EUREKA and the European Community

EUREKA

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

FFG

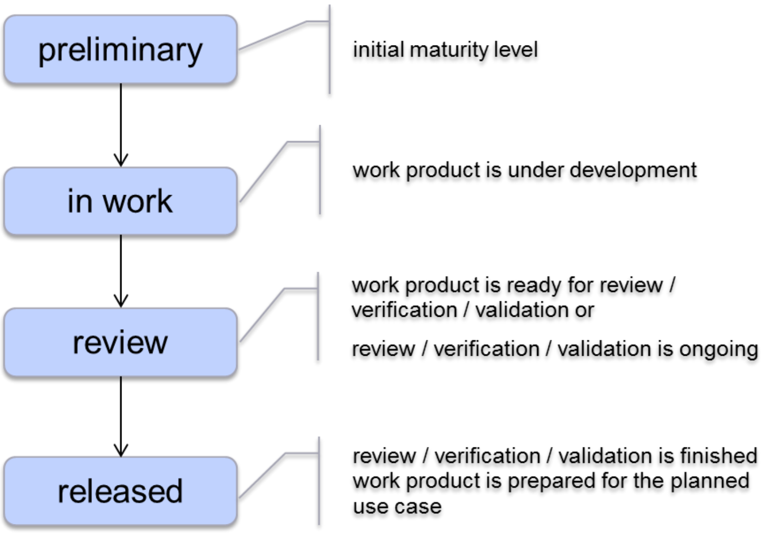

preliminary
initial maturity level
in work
work product is under development
review
work product is ready for review / verification / validation or
review / verification / validation is ongoing
released
review / verification / validation is finished
work product is prepared for the planned use case

Figure 46: Maturity of safety relevant work products

This maturity levels can be used for each safety relevant work product and also for the safety requirements.

To show the maturity of an architectural element the ISO 26262 differs between preliminary architectural elements and architectural elements. Architectural Elements shall be marked as preliminary if they are not finally verified or validated.

Further details see ISO 26262 part 8 chapter 7.

8.1.3.4 [8-8] Change Management

Change management shall be applied to analyze and control changes to safety-related work products throughout the safety lifecycle.

Example:

If new hazard are identified during safety analyses a change request shall be created to trigger an update of the hazard analysis and risk assessment according to the new identified hazard.

Further details see ISO 26262 part 8 chapter 8.

8.1.3.5 [8-9] Verification

This supporting process shall be applied to ensure that the work products comply with their requirements.

Verification of Technical Safety Requirements shall be done by appropriate analysis.

Technical Safety Requirements shall be

- compliant to the requirements defined in the Functional Safety Concept
- consistent to the requirements defined in the Functional Safety Concept
- compliant to the preliminary architectural design
- consistent to the preliminary architectural design

Further details see ISO 26262 part 8 chapter 9.

The Eurostars Programme is powered by EUREKA and the European Community
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

8.1.3.6 [8-10] Documentation

This supporting process shall be applied to develop a documentation management strategy for the entire safety lifecycle in order to facilitate an effective and repeatable documentation management process in accordance with ISO 26262 part 8 chapter 10.

8.1.3.6.1 Documentation Management

The actual specification contains parts of the safety requirement documentation:

- Functional Safety Concept (see chapter 5.1.3)
- Technical Safety Concept (see chapter 5.4.5.3)
- System Design (see chapter 5.4)

These three documents are defined as safety relevant work products in the ISO 26262. They shall contain safety requirements of the item. But there are further requirement documents needed to describe the safety relevant item throughout all its levels:

- Hardware Software Interface Specification
- Hardware Safety Requirement Specification
- Software Safety Requirement Specification.

These requirement documents are also defined in ISO 26262 as safety relevant work products. The next improvement step of the SAFE meta-model shall provide a solution for modeling these three documents.

8.1.3.6.2 Consistency of safety requirement documents

The safety requirements that are specified in the safety requirement documents shall be consistent throughout the entire safety lifecycle. A consistency check of the safety requirements is defined as a process activity

General characteristics of a safety requirement that are needed to execute a consistency check:

- unambiguous and comprehensible
- atomic
- internally consistent
- feasible
- verifiable
- unique identifier remaining unchanged during the entire safety lifecycle

8.1.3.6.3 Traceability

Traceability shall be provided between Safety goal, safe state and fault tolerant time interval on vehicle level and on item level.

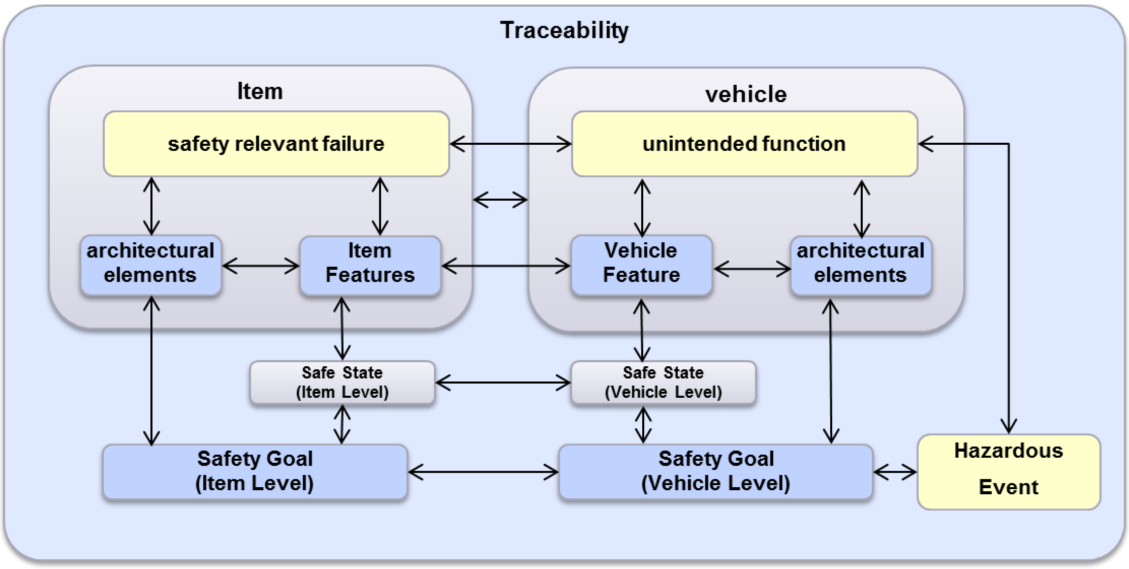


Figure 47: Traceability Vehicle <-> Item

Traceability is an essential topic for developing a safety related item without creating redundant information. The traceability concept shall allow an effective way of change/modification of the requirements defined for the item under development. In addition to that the consistency of the requirements shall be ensured throughout the entire safety lifecycle.

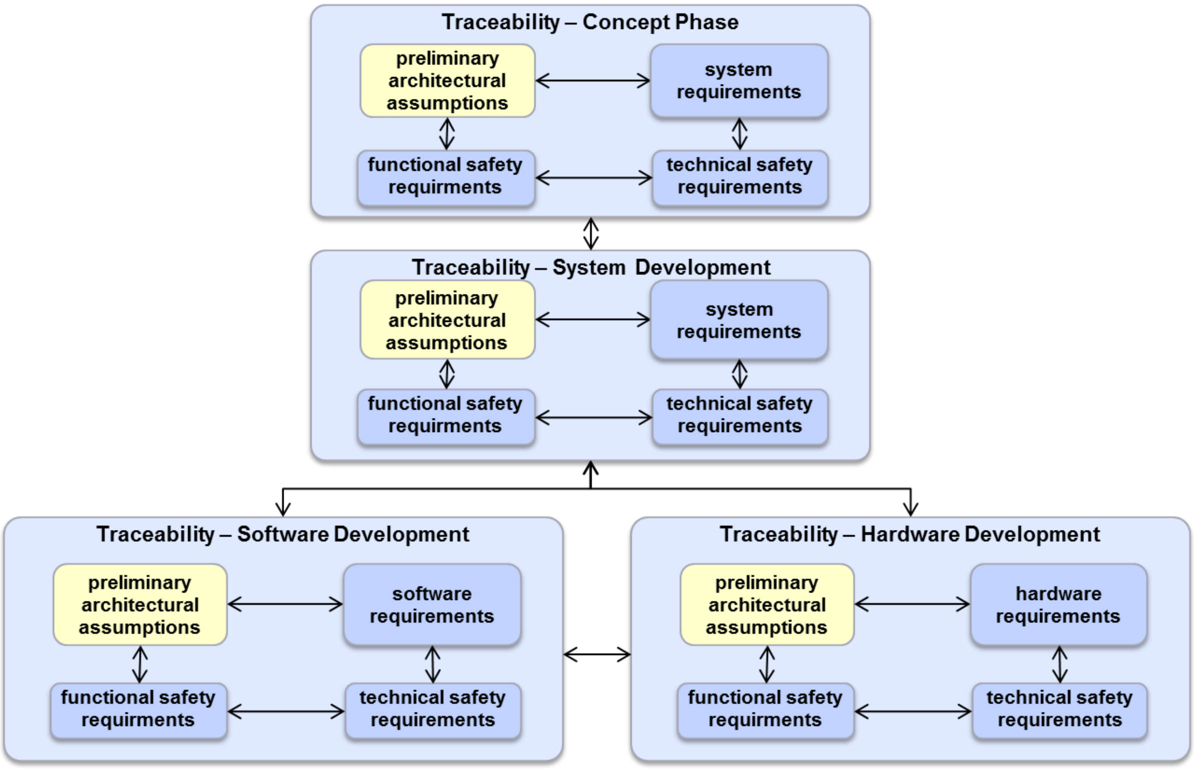

Traceability – Concept Phase
preliminary architectural assumptions
system requirements
functional safety requirments
technical safety requirements
Traceability – System Development
preliminary architectural assumptions
system requirements
functional safety requirments
technical safety requirements
Traceability – Software Development
preliminary architectural assumptions
software requirements
functional safety requirments
technical safety requirements
Traceability – Hardware Development
preliminary architectural assumptions
hardware requirements
functional safety requirments
technical safety requirements

Figure 48: Requirement Traceability

The Eurostars Programme is powered by EUREKA and the European Community
EUREKA
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
FFG

8.1.3.6.4 Architecture and structure of safety relevant product documentation

Based on the fact that in most cases more than one organization is involved to the development of a vehicle different documentation strategies shall be matched to one entire safety case of the vehicle. To ensure this a documentation management plan shall describe the different strategies that are used during vehicle development to ensure traceability of the safety relevant information throughout the entire safety documentation.

Depending on the complexity of the items and the structure of the development teams planned for vehicle development work products specified in the ISO 26262

- can be summarized to one document if reasonable.
- can be split up to different documents if reasonable.

The documentation management plan is a result of the safety related supporting processes.

Further details according to supporting processes see D3.1.3 [8].

8.1.3.7 [8-11] Confidence in the use of software tools

This supporting process shall be applied to provide

- criteria to determine the required level of confidence in a software tool when applicable.
- means for the qualification of the software tool when applicable, in order to create evidence that the software tool is suitable to be used to tailor the activities or tasks required by ISO 26262

Further details see ISO 26262 part 8 chapter 11

8.1.3.8 [8-12] Qualification of software components

This supporting process shall be applied to provide evidence for their suitability for re-use in items developed in compliance with ISO 26262.

The results of the qualification of software components shall be documented.

Interface description shall be provided for each software component that shall be reused.

Integration tests shall be done for each reused safety component. Integration test activities shall be defined as safety activities.

Further details see ISO 26262 part 8 chapter 12

8.1.3.9 [8-13] Qualification of hardware components

This supporting process shall be applied to to provide

- evidence of the suitability of intermediate level hardware components and parts for their use as part of items, systems or elements, developed in compliance with ISO 26262, concerning their functional behavior and their operational limitations for the purposes of the safety concept.
- relevant information regarding their failure modes,

- relevant information regarding their failure mode distribution, and
- relevant information regarding their diagnostic capability with respect to the safety concept for the item.

Further details see ISO 26262 part 8 chapter 13

8.1.3.10 [8-14] Proven in use argument

This supporting process shall provide guidance for a proven in use argument. A proven in use argument is an alternate means of compliance with ISO 26262 that may be used in the case of reuse of existing items or elements when field data is available.

Further details see ISO 26262 part 8 chapter 14

9 References

[1] International Organization for Standardization: ISO 26262 Road vehicles - Functional safety. (2011)

[2] ATTEST2-Project: ATTEST2-Partners, EAST-ADL Specification (www.east-adl.info)

[3] AUTOSAR 4.0 Specification (www.autosar.org)

[4] CESAR Project, http://www.cesarproject.eu/

[5] SAFE-Project: SAFE-Partners, D2.1.c: Needs description to apply ISO26262 with architecture and component modeling

[6] SAFE-Project: SAFE-Partners, D3.1.1.c: Initial proposal for extension of meta-model for hazard and environment modeling

[7] SAFE-Project: SAFE-Partners, D3.1.2.c: Proposal for extension of meta-model for safety requirement expression modeling

[8] SAFE-Project: SAFE Partners, D3.1.3: Proposal for extension of meta-model for safety case modeling

[9] SAFE-Project: SAFE Partners, D3.2.1.c Final proposal for extension of meta-model for software and system modeling

[10] SAFE-Project: SAFE Partners, D3.2.2.b: Proposal for extension of meta-model for hardware modeling

[11] SAFE-Project: SAFE Partners, D3.3.1.b: Proposal for extension of meta-model for error failure and propagation analysis

[12] SAFE-Project: SAFE-Partners, D3.5.c: Initial proposal for meta-model definition

[13] SAFE-Project: SAFE-Partners, D6.b Methodology and application rules documentation

[14] Wikipedia (www.wikipedia.org)

[15] SAFE-E Project: SAFE-E Partners, D3.7.a Specification of System Modeling Platform

[16] SAFE-E Project: SAFE-E Partners, D3.7.b System Modeling Package

[17] SAFE-E Project: SAFE-E Partners, D3.7.c Description of System Modeling

The Eurostars Programme is powered by EUREKA and the European Community

EUREKA

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

FFG

10 Acknowledgments

This document is based on the SAFE and SAFE-E projects. SAFE is 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). SAFE-E is part of the Eurostars program, which is powered by EUREKA and the European Community. The work has been funded by the German Ministry of Education and Research (BMBF) and the Austrian research association (FFG) under the funding ID E!6095. The responsibility for the content rests with the authors.