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.b:

Final proposal for extension of meta-model for software and system modeling

Due date of deliverable: 28/02/2013

Actual submission date: 04/03/2013

Organization name of lead contractor for this deliverable: AVL

Editor: Elvira Biendl

Contributors: WT3.2.1 Participants

Reviewer: Philippe Cuenot, Christoph Ainhauser; Markus Ortel, Hans-Leo Ross, Stefan Voget

The Eurostars Programme is powered by EUREKA and the European Community

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

FFG

Revision chart and history log

Version	Date	Reason
0.1	12.11.2012	1st Draft based on D3.2.1.a
0.2	04.02.2013	Update hierarchical structure of the document. Specification of System- and Software-package added. Interfaces to other SAFE meta-model packages added. Preparation for 1st Review
0.3	05.02.2013	Integration of review comments from contributors
0.4	12.02.2013	Update of document structure as preparation for internal review
0.5	18.02.2013	Update Chapter 5.1, 5.2 und 7 for final review with work task participants
1.0	25.02.2013	Update during final review
1.1	28.02.2013	Integration of review comments update of chapter 7
1.2	01.03.2013	Editorial changes

1 Table of contents

The Eurostars Programme is powered by EUREKA and the European Community
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 project is to analyze existing models like EAST ADL, SysML or AUTOSAR with the requirements given in the ISO 26262. The result of this analysis shall provide input for creation of a model that can be used to describe safety relevant systems in accordance with the requirements given in the ISO 26262.

The solution that is described in this document is a draft version 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.

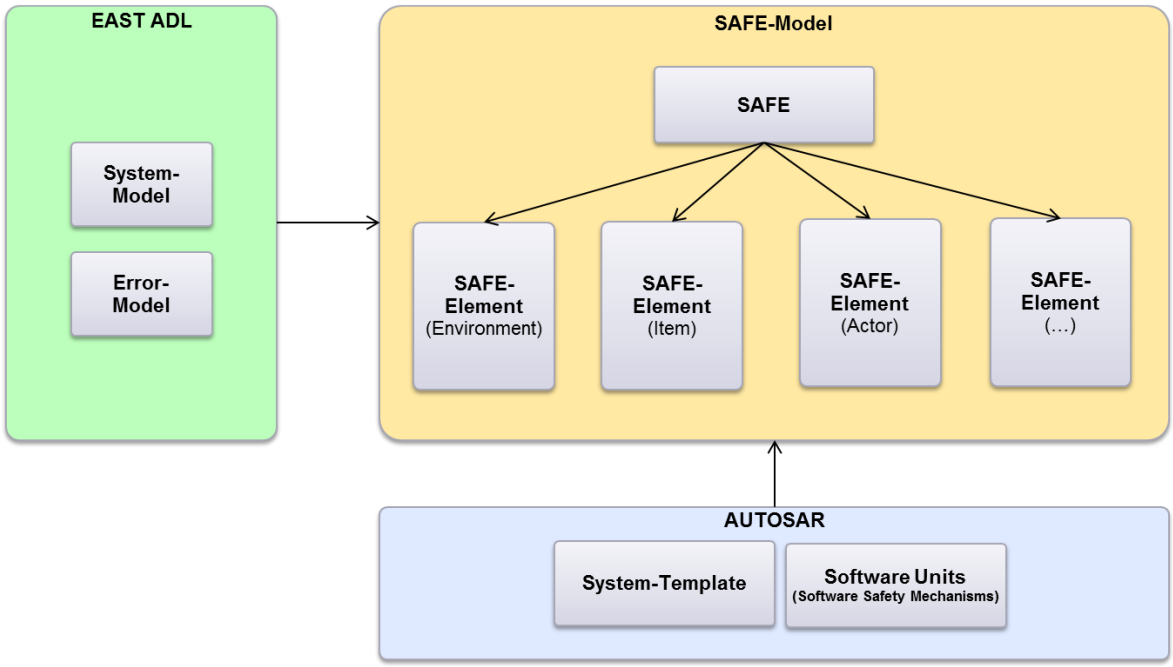

Figure 1: SAFE meta-model

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

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 unreasonable 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.

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

4.1 Abbreviation, Special Terms, Akronyms

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

Abbreviation/ Accronym	Description
ASIL	**A**utomotive **S**afety **I**ntegrity **L**evel
AUTOSAR	**Aut**omotive **O**pen **S**ystem **Ar**chitecture
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.
Controllability	**Controllability** is the ability to avoid a specified harm by any action of the driver or other persons involved during the hazardous event that is currently under analysis.
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)
Exposure	**Exposure** is the state of being in a **hazardous event** that meets with the **failure mode** that currently is under analysis without regarding any already planned safety measures.
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
Severity	**Severity** is the estimation of the extent of harm to one or more individuals that can occur during the hazardous event that currently is under analysis.

4.2 Scope of the document

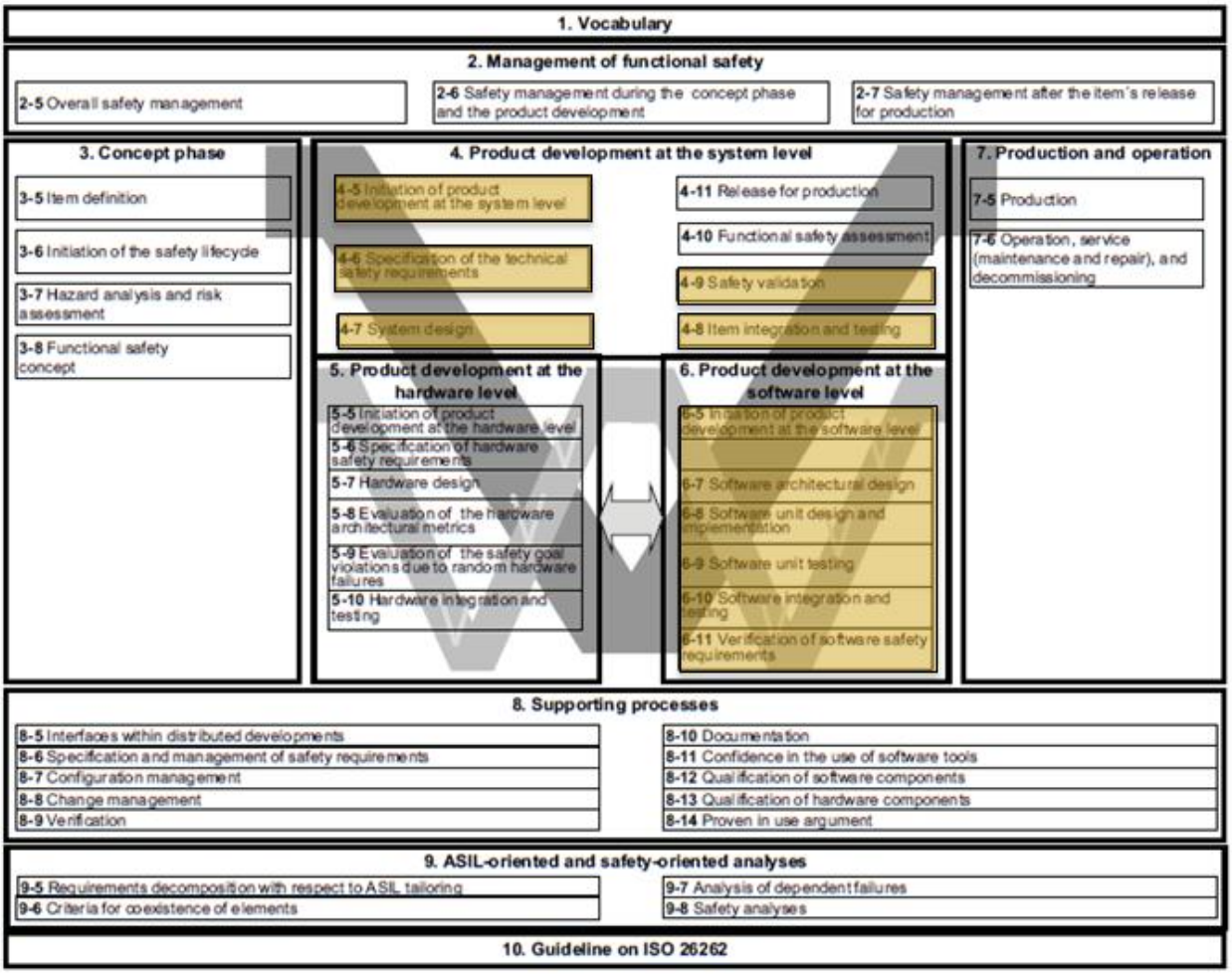

Figure 2: Scope of this Document

This document is created based on the requirements allocated to work task WT3.2.1. The allocation of the requirements is documented in the referenced deliverables D2.1.b [5]. Based on these requirements the specification of SAFE meta-model package system was created. It describes how to model a safety relevant item according to ISO 26262 by using already existing models like EAST-ADL and AUTOSAR.

This specification shall be used as a base for discussions with the EAST-ADL and AUTOSAR consortium how to handle the described topics in future.

5 System Package Specification

This chapter contains the specification of elements that are needed to fulfill the requirements allocated to WT3.2.1. The system package contains

- description of the different architectural level of an item
- Safety measures and safety mechanisms to avoid, mitigate, detect or control safety relevant failures
- Safety Architecture as a base for safety related analyses.

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 goal analyzed in the Hazard Analysis and Risk Assessment and the technical solution described in the safety requirement documentation.

5.1 Architectural Overview

The following chapters describe the architecture of the SAFE meta-model packages that have interfaces to the system package.

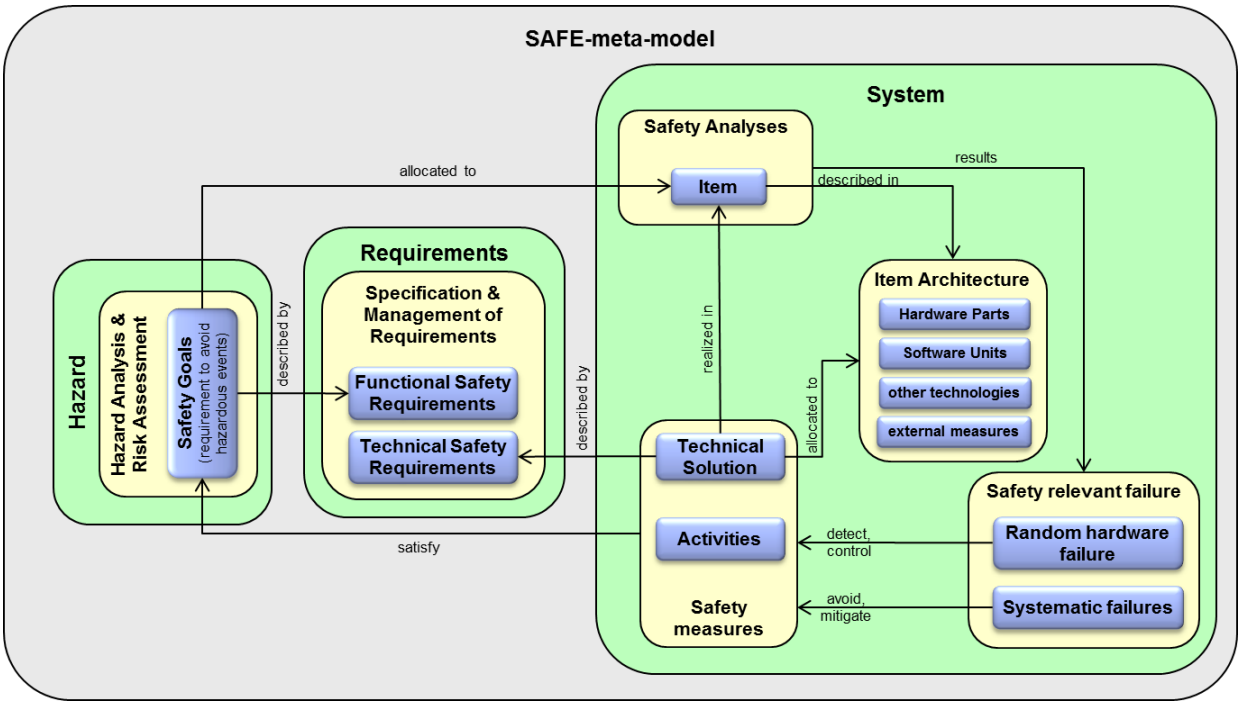

Figure 3: Architectural Overview

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

5.1.1 Hazard

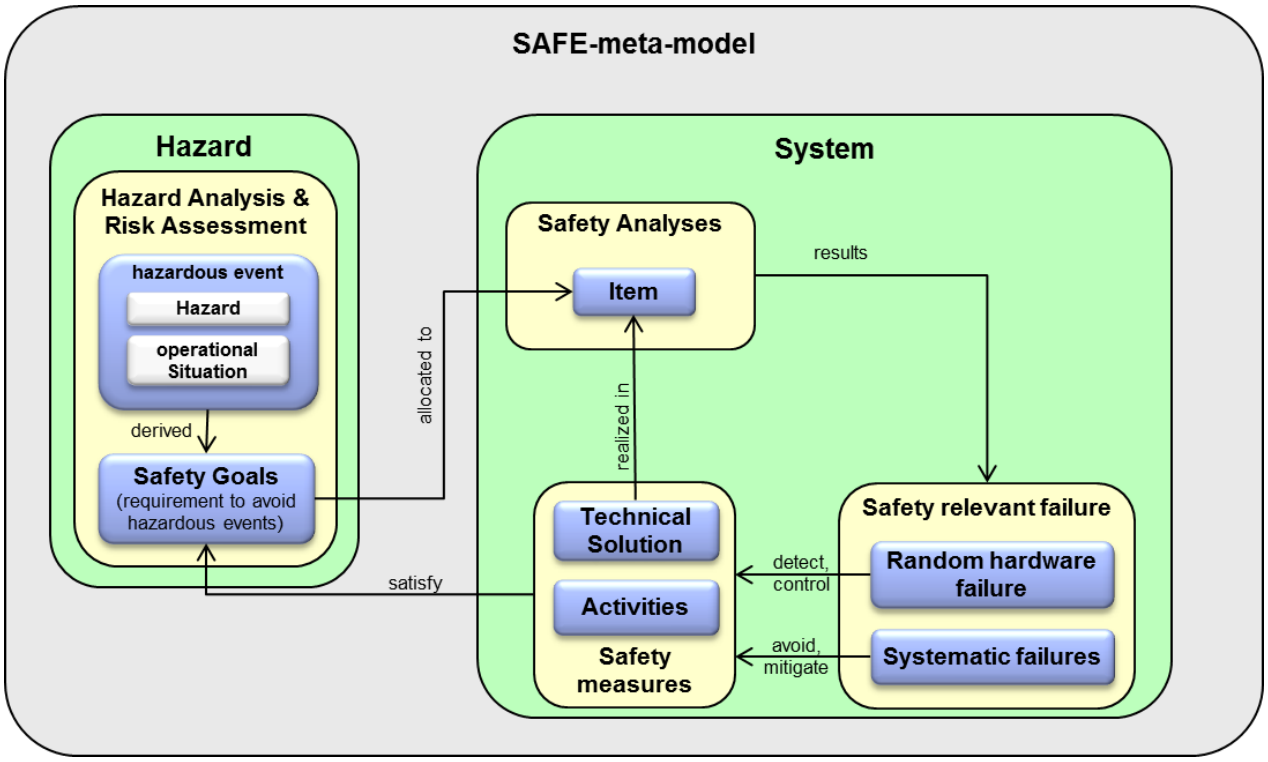

Figure 4: Interfaces to Hazard packages

The safety goals are derived from hazardous events analyzed in the Hazard Analysis and Risk Assessment. Hazardous event is a combination of a hazard with an operational situation. The hazard analysis and risk assessment shall be executed based on the item definition. Further details according to Hazard Analysis and Risk Assessment are described in D3.1.1.b [6].

The safety goals shall be

- described as functional safety requirements (see 5.1.3) and
- allocated to architectural elements (see 5.3.1.1) of the item.

Safety Analyses shall be executed to identify safety relevant failures.

A technical solution shall be defined 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.

5.1.2 System

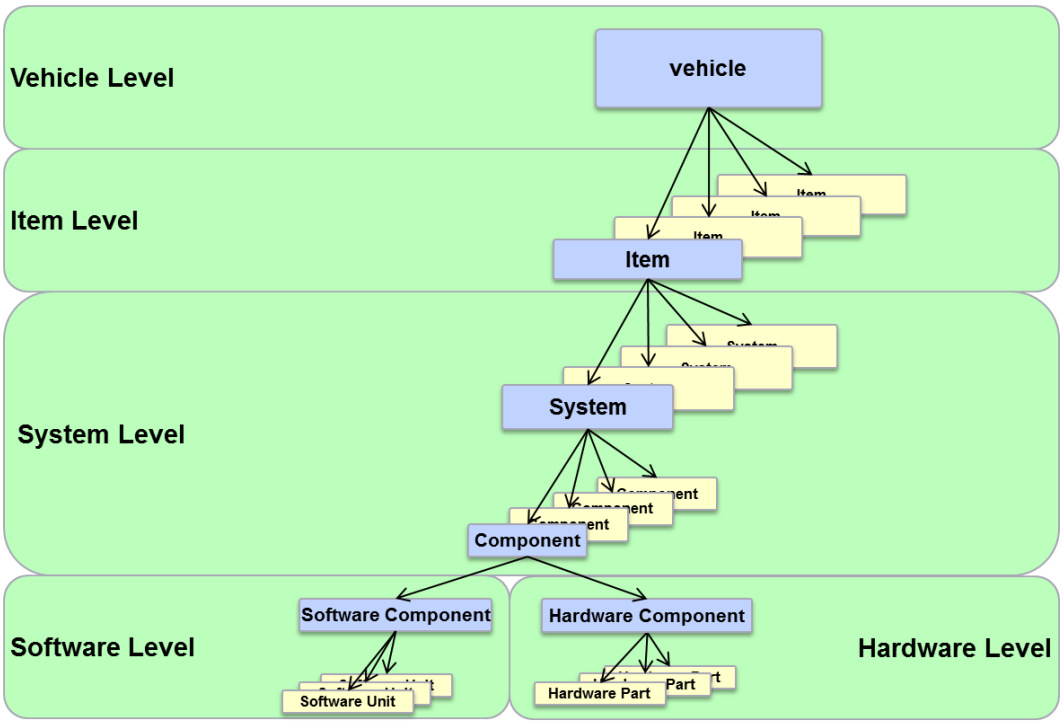

Figure 5: Item Architecture overview

This package contains all needed artifacts to model a safety-related system in accordance to the requirements of the ISO 26262.

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.

Safety Analyses shall be done on different levels of the item architecture. Therefore the SAFE meta-model provides different architectural levels.

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

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.

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 each software component to 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 each hardware component to Hardware Parts. The architectural description of the interfaces between the Hardware Parts is also part of this level.

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

5.1.3 Requirements

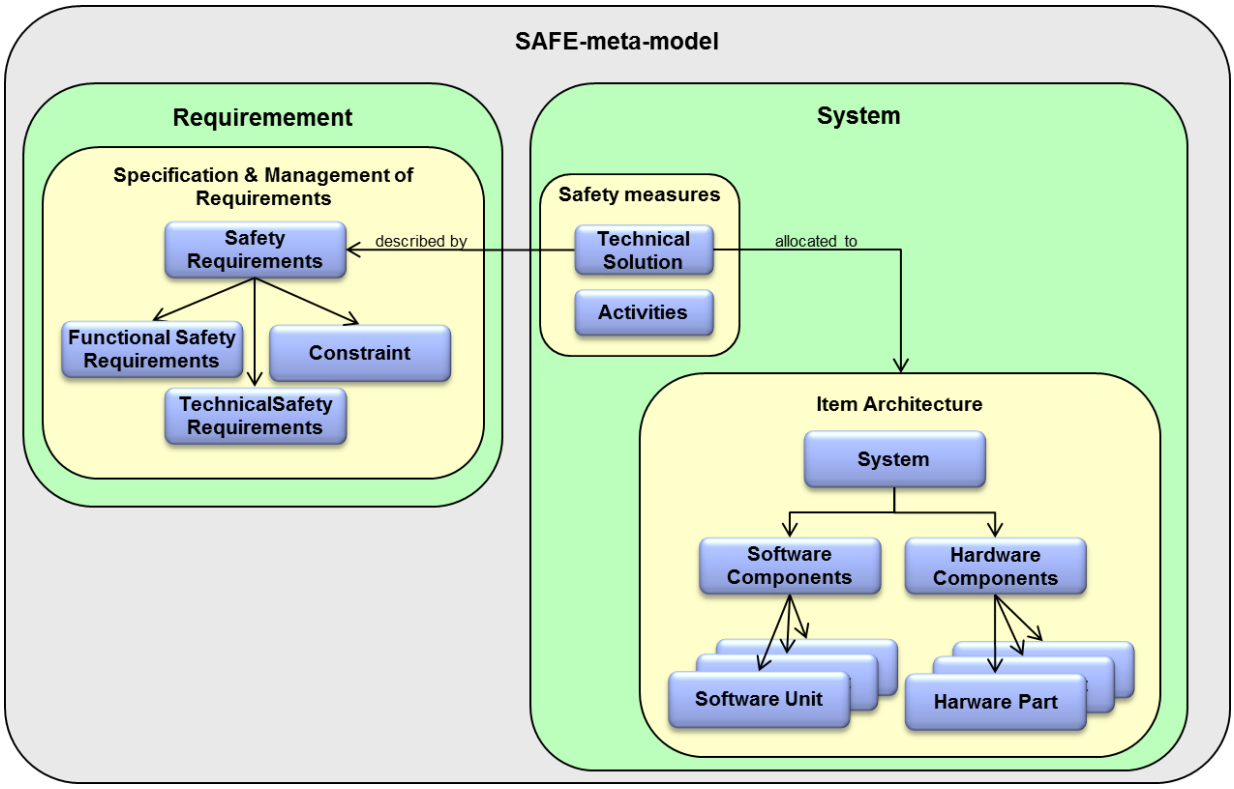

Figure 6: Interfaces to Requirements packages

Safety Requirements shall be categorized into different groups:

- Functional Safety Requirement
- Technical Safety Requirement
- Constraint

Constraints describe for example architectural assumptions or design constrains given from the higher level architecture.

Further details to constraints see chapter 7.1.1.1. Detailed description of safety requirements see D3.1.2.b [7]

5.2 Vehicle Level

5.2.1 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.

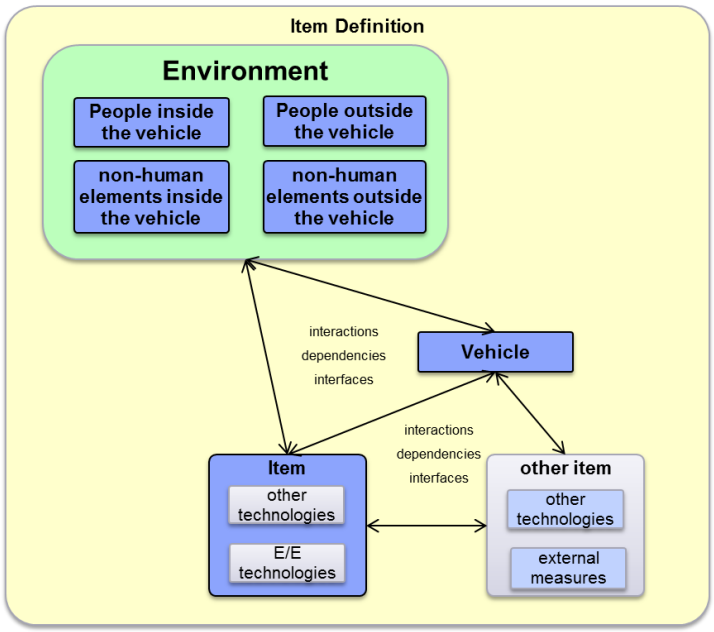

Figure 7: Item Definition

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. Further details related to this topic see chapter 7.1.6.

5.2.1.1 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.

5.2.1.2 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.

5.3 Item Level

5.3.1 Item Structure

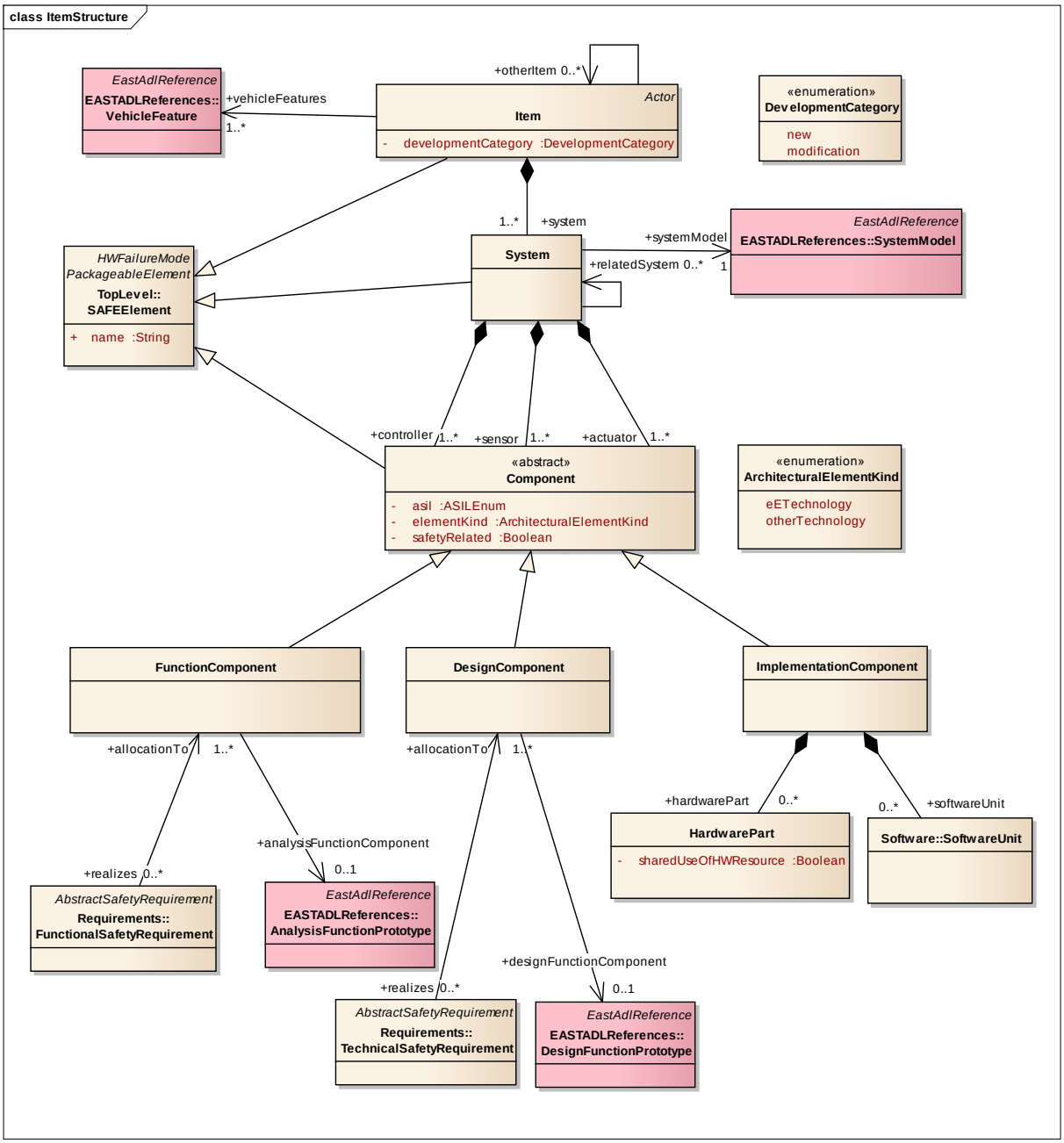

Figure 8: Item Structure

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

The item structure is used to generate the architectural overview of the item. The SAFE meta-model shall be able to refer to external models to implement systems that are part of the item.

The Item can consist of one or more Systems. The following figure contains references to the system model given in EAST-ADL. Each system shall be represented by one EAST-ADL ***SystemModel***. The item is represented by an EAST-ADL ***VehicleFeature.***

The safety-relevant ***System*** shall contain at least one sensor, one actuator and one controller. The SAFE meta-model describes them as ***Components.*** The Component level contains three different component types.

FunctionComponent
Non-system level element that is logically and technically separable described on a functional abstraction level.

DesignComponent
Non-system level element that is logically and technically separable described on a system design abstraction level.

ImplementationComponent
Non-system level element that is logically and technically separable. The Implementation Component is comprised of

- one or more than one hardware parts or
- one or more than one software units.

5.3.1.1 Architectural elements

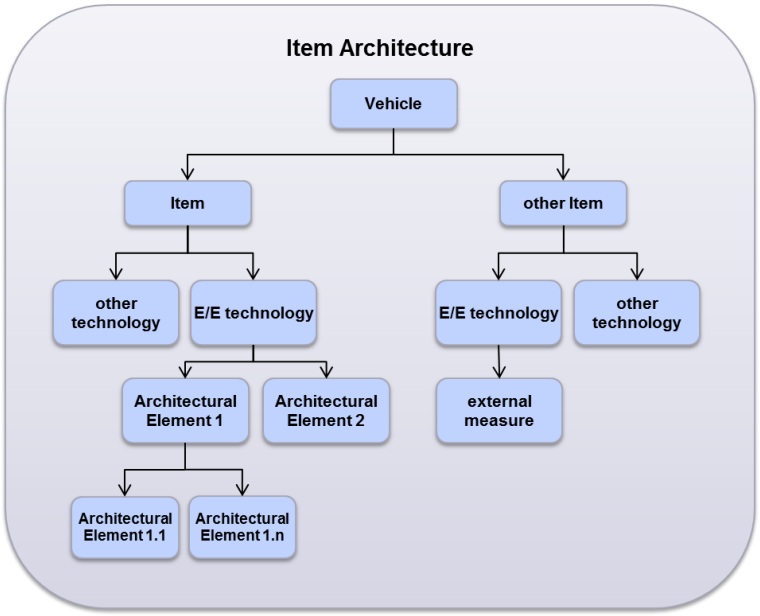

Figure 9: Item Architecture

The SAFE meta-model shall contain architectural elements.

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

For further details to this topic see chapter 7.1.2.7.

5.3.2 Safety Concept

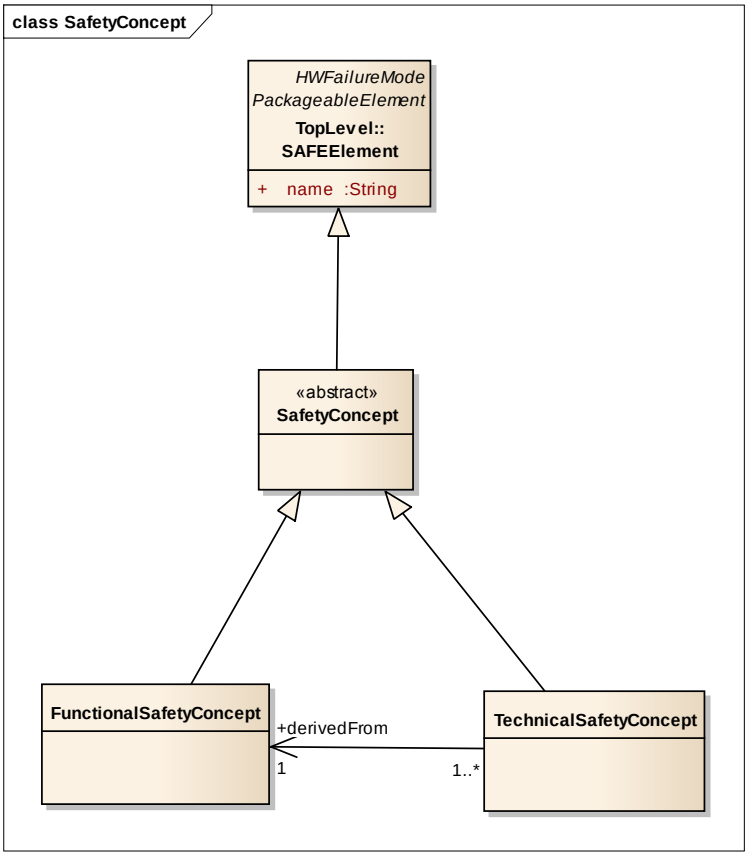

Figure 10: Safety Concept

The ***SafetyConcept*** shall be defined as a Top Level Element in the SAFE meta-model. One or more ***TechnicalSafetyConcepts*** can be derived by the ***FunctionalSafetyConcept***.

5.3.2.1 Functional Safety Concept

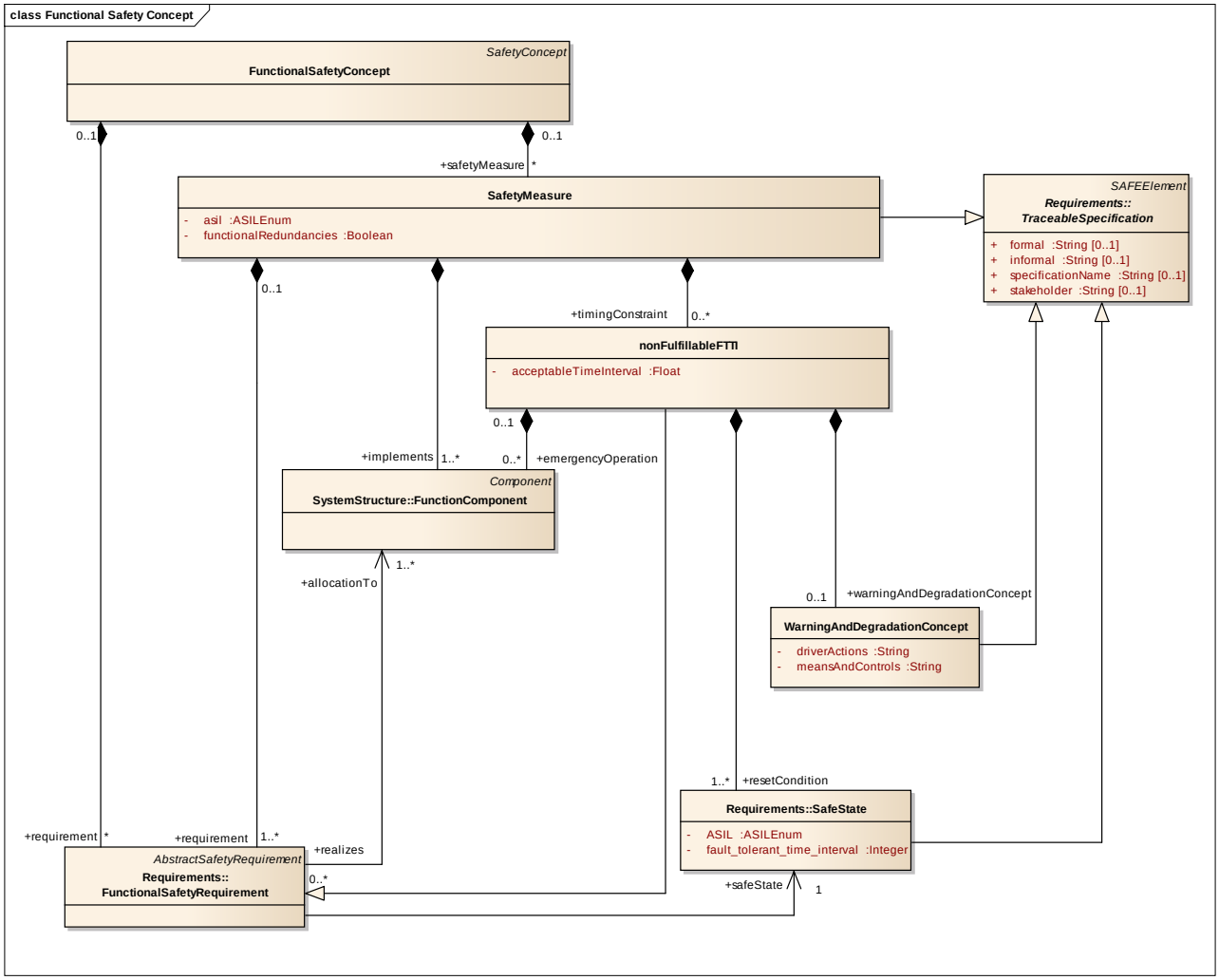

Figure 11: Functional Safety Concept

The ***FunctionalSafetyConcept*** describes the safety measures that are needed to avoid violation of safety goals. 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.

The allocation and distribution of ***FunctionComponents*** that are used to realize ***SafetyMeasures*** shall also be part of the ***FunctionalSafetyConcept***.

If it is not possible to reach the safe state within the defined fault tolerance time interval a warning and degradation concept shall be specified for this case. For further details to this topic see chapter 7.1.2.4.

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

5.3.2.2 Safety Measures

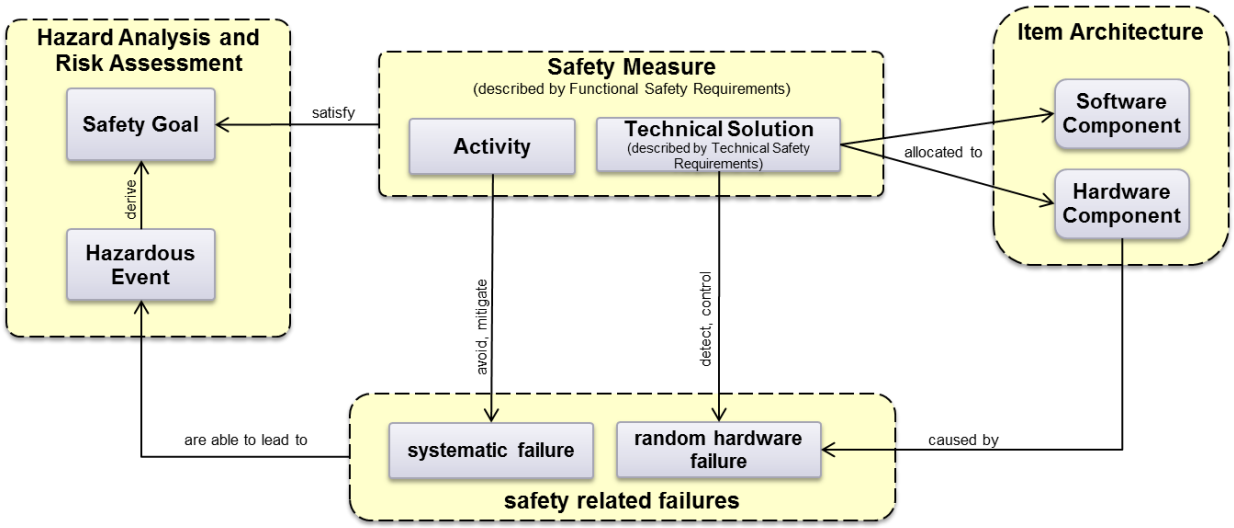

Figure 12: Safety Measures

The ISO 26262 describes different kinds of safety measures:

- an 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

Safety measures can also contain requirements according to production, operation, service and decommissioning instructions, if they are needed to satisfy at least one allocated safety goal.

Safety Measures are specified to satisfy the derived Safety Goals. They are described by functional safety requirement and are part of the Functional Safety Concept.

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

5.3.2.3 Technical Safety Concept

The Technical Safety Concept is derived by the Functional Safety Concept. It contains the refinement of the functional safety requirements and the allocation to the architectural elements.

The Technical Safety Concept shall contain requirements according to production, operation, service and decommissioning like

- 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

if they are needed to fulfill at least one of the safety goals allocated to the item.

The Technical Safety Concept refines the technical solution described in the Functional Safety Concept. The traceability shall be given from the safety goal, derived on vehicle level to the safety mechanisms specified in the Technical Safety Concept. The allocation of the safety mechanisms to hardware parts or software units shall be described in the Technical Safety Concept.

Interfaces between safety relevant software units and safety relevant hardware parts that are needed to realize safety mechanisms shall be specified in the Hardware Software Interface Specification. Further details to this topic see chapter 7.1.3.3.

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

5.3.2.4 Safety Mechanisms

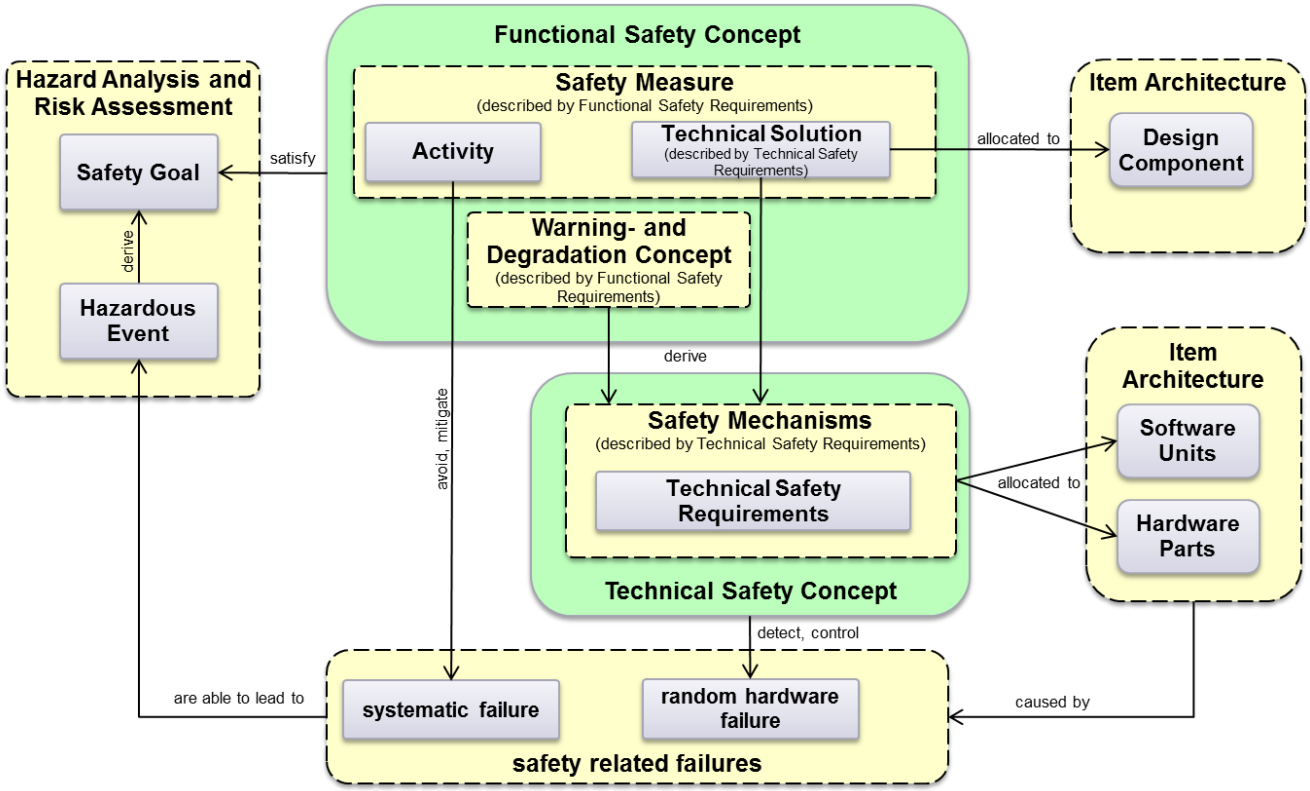

Figure 13: Safety Mechanism Structure

A safety mechanism is a technical solution implemented by E/E functions to detect faults or control failures that are able to lead to a violation of a safety goal. Safety mechanisms are derived by safety measures defined to avoid or control systematic failures or to detect random hardware failures. They are described by Technical Safety Requirements.

The safety mechanism shall be allocated to the corresponding architectural element in the item architecture. That means software safety mechanisms shall be allocated to those software unit where they are implemented. Hardware safety mechanisms shall be allocated to the hardware parts that realize the mechanisms.

Further details to the topic safety mechanisms see chapter 7.1.2.6.

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

5.4 System Level

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

5.4.1 System Design

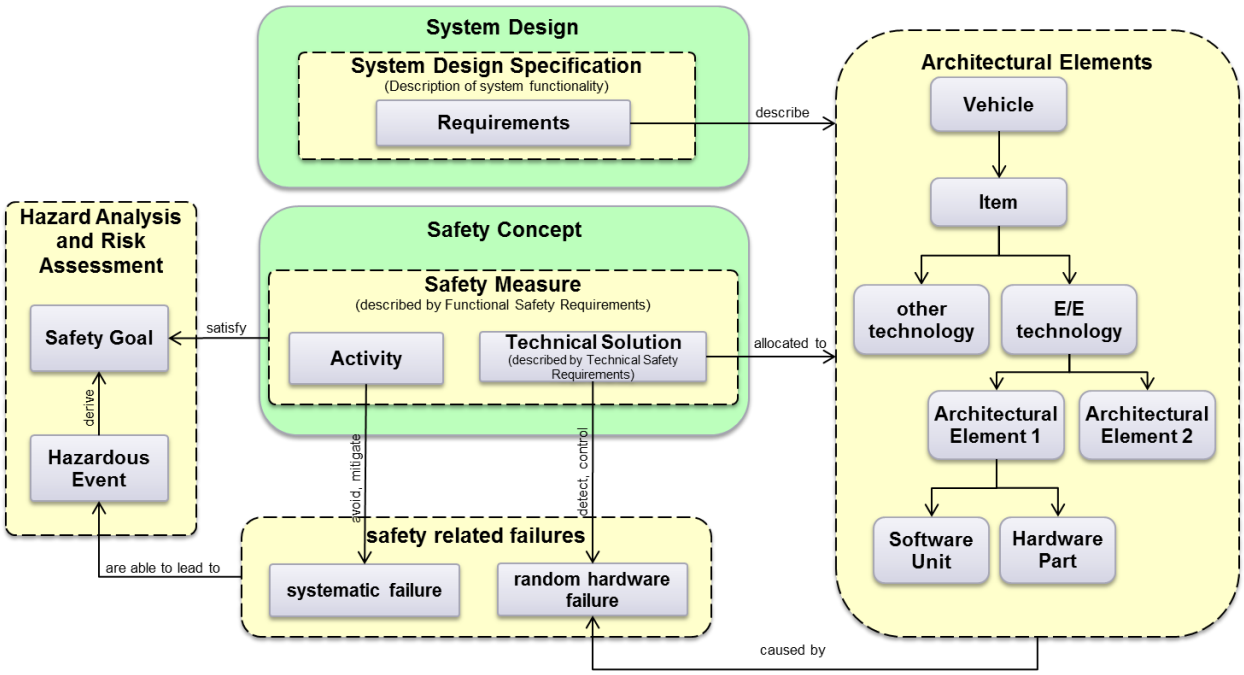

Figure 14: System Design

The System Design shall contain the specification and the architecture of the safety relevant functionality. 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. For further details to this topic see chapter 7.1.3.6

The target value for single-point fault metric and latent-point fault metric for the architectural element of the item architecture on system level shall be specified in the System Design. For further details to this topic see chapter 7.1.5.1

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

5.4.1.1 System Array

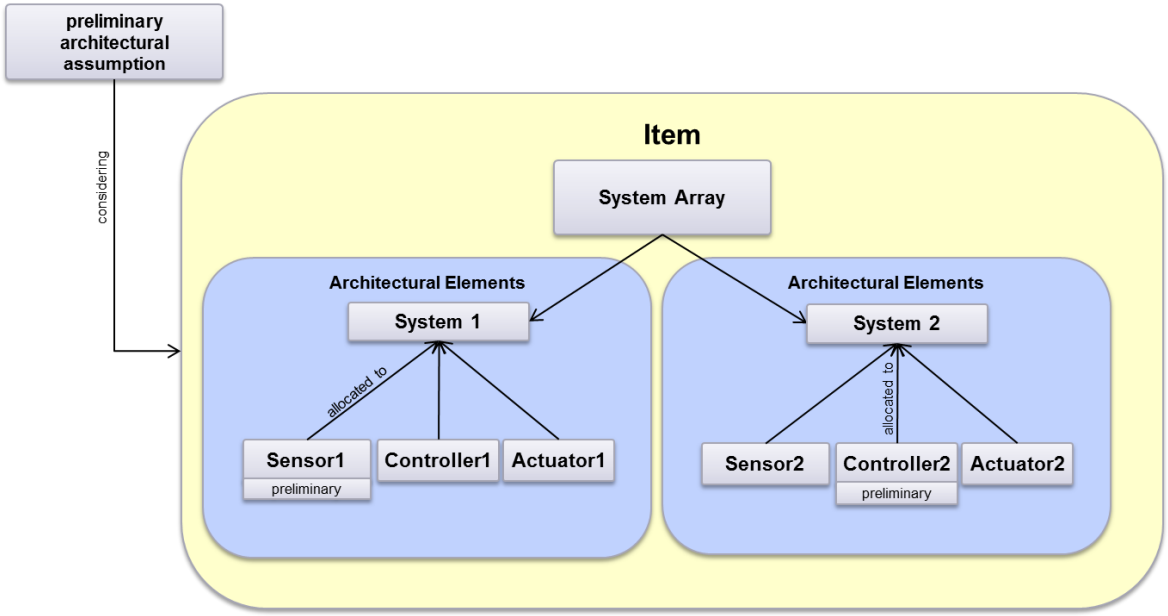

Figure 15: System-Array

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.

6 Implementation of the SAFE meta model

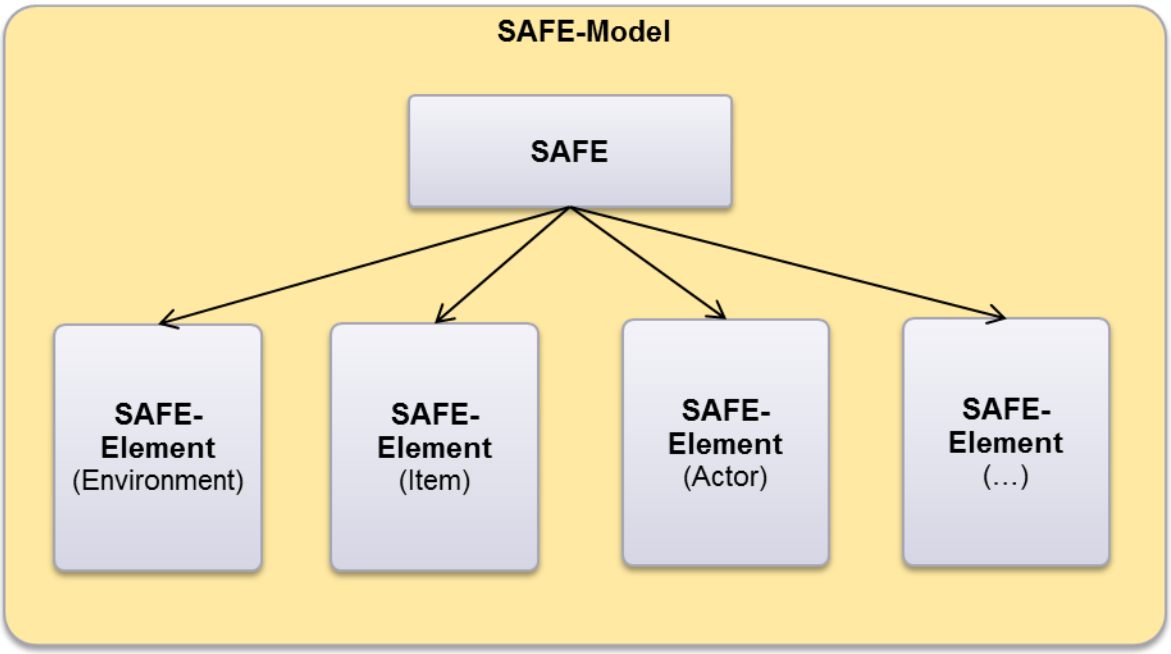

Figure 16: SAFE meta-model

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.

6.1 Description of the SAFE meta-model

The description of all the artifacts contained in the SAFE meta-model is part of D3.5.b [11]

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
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
FFG

7 Further Topics and Outlook

This chapter contains topics that will be discussed in further improvement steps of the SAFE meta-model. They address requirements that are allocated to the system and software package described in this document.

Many of the topics addressed in this chapter provide input to other work tasks that are also part of the SAFE-project. Work task 6 for example will provide a series of guidelines for the use of the methods and tools developed in the preceding phases of the SAFE project.

7.1.1 Further Topics addressed to Package Requirement

The following topics are part of the next iteration step of the requirement package.

7.1.1.1 Categories of Safety Requirements

As described in chapter 5.1.3 safety requirements contain functional and technical safety requirements. But there shall also be a solution for modeling of

- Quantitative requirements
- Constraints
- Requirements addressed to production process
- Requirements addressed to process activities

Quantitative Requirements
A quantitative requirement is used to specify for example the random hardware failure target value of components or the target value for the diagnostic coverage of a safety mechanism.

Constraints
Safety relevant systems also contain constraints for example from higher architectural levels or given from the already existing design (e.g. environmental conditions, functional constraints, design constraints…).

Requirements addressed to production process
Violation of a safety goal can also be caused by topics addressed to the production process. In this case safety measures shall be defined to avoid this violation of a safety goal.

Process Requirements
As described in chapter 5.3.2.2 safety measures consist of activities and technical solutions. The technical solution is described by technical safety requirements, but the actual specification of the SAFE meta-model does not contain a category for requirements that describe activities that are needed to avoid safety goal violation.

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

7.1.1.2 Safety requirement documents

The actual specification contains parts of the safety requirement documentation:

- Functional Safety Concept (see chapter 5.3.2.1)
- Technical Safety Concept (see chapter 5.3.2.3)
- System Design (see chapter 5.4.1)

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.

7.1.1.3 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 (see chapter 7.1.7)

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

7.1.1.4 Maturity of safety requirements

During the safety lifecycle the safety requirement can have a different level of maturity. To ensure an effective development process the maturity of the safety requirement shall be visible for the user. The maturity of the safety requirement can be shown by its status (e.g. new, accepted, rejected, clarify…)

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

7.1.1.5 Usage of Safety Requirements

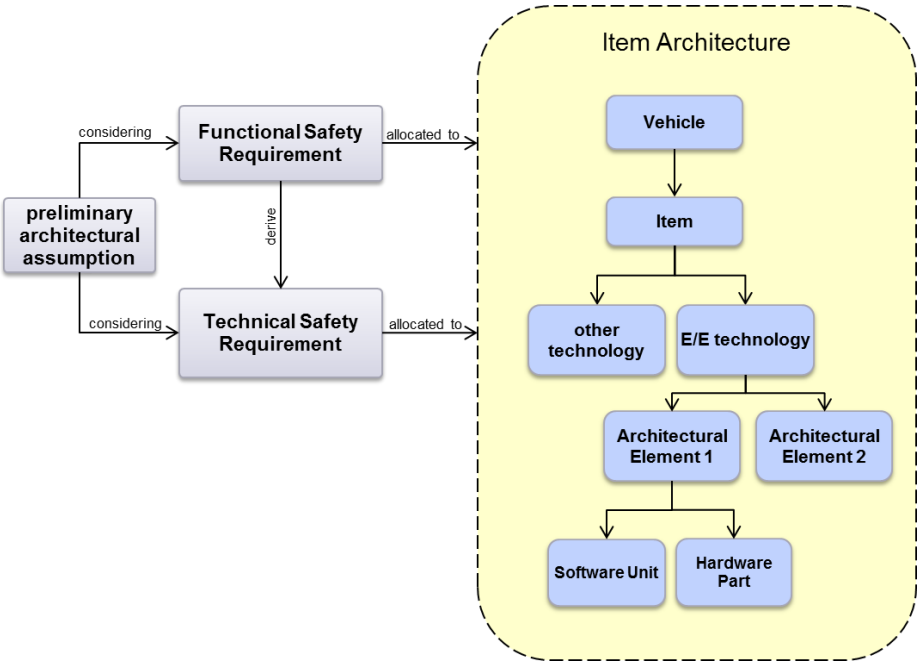

Figure 17: Outlook - Safety Requirements

Technical Safety Requirements shall be specified in accordance with the following information that is already defined in System/Item scope:

- System/Item-Interfaces (part of the Functional Safety Concept external interfaces of the item such as communication and user interfaces
- environmental conditions or functional constraints
- system configuration requirements

All System/Item scope specific information that is used to derive technical safety requirements shall be allocated to get the bidirectional traceability of source information and derived requirements.

Technical Safety Requirements shall be allocated to the preliminary System/Item-Architecture that is part of the Functional Safety Concept. The preliminary architectural assumptions shall be considered by deriving the Technical safety requirements from the functional safety requirements.

7.1.1.6 Verification of safety 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

7.1.2 Further Topics on Item Level

The following topics are part of the next iteration step of the system package.

7.1.2.1 Environment

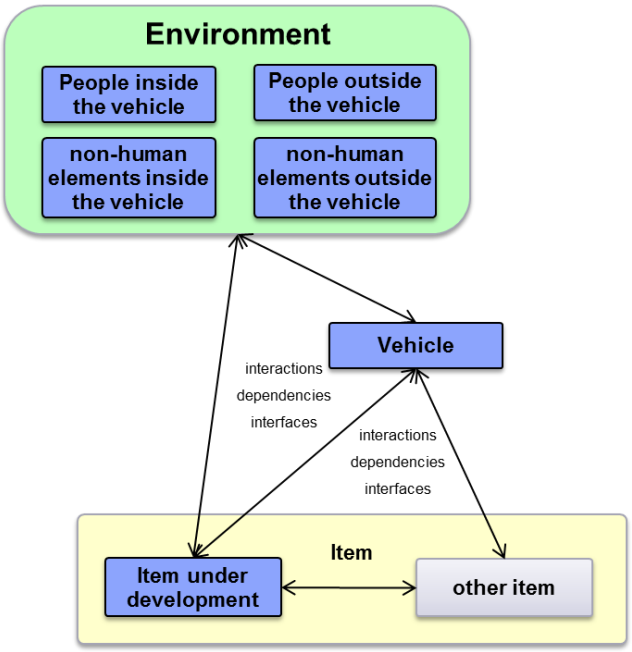

Figure 18: Outlook - Item Environment

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

All interactions of the item to its environment shall be specified that have the potential to influence the safety relevant funcitonality.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.

7.1.2.2 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
- modification

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.

7.1.2.3 Safety Element out of Context (SEooC)

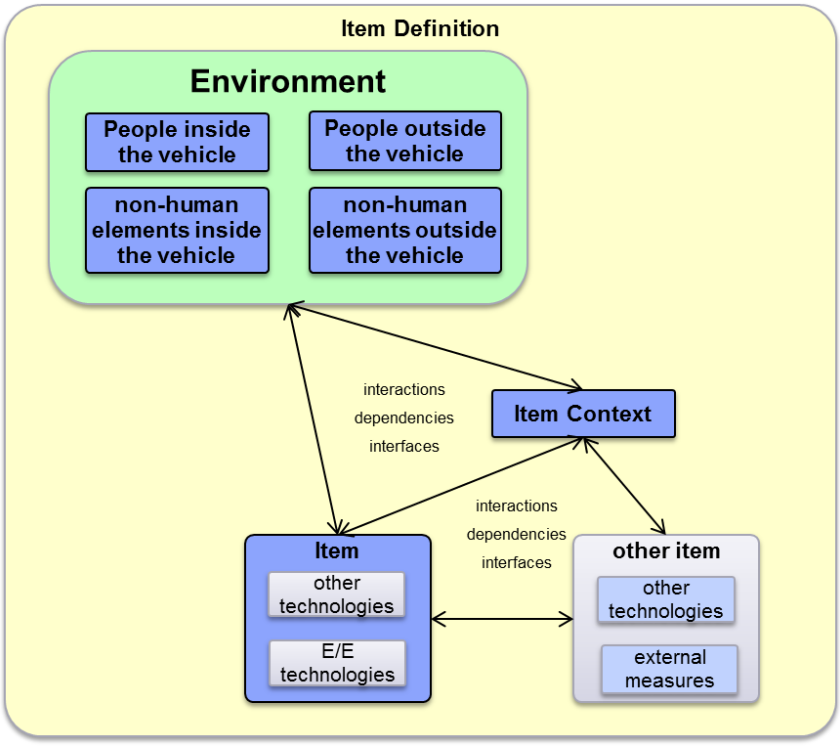

Figure 19: Outlook - Safety Element out of Context (SEooC)

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.

7.1.2.4 Warning- and Degradation Concept

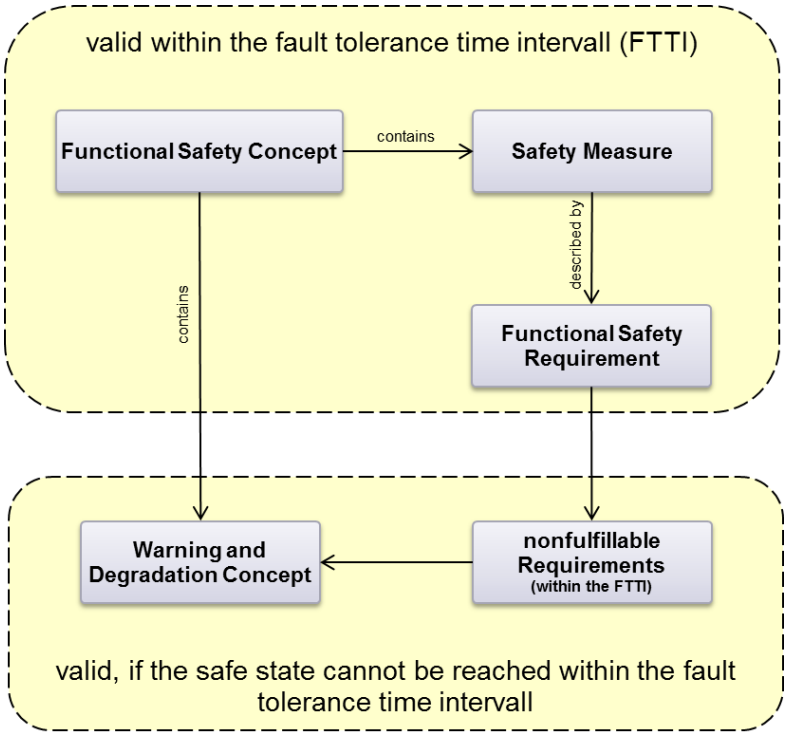

Figure 20 : Outlook – Warning- and Degradation Concept

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

7.1.2.5 Safety Measure

Figure 21: Outlook - Safety Measures

As described in chapter 5.3.2.2 safety measures contain technical solutions and activities. Activities shall be planned to avoid or mitigate systematic failures. Further details see chapter 7.1.7

7.1.2.6 Safety Mechanism

Safety mechanisms are described in chapter 5.3.2.4. The following topics shall be discussed during the next iteration step of SAFE meta-model.

Category:

Safety Mechanisms can be used to achieve different targets. These targets shall be defined for each safety mechanism by selecting on 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

Content:

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

Safety mechanisms shall specify the behavioral description to achieve or maintain the safe state.

Therefore it shall contain

- fault tolerance time
- operation modes
- emergency operation intervals
- functional redundancies
- safe state
- transition from the hazardous event to the safe state
- allocation to the corresponding warning and degradation concept, if needed

Safety Mechanism Structure:
During the next iteration step of the system package a structure shall be provided to model the safety mechanisms realized by Software Units as well as for the safety mechanisms realized by Hardware Parts.

7.1.2.7 Architectural Elements

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.

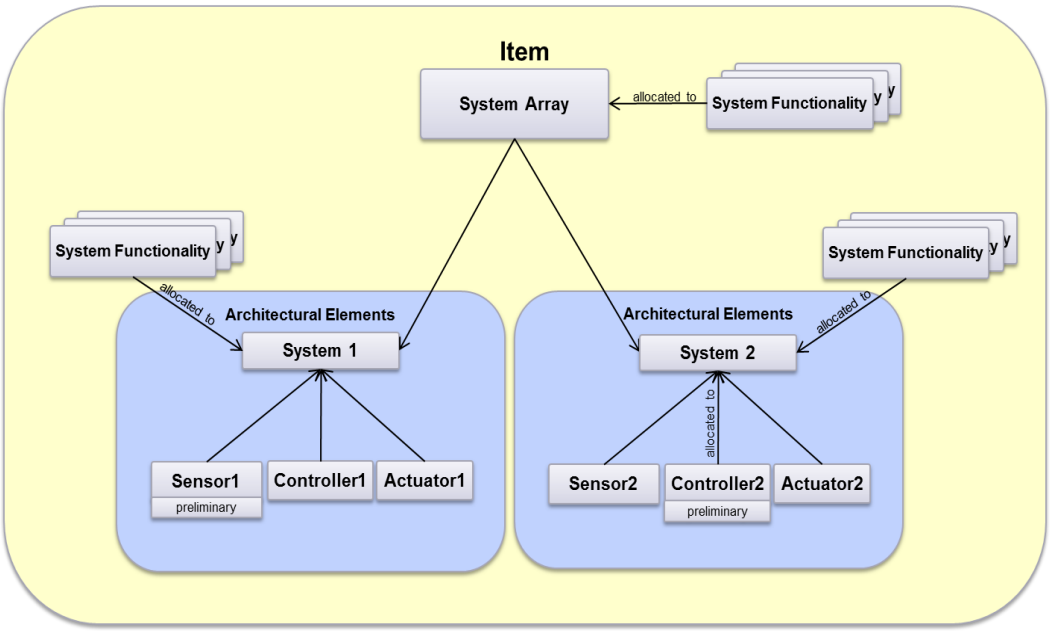

Figure 22: Outlook - Architectural Elements

The maturity of the architectural elements shall also be part of the SAFE meta-model. 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.

7.1.3 Further Topics on System Level

The following topics are part of the next iteration step of the system package.

7.1.3.1 System Design

As described in chapter 5.4.1 the following topics shall be detailed in the next iteration step.

Content:
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

Well-trusted Design Principles
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.

7.1.3.2 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.

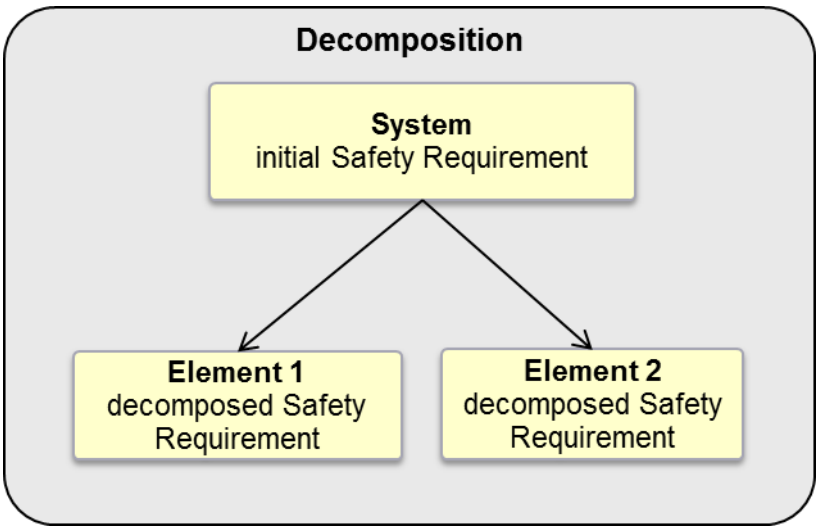

Figure 23: Outlook - Decomposition

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 [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:

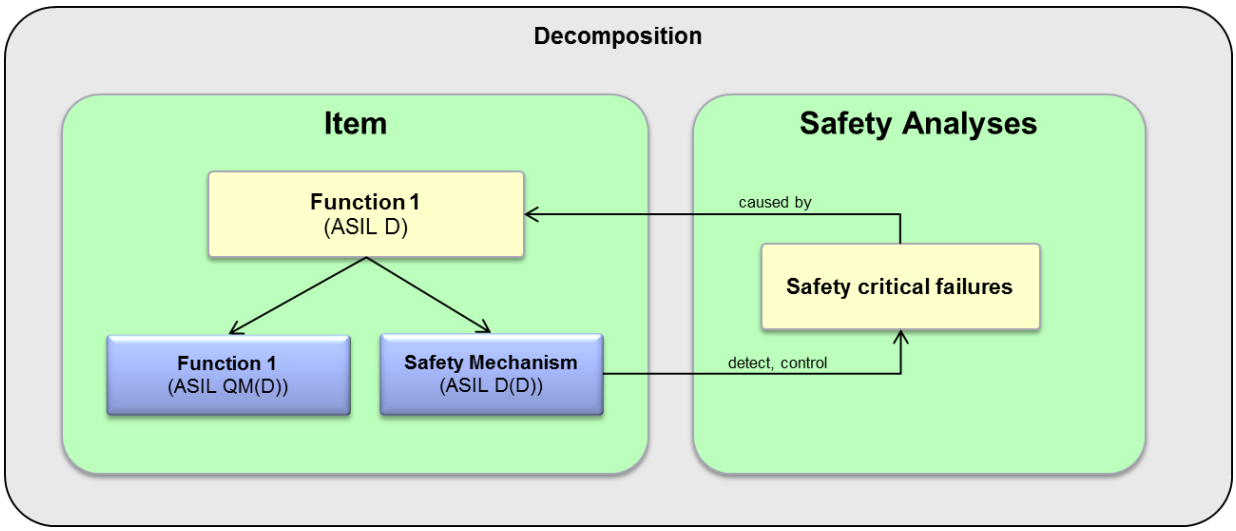

Figure 24: Outlook - Decomposition Function + Safety Mechanism

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

7.1.3.3 Hardware-Software Interface

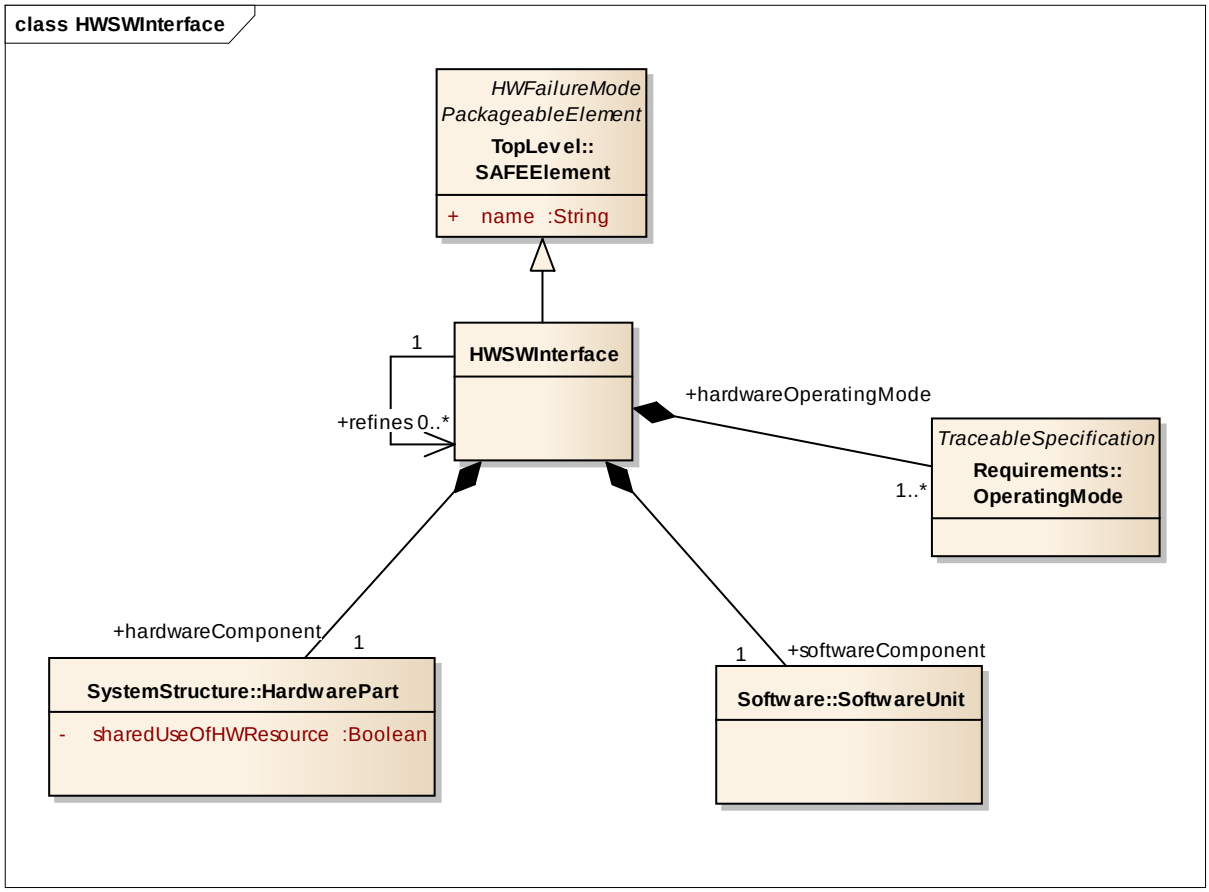

Figure 25: Outlook - Hardware Software Interface Specification

The Hardware Software Interface Specification (HSI) is input for the Hardware development phase.

The HSI shall

- describe the diagnostic capabilities of the hardware elements and their use by 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 the following characteristics:
  - relevant operating modes of hardware elements/parts and the relevant configuration parameters
  - the hardware features that ensure the independence between elements and that support software partitioning
  - shared and exclusive use of hardware resources

- the access mechanism to hardware devices
- the timing constraints defined for each service involved in the technical safety concept

7.1.3.4 System Failure Propagation

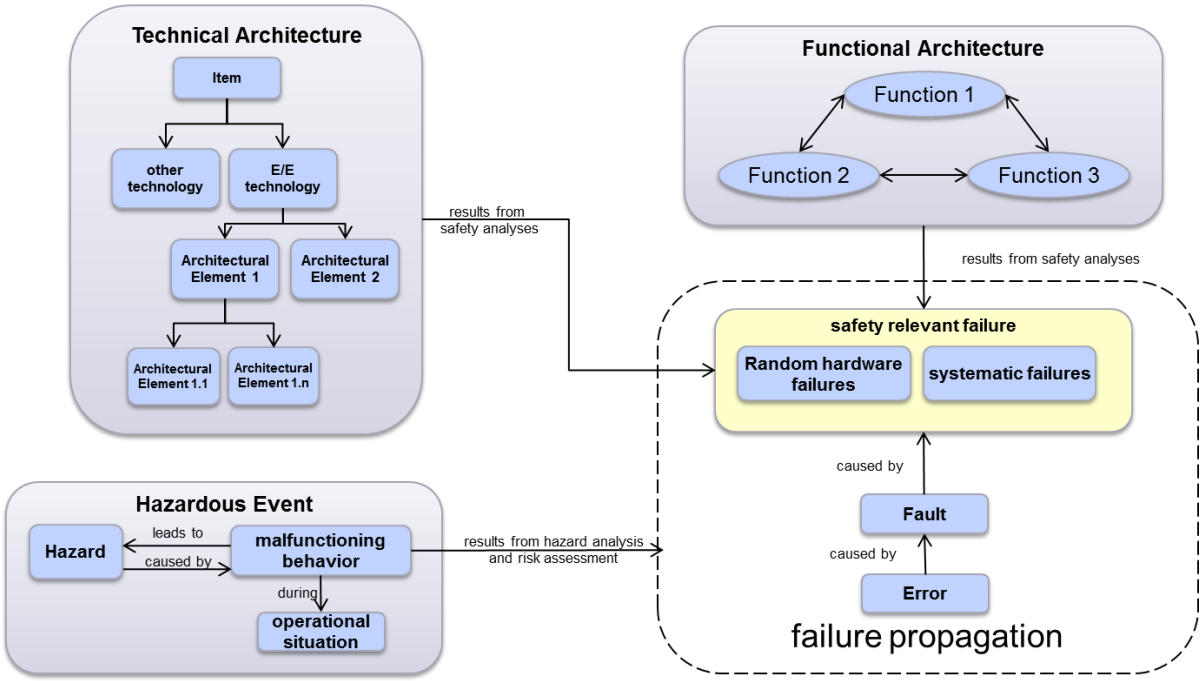

Figure 26: Outlook - Failure Propagation on system level

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

The next iteration step of the system package shall provide a solution for this topic.

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

7.1.3.5 Safety relevant failures

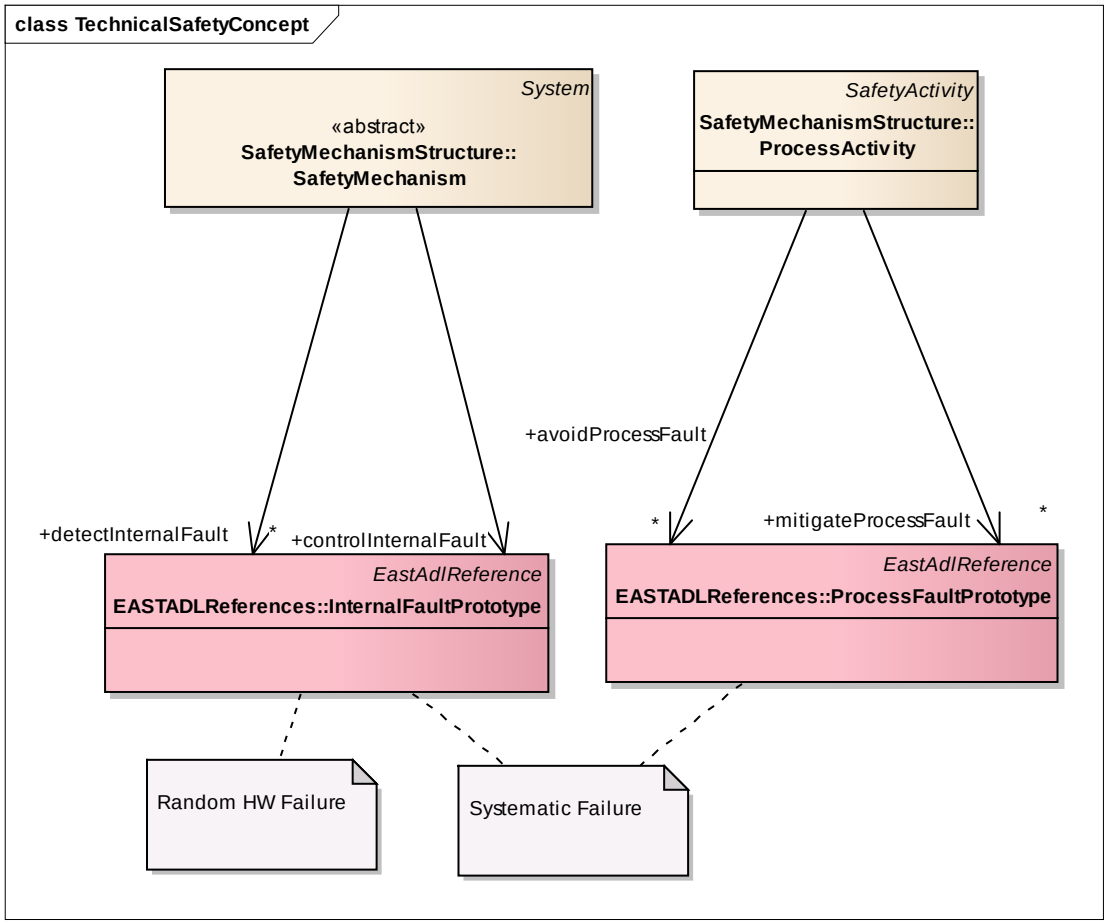

Figure 27: Outlook - Safety relevant failures

ProcessFaultPrototype defined in the EAST-ADL Error-Model could be used as an external reference to model the systematic failures that are able to lead to a violation of a safety goal.

InternalFaultPrototype defined in the EAST-ADL Error-Model could be used as an external reference to model the random hardware failures that are able to lead to a violation of a safety goal.

This topic will be discussed during the next iteration step of the system package.

7.1.3.6 Criteria of the coexistence of elements

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

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

7.1.3.7 Safety Analyses

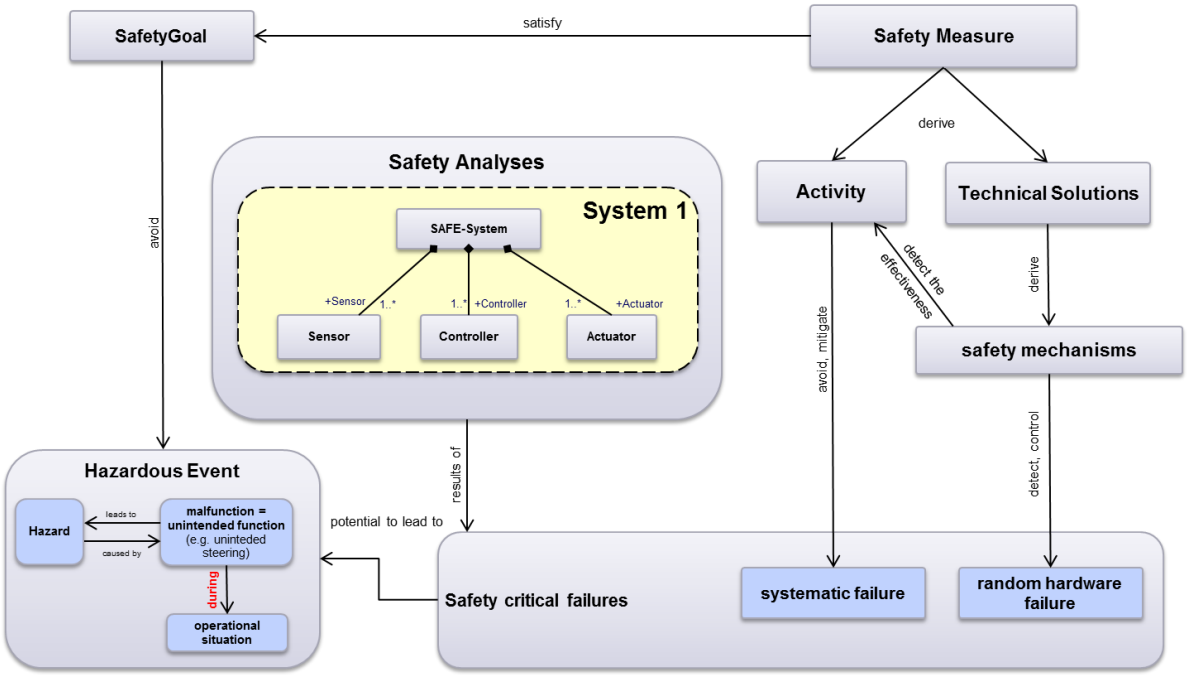

Figure 28: Outlook - Safety Analyses

Safety Analyses shall be executed to identify the safety relevant failures that are able to cause harm to people. Safety Measures shall be included to the functional safety concept to reduce or mitigate the safety relevant failures to an unreasonable level of risk. These Safety Measures shall be derived to technical solutions described in the system design.

The results of the safety analyses are the input for developing an appropriate solution of the specified functionality with an appropriate level of functional safety.

The technical solution shall contain safety mechanisms to avoid, control or mitigate safety relevant failures that are identified during the safety analyses. The safety mechanisms shall be implemented in software and hardware components.

Safety Analyses can be executed as quantitative or qualitative analysis.

Stimulus:

A stimulus in scope of the ISO 26262 is an impact received by the item that affects the specified system behavior.

The stimulus can be received form the environment, a system or an element of a system.

Stimuli include failures in combination with relevant operating mode and defined system states.

Evidence of Effectiveness

Safety Mechanisms are specified to avoid, control or mitigate safety relevant failures. These failures are the results of the safety analyses. To verify the effectiveness of the safety mechanisms test cases shall be defined. The test cases shall be allocated to the corresponding safety mechanism.

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

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.

7.1.4 Further topics on software level

The following topics are part of the next iteration step of the software package.

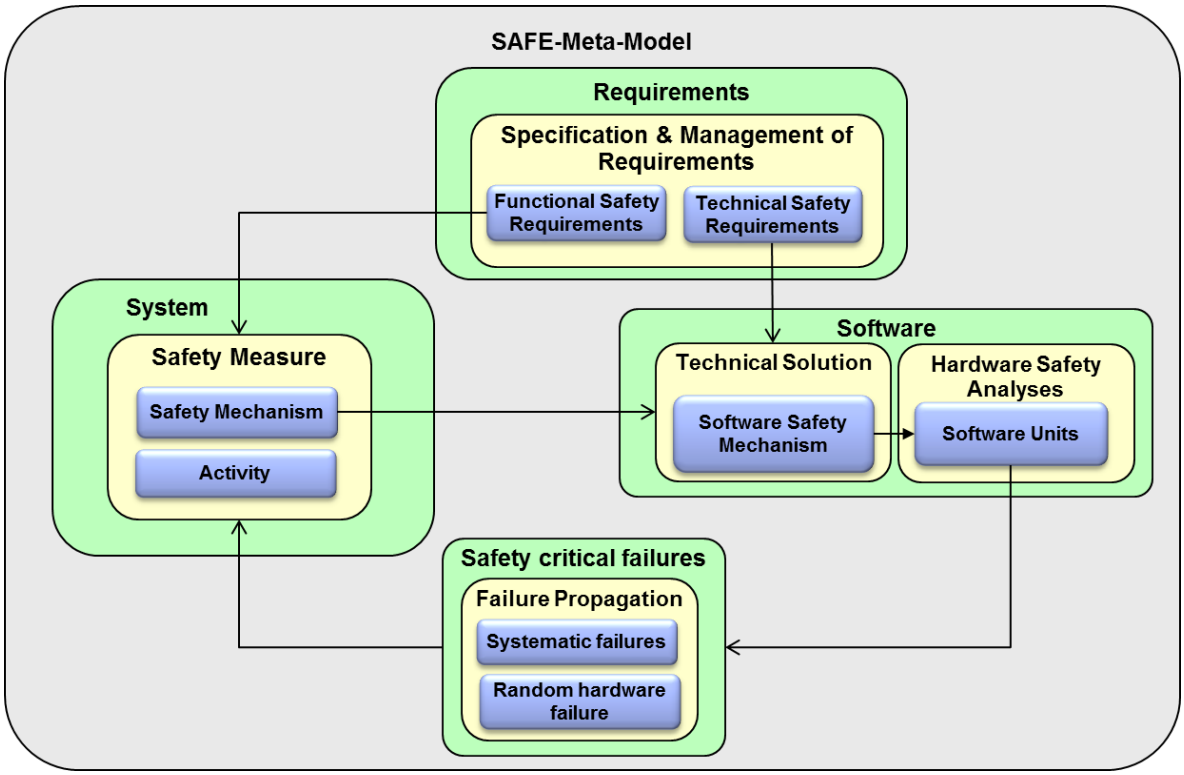

Figure 29: Outlook - Interface to Software Package

7.1.4.1 Software Component Model

As a basis for generating software assets, which realize measures necessary to integrate safety relevant software components, the model will support the explicit expression of applied integration measures.

7.1.4.2 Software Architecture

Therefore the following architecture is supposed for a software system specified in the SAFE meta-model.

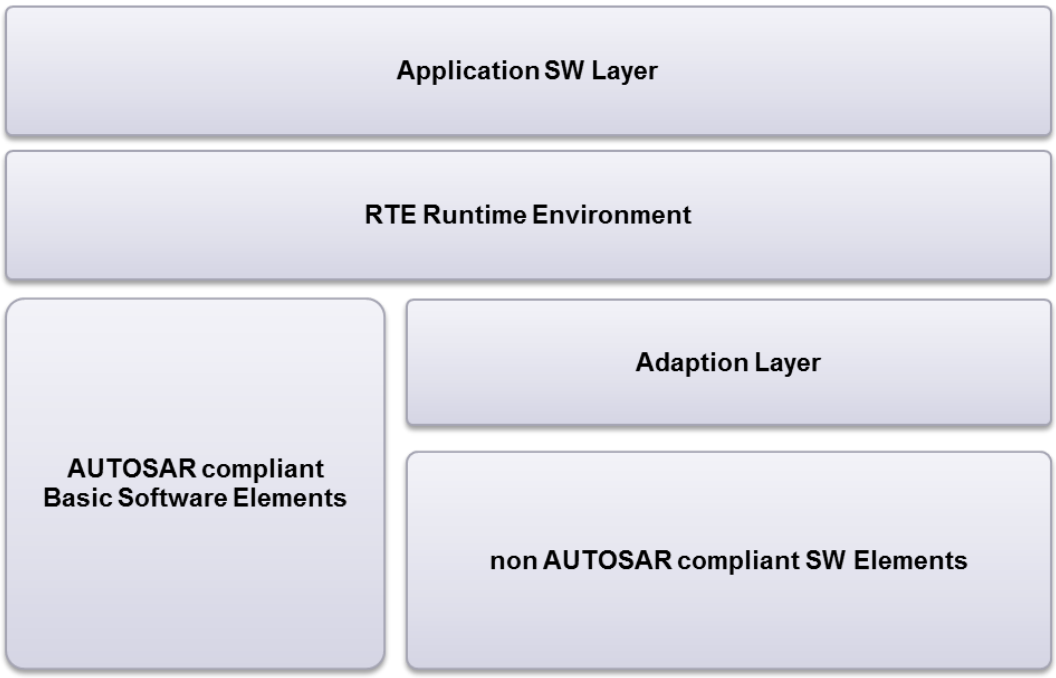

Figure 30: Outlook - SW-System Architecture

Software architectural design shall be described by using an appropriate notation.

Software safety mechanisms allocated to ASIL A functionality shall be described with informal notation.

Software safety mechanisms allocated to ASIL A or ASIL B functionality shall be described with informal or semi-formal notation.

Software safety mechanisms allocated to ASIL C or ASIL D functionality shall be described with semi-formal notation.

In addition to that formal notation can be used for software safety mechanisms allocated for ASIL A, ASIL B, ASIL C or ASIL D functionalities.

The architectural design shall be designed according to the following principles:

- Notations for software architectural design
- Principles for software architectural design

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

- newly developed
- reused with modifications
- reused without modifications

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

7.1.4.3 Software Safety Requirements

The software safety requirements are part of the software safety requirement specification that is derived from the Technical Safety Concept. The software safety requirements describe the technical solution allocated to software components that is needed to avoid violation of safety goals. Therefore the software safety requirement specification contains technical safety requirements that are allocated to software components.

The software safety requirement specification is defined as an input document for starting the safety relevant software development. To ensure a consistent implementation of the safety software system this input document shall be derived from the Technical Safety Concept specified on system level.

7.1.4.4 Software Unit Testing

Software Unit Tests shall be executed to verify that the software units fulfill the software unit design specifications and do not contain undesired functionality. To demonstrate that there is no unintended functionality structural coverage metrics shall be measured for safety relevant software units.

Test cases shall be generated (semi-)automatic based on the software safety requirements and the software unit design. It shall be able to add additional test cases manually.

Therefore an appropriate combination of the following verification mechanisms shall be used to verify the software units:

- Mechanisms for error detection at the software architectural level
- Mechanisms for error handling at the software architectural level
- Methods for the verification of the software architectural design

7.1.4.5 Qualification of SW-Components

Qualification of software components is used 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.

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

7.1.5 Further topics on hardware level

The following topics are part of the next iteration step of the hardware package.

7.1.5.1 Safety relevant Hardware Elements

Safety relevant hardware elements shall contain the following fields:

- failure modes
- failure rate distributed from the failure modes

Each hardware failure mode shall have a field to define if there is possibility of violation of the allocated safety goal in absence of the defined safety mechanism. (Yes/No)

Each hardware failure mode shall have a field to define if there is possibility of violation of the allocated safety goal in combination with an independent failure of another component. (Yes/No)

Each safety mechanism that is defined to avoid a failure mode of a safety related hardware element shall be allocated to the failure mode. It shall be possible to specify the failure mode coverage of a safety related hardware element in %.

It shall be possible to specify the coverage of latent failures for each failure mode in %.

7.1.5.2 Evaluation of safety goal violations due to random hardware failures

The target value of single-point fault metric and the latent point fault metric on item level based on the system design model and the hardware design specification shall be calculated (semi-) automatically based on the functional safety requirements given in the functional safety concept. They shall be described by quantitative safety requirements

The results of the evaluation of the hardware architectural metrics for a given version of the system design shall be part of the system design model. It shall be possible to provide traceability from the planned target value for hardware architectural metrics and the results given by the hardware analysis.

Diagnostic Coverage

The diagnostic coverage of safety-relevant hardware elements reached by allocated safety mechanisms shall be estimated with respect to residual faults and with respect to relevant latent faults.

The ISO 26262 part 5 Annex D contains tables that describe typical safety mechanisms of hardware elements with the typical diagnostic coverage. These tables can be used as a base for deriving the correct subset of safety mechanisms for the defined item.

7.1.5.3 Qualification of Hardware Components

Qualification of intermediate-complexity hardware parts and components for re-use shall be supported by the SAFE meta-model, by providing methods to specify re-use relevant component attributes as well as documenting the results of qualification (by analyses or testing).

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

7.1.6 Safety Validation

This topic shall be addressed in the next iteration step of the SAFE meta-model.

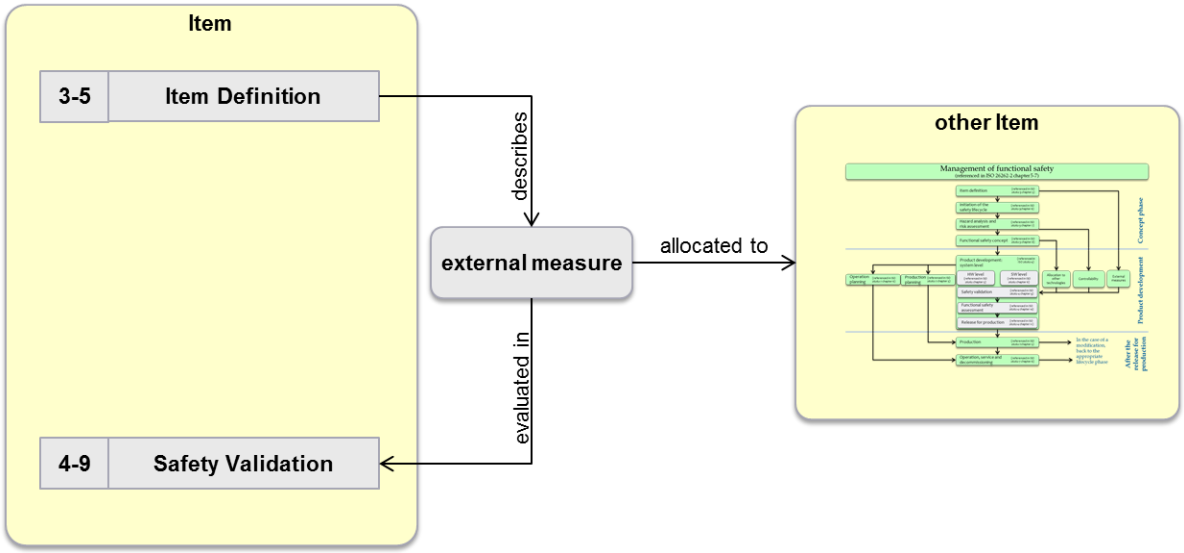

Figure 31: Validation of external measures

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

7.1.7 Process Activities

As described in chapter 7.1.2.5 activities can also be part of a safety measure. Therefore this chapter describes some of these process activities that can be used as safety measures to avoid systematical failures during development of safety relevant systems. Therefore the following topics shall be addressed during the next iteration step of the SAFE meta-model.

7.1.7.1 Management of safety requirements

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)

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

7.1.7.2 Configuration Management

It shall be possible to identify differences from the current architecture to former versions of the architecture. To ensure this an appropriate configuration management system shall be applied to identify the differences.

7.1.7.3 Traceability

Traceability is an essential requirement 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.

7.1.7.4 Impact Analysis

An impact analysis shall be carried out for

- each modification applied to the item or its environment
- each change request allocated to the item

The impact analysis shall

- address the effected elements of the item (e.g. operational situation, interface with the environment, software unit, hardware part...)
- allow identifying the differences between the item behavior before the modification/change request and after modification/change request
- describe the implication of the modification/change request to the functional safety aspects of the item and its environment

The results of the impact analysis shall be represented in the SAFE meta-model.

7.1.7.5 Change Management

If the safety analyses results a new hazard a change request shall be created according to the change management process defined in the ISO 26262 part 8.

Safety measures for the resolution of potential dependent failures shall be documented as a change request. The change request shall be handled according to the change management process defined in ISO 2626 part 8

The next iteration step shall address this topic and provide a solution how to handle change requests in the SAFE meta-model.

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

8 SAFE 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.b: Needs description to apply ISO26262 with architecture and component modeling

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

[7] SAFE-Project: SAFE-Partners, D3.1.2.b: 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, D3.2.2: Proposal for extension of meta-model for hardware modeling

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

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

[12] SAFE-Project: SAFE-Partners, D3.6.a: Safety Code Generator Specification

9 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.