



Contract number: ITEA2 – 10039



Safe Automotive software architecture (SAFE)

ITEA Roadmap application domains:

Major: Services, Systems & Software Creation

Minor: Society

ITEA Roadmap technology categories:

Major: Systems Engineering & Software Engineering

Minor 1: Engineering Process Support

WP 6, WT 6.1.1

Deliverable D.6.1.1 Methods for Assessment Activity Architecture Model (AAM)

Due date of deliverable: 30/06/2013

Actual submission date: 28/06/2013

Start date of the project: 01/07/2011

Duration: 24 months

Project coordinator name: Dr. Stefan Voget

Organization name of lead contractor for this deliverable: Continental

Editor: Hans-Leo Ross

Contributors: Philippe Cuenot, Thomas Peikenkamp, Thomas Wenzel, Maged Khalil, Alexander Rudolph, Juergen Lucas, Jörg Kemmerich, Stefan Voget, Karol Niewiadomski

Reviewer: all safe members.

Version	Date	Reason
0.1	2013-01-29	H.L. Ross – Extract chapter 7 from “Safe_D62_Guideline_draft_V07”
1.0	2013-06-28	S. Voget – editorial revision

1 Table of contents

1	Table of contents	3
2	Executive Summary	4
3	Introduction - General description of assessment activity/architecture model for functional safety development (AAM)	6
4	Assessment activity / architecture model for functional safety development (AAM)	8
4.1	ISO26262 as the starting point	8
4.2	Modell-based Development and Simulations	9
4.3	Hierarchical Error Analysis	10
4.4	Verifications by Safety Analysis.....	13
4.5	Safety Validation.....	16
4.6	Functional Safety Assessment	16
5	Common Metrics for evaluation.....	17
6	Outlook to the final paper	18
7	References	19
8	Acknowledgments	20

2 Executive Summary

The objective introduced in this document is to tackle the introduction of an information flow combining the work products requested in ISO26262 to a real engineering team. Based on this information flow, an assessment methodology for functional safety is specified, which accompanies the development process until safety validation, also taking into account the collaboration of OEMs and a tier one suppliers or tier 1 and tier 2 suppliers. Work-products and safety activities realized by the Safe Project and adequate measures are documented to allow seamless implementation in the different engineering disciplines.

This goal is realized in three tasks. The document in hand covers the first subtask. The three tasks are

1. Assessment model for functional safety process (this document with number D6.1.1)
2. Description of methodology for the information flow between activities (how we go from one element to the next (D6.1.2 planned to be published at 27.12.2013))
3. Application rules documentation (D6.2 planned to be published at 29.04.2014)

Task 1:

The first task aims at defining a methodology to achieve compliance with the ISO26262 standard. This goal is achieved by the delivery of an assessment activity/architecture model for functional safety development (AAM). The AAM provides a reference performing an assessment according to ISO26262. In particular the AAM consists of all safety activity and the data flow between them.

Based on analysis of the standard and required measures and considering the overall automotive supply chain, templates for verification planning are created. These templates show how the methods developed within the SAFE project support the safety activities mentioned in the verification plan.

This is done at all levels (incomplete list: HW component level, SW component level, system level), i.e. by defining the safety-related inputs/outputs that are required at each of the design stages.

Criteria and concrete measurements of a process (based on activities in the templates) are provided to verify e.g. the completeness of assessment.

The document in hand provides a draft AAM only. A final version will be included in the application rules documentation (D6.2).

Task 2:

The second task aims at defining reasonable sequences of AAM that are derived from the methods developed within the SAFE project.

For this a reference process for the model based development of safety relevant systems is identified. This reference process integrates and concatenates the methods and reflects the specific techniques developed in parallel in the first subtask.

Main references for this process are the domain specific languages EAST/ADL and AUTOSAR meta-models and methodologies.

Results from the ATESS2 and EASIS projects are taken into account in order to establish the reference process (SAFE Engineering Process, SEP). The parts of SEP are allocated to (analysis) levels of the EAST-ADL and AUTOSAR meta-models and methodologies.

Process steps with referenced work products are documented. This reference process focuses on portions that are of highest importance for ISO26262. The outcome of this work constitutes a reference for the application of the ISO26262 standard.

Task 3:

This activity makes available a series of guidelines for the use of the methods and tools developed in the preceding phases of the SAFE project. Starting from analysis of different industrial development scenarios, an exhaustive list of recommendations and guidelines is provided for the development of a safe automotive architecture. These application rules detail best practice, standard patterns, and concrete examples to document specific highlights of the safety standard applied in context of product development.

More specifically, the application rules address the following topics:

- Decomposition recommendations for effective design of safety mechanisms
- Compliance with architecture constraints and safety mechanisms and supervisor architectures
- AUTOSAR platform configuration for safety
- Inclusion of COTS in a system developed according to the ISO26262 standard
- Application rules for mixed criticality approach.

Furthermore, application rules for a mixed criticality approach contain decomposition recommendations and instructions how to use and integrate the software layer into a system using AUTOSAR basic software components in combination with the safety layer. It shows how to proceed to satisfy overall ASIL-D requirements despite the use of non ASIL-D components (AUTOSAR basic software components) such system using the safety layer concept.

3 Introduction - General description of assessment activity/architecture model for functional safety development (AAM)

The automotive specific functional safety norm ISO26262 [5] defines process requirements for functional safety-aware development in the automotive domain. It has high demands on process documentation and analysis. Some of the system characteristics important in the context of the ISO26262 are also relevant for non-safety related development and are therefore already addressed in conventional models. However, it is currently not clear how the development view and models necessary for safety documentation and analysis can and should be integrated in order to minimize modeling effort, to keep consistency between artifacts and to enable effective reusability and change management. Methods which allow demonstration of functional safety of automotive products according to ISO26262 are needed to be applicable to such an integrated model. While AUTOSAR [1] provides some technical prerequisites necessary to realize safety relevant systems, such as protection mechanisms or safe end-to-end communication, it is not yet clear how to use the AUTOSAR methodology within an ISO26262 compliant process.

The above challenges must be addressed if the European automotive industry is to cope with the increasing vehicle system complexity and a massive increase in safety-relevant functions (e.g. for driver assistance systems or electrical or hybrid vehicles). They can only be tackled effectively in a joint initiative that includes the complete automotive supply chain (OEMs, Tier 1's, Silicon vendors and tool suppliers) as well as academia that provide a significant research background in relevant fields. The European funding project SAFE addresses these challenges and speeds up the efficient development of safety critical features in cars. The objective is to enhance method, e.g. for defining safety goals and define development processes compliant with the ISO26262 standard for functional safety in automotive electrical and electronic systems.

The project started July 2011 and published the concepts, an integrated meta-model and an Eclipse based open source technology platform in early 2013. This document is a starting point for the process and assessment model [10].

Target of this document is a reference process model for functional safety assessment activities based on required functional safety activities according ISO 26262 and the description of the methodology. The methodology is based on results from the concepts (developed in WP3) and should deliver templates or guidelines to apply automated model-based verifications (in the meaning of ISO 26262).

The AAM is closely related to the result of the guideline (see description of task 3 above) and the collected methods linked in this guideline. The analysis of dependent failure is taken as an input for identification of the structure for AAM. The AAM provides at the end further content to the guideline.

Attached picture should show the dependency between other related projects and work-tasks within the SAFE project.

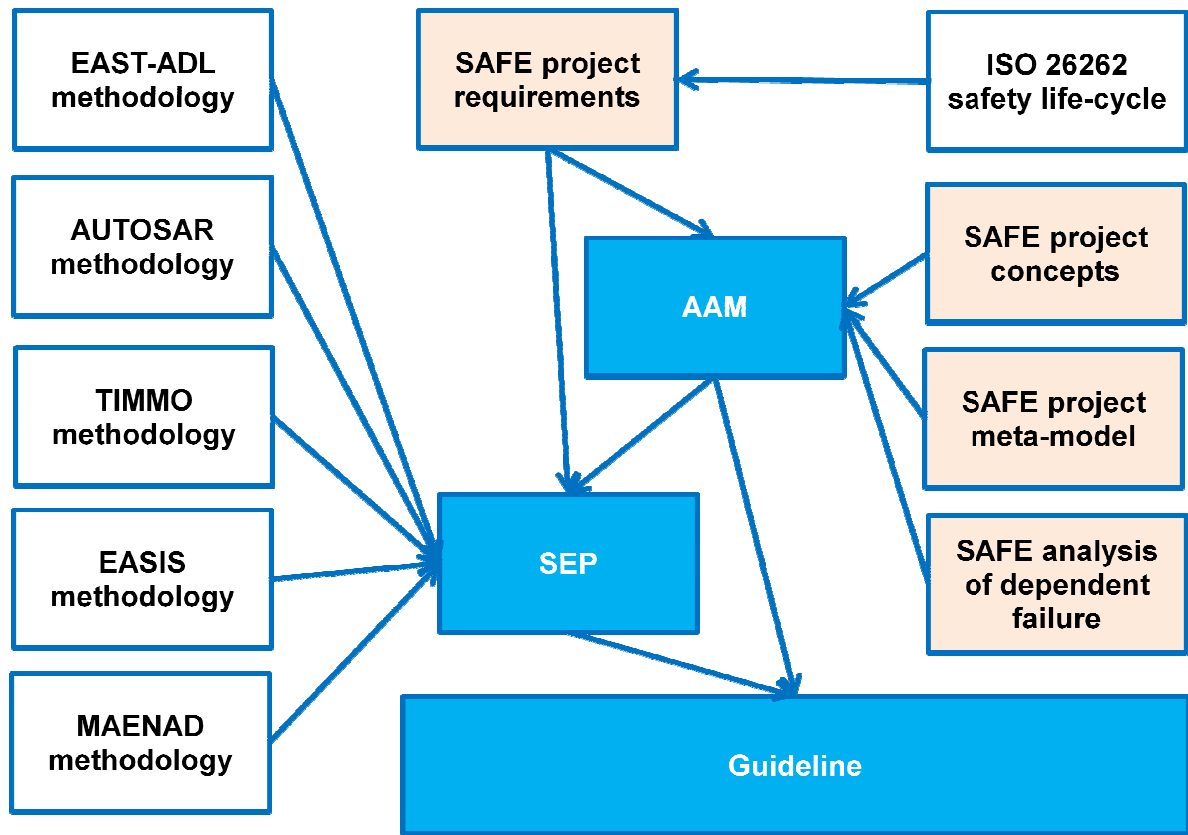


Figure 1: relationship between introduced tasks. White fields list material from external sources. Yellow fields list material from other work tasks of the SAFE project.

4 Assessment activity / architecture model for functional safety development (AAM)

4.1 ISO26262 as the starting point

ISO 26262 introduces 3 different confirmation measures:

- Functional Safety Audits
- Confirmation Reviews
- Functional Safety Assessment.

ISO 26262, Part 2, 6.2 defines the following:

6.2 General

Safety management includes the responsibility to ensure that the confirmation measures are performed.

Depending on the applicable ASIL, some confirmation measures require independence regarding resources, management and release authority (see 6.4.7).

Confirmation measures include confirmation reviews, functional safety audits and functional safety assessments:

- *the confirmation reviews are intended to check the compliance of selected work products to the corresponding requirements of ISO 26262;*
- *a functional safety audit evaluates the implementation of the processes required for the functional safety activities;*
- *a functional safety assessment evaluates the functional safety achieved by the item.*

In addition to the confirmation measures, verification reviews are performed. These reviews, which are required in other parts of ISO 26262, are intended to verify that the associated work products fulfill the project requirements, and the technical requirements with respect to use cases and failure modes.

The means of those measures are given in the following table from ISO 26262, Part2, Table 2.

Table 2 — Procedural requirements for confirmation measures

Topic	Confirmation review	Functional safety audit	Functional safety assessment
Subject for evaluation	Work product	Implementation of the processes required for functional safety	Item as described in the item definition in accordance with ISO 26262-3:2011, Clause 5
Result	Confirmation review report ^a	Functional safety audit report ^a in accordance with 6.4.8	Functional safety assessment report in accordance with 6.4.9
Responsibility of the persons that perform the confirmation measure	Evaluation of the compliance of the work product with the corresponding requirements of ISO 26262	Evaluation of the implementation of the required processes	Evaluation of the achieved functional safety Provision of a recommendation for acceptance, a conditional acceptance or a rejection, in accordance with 6.4.9.6
Timing during the safety lifecycle	After completion of the corresponding safety activity Completion before the release for production	During the implementation of the required processes	Progressively during development, or in a single block Completion before the release for production
Scope and depth	In accordance with the safety plan	Implementation of the processes against the definitions of the activities referenced or specified in the safety plan	The work products required by the safety plan, the implementation of the required processes and a review of the implemented safety measures that can be assessed during the item development

^a This report can be included in a functional safety assessment report.

4.2 Modell-based Development and Simulations

The development of products with support by model based engineering is already addressed in ISO 26262. Since ISO 26262 does not address any process iterations, it is a matter of interpretation to assure the specific requirements from ISO 26262. The iteration is addressed e.g. in the verification of architecture:

See ISO26262, Part 4

7.4.8.1 The system design shall be verified for compliance and completeness with regard to the technical safety concept using the verification methods listed in Table 3.

Table 3 — System design verification

Methods		ASIL			
		A	B	C	D
1a	System design inspection ^a	+	++	++	++
1b	System design walkthrough ^a	++	+	o	o
2a	Simulation ^b	+	+	++	++
2b	System prototyping and vehicle tests ^b	+	+	++	++
3	System design analyses ^c	see Table 1			

^a Methods 1a and 1b serve as a check of complete and correct implementation of the technical safety requirements.

^b Methods 2a and 2b can be used advantageously as a fault injection technique.

^c For conducting safety analyses, see ISO 26262-9:2011, Clause 8.

See ISO 26262, Part 6

7.4.18 The software architectural design shall be verified in accordance with ISO 26262-8:2011, Clause 9, and by using the software architectural design verification methods listed in Table 6 to demonstrate the following properties:

- a) compliance with the software safety requirements;
- b) compatibility with the target hardware; and
- NOTE This includes the resources as specified in 7.4.17.
- c) adherence to design guidelines.

Table 6 — Methods for the verification of the software architectural design

Methods		ASIL			
		A	B	C	D
1a	Walk-through of the design ^a	++	+	o	o
1b	Inspection of the design ^a	+	++	++	++
1c	Simulation of dynamic parts of the design ^b	+	+	+	++
1d	Prototype generation	o	o	+	++
1e	Formal verification	o	o	+	+
1f	Control flow analysis ^c	+	+	++	++
1g	Data flow analysis ^c	+	+	++	++

^a In the case of model-based development these methods can be applied to the model.
^b Method 1c requires the usage of executable models for the dynamic parts of the software architecture.
^c Control and data flow analysis may be limited to safety-related components and their interfaces.

It is up to the tailoring of the lifecycle if the focus is more on simulations or on prototyping. Simulations are generally seen as a method for verification. In model-based development, it is a basic requirement to verify the correctness of the model used for the simulation, before the model could be used to verify the prototype or the realized product or characteristics, behavior or parts of it.

4.3 Hierarchical Error Analysis

Functional modeling and safety analysis [9] are two important aspects of safety-critical embedded systems. However, they are often conducted separately. Following SPES2020 [11] both aspects are called perspectives. Figure 2 introduces further perspectives, but we will concentrate on the functional perspective and the ISO26262 perspective. Each cell in the matrix of abstraction levels versus perspectives provides a view.

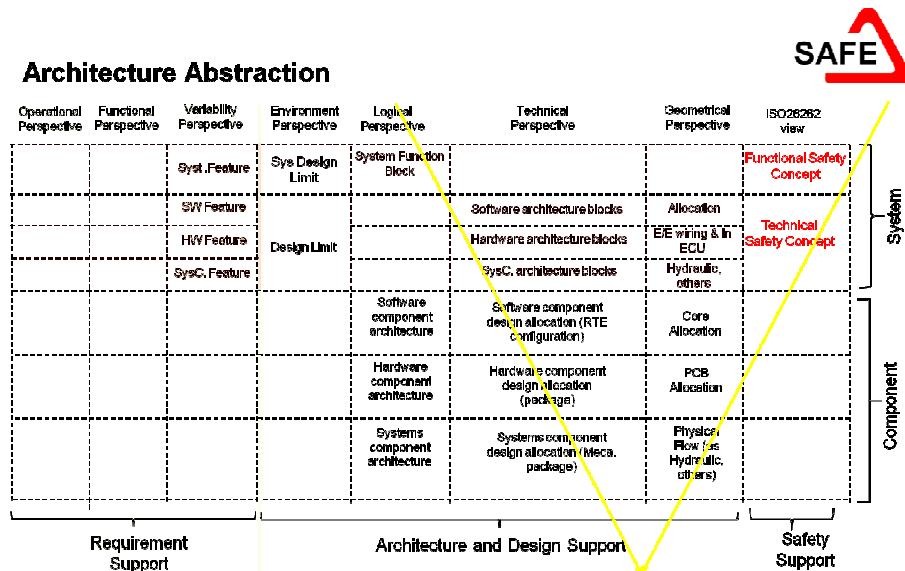


Figure 2: Abstraction levels versus Perspectives

A system model that takes failures into consideration is crucial in ensuring that safety is considered throughout the development process. It will offer the following benefits.

The model will allow engineers to be knowledgeable about the undesirable conditions and system failures and to understand how the behavior of the system is affected by these failures. It will help them to understand the interaction between the software and other system components. The model will also identify the components that are responsible for the system functions that were previously identified by the hazard analysis. These components should then be given special attention in the system development process.

As shown, system, hardware and software modeling is usually done by hierarchical modeling – from an abstract concept to a detailed design. The safety analysis has to be adapted to these abstraction levels, too.

In a safety-critical system, every major failure is classically represented by a fault tree. Each fault tree describes how the individual fault components combine to result in an undesirable system behavior or catastrophic failure. The root of a fault tree represents the major failure or the most general failure. As we go down the tree, the nodes represent more specific or detailed faults. Thus, a fault tree describes the catastrophic event in terms of its causal factors or faults in a hierarchical fashion.

Figure 3 shows an example of architectural layers corresponding to a fault-tree analysis. The analysis of a fault tree could be done either qualitatively or quantitatively. A model transformation from a fault tree model to another safety analysis model (e.g. reliability block diagram) should be discussed in further studies. A mixed safety modeling strategy will combine advantages of all safety models and reduce their drawbacks.

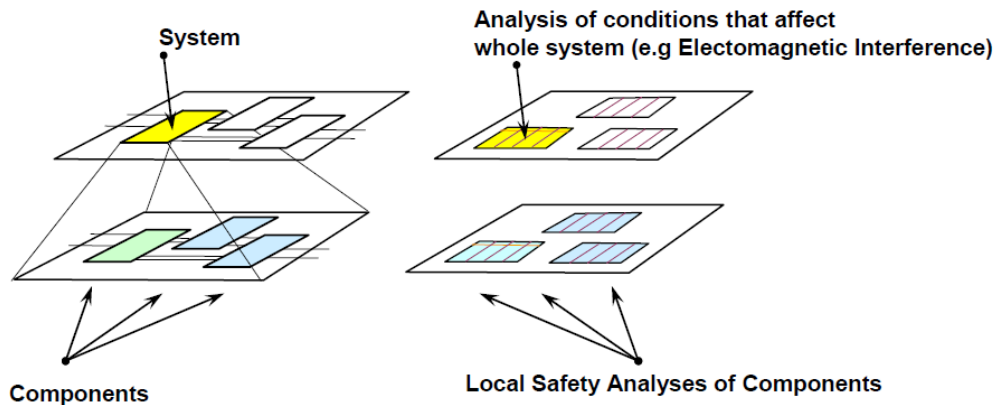


Figure 3: Abstraction between system- and component-layer

In the abstraction levels of EAST-ADL, safety analysis models are applicable. An early safety evaluation will be performed to demonstrate the diagnosis coverage using model based FTA analysis and error hazard occurrence, based on the error models propagation between safety critical and safety relevant functions/features.

Especially, the improvement and the definition of interfaces between requirements, architecture, design and its verification during development via Safety Analysis will be highlighted. The definition of the interfaces, the identification of functional and non-functional requirements, technical requirements, design limitations and preliminary architectural assumptions will be clarified, and assigned to the different hierarchical levels such as vehicle, system or component. A secondary target is to avoid or mitigate systematic failure at the interfaces to supplier and customer.

Insure a seamless handling of safety requirements within overall requirement management providing relevant coverage and impact synthesis for the safety case documentation. Avoid inefficiency of document oriented traceability by introducing model centric requirement management in design activities (refer to dysfunctional modeling improvements).

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

By studying all relevant accidental events (that have been identified by a preliminary hazard analysis, a HAZOP, or some other technique), the ETA can be used to identify all potential accident scenarios and sequences in a complex system.

The purpose of hazard analysis is to examine the system and determine which components of the system may lead to a mishap. There are two basic strategies to such analysis that have been termed inductive and deductive. Essentially, inductive techniques, such as event tree analysis and failure modes and effects analysis, consider a particular fault in some component of the system and then attempt to reason what the consequences of that fault will be.

Deductive techniques, such as fault tree analysis, consider a system failure and then attempt to reason about the system or component states that contribute to the system failure. Thus, the inductive methods are applied to determine what system states are possible and the deductive methods are applied to determine how a given state can occur.

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

The Fault Tree Analysis (FTA) [2] is a top-down method that systematically breaks down hazards to their causes. The result is then visualized in a tree structure. Thus, the FTA rather helps systematically analyzing hazards than to detect them.

The first step in the FTA is to define the system bounds. This includes the definition of the hazard, the events the analysis has to take into account, the physical system bounds, and the initial system state. For example, there is a huge difference between a car in city traffic and a car on the autobahn being analyzed. Once the system bounds are defined, the occurrence of a hazard (top-event in an FTA) is iteratively decomposed into its causes. To visualize this process a standardized graphical notation (IEC 1025 standard) is used.

The Failure Mode and Effects Analysis (FMEA) [2] represent a preventive safety analysis approach. It is utilized to identify and assess potential failure causes as early as possible, e. g. during the early design phases. This helps to prevent control and further failure costs during the production or even the operation phase. Furthermore, a systematic failure analysis approach inhibits further repetition of design faults in other products.

FMEAs should be applied in early product life cycle phase's, e.g. the concept and design phase, since it proved to be most beneficial for a cost–benefit analysis. The sooner a fault is discovered, the cheaper its correction can be.

A reliability block diagram (RBD) is a diagrammatic method for showing how component reliability contributes to the success or failure of a complex system. RBD is also known as a dependence diagram (DD).

A RBD or DD is drawn as a series of blocks connected in parallel or series configuration. Each block represents a component of the system with a failure rate. Parallel paths are redundant, meaning that all of the parallel paths must fail for the parallel network to fail. By contrast, any failure along a series path causes the entire series path to fail.

An event tree analysis (ETA) is an inductive procedure that shows all possible outcomes resulting from an accidental (initiating) event, taking into account whether installed safety barriers are functioning or not, and additional events and factors.

Final objective of the whole set of analysis methods is to allow continuous verification while walking through the development cycle and involving the different development teams.

This leads to a set of qualitative and quantitative measures:

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

4.4 Verifications by Safety Analysis

Safety analysis methods are basically just special methods for verification. Particularly the different FMEA methods support the verification of systems.

A System-FMEA primarily supports the verification of requirements and their allocation to functions as well as to logical or technical elements. A Design-FMEA questions the correct interpretation of the design or implementation, due to the criticality of the failure effect a risk based approach for verification measure could be provided. This is usually started with the design concepts in the later iterations it incorporated to the realized product. The Design-FMEA primarily supports the design verification and is finalized by a Design Review of a cross-functional team. This has also strongly points to the so-called Toyota-FMEA (DRBFM -Design Review Based on Failure Modes). Usually with a Process-FMEA the production process should be analyzed. Formally it would be possible to any process to be analyzed by this method, see also the chapter "Process Verification." In any FMEA standard requires a final review to confirm the goal achievement of the analysis. A final review of the FMEA is formally part of any FMEA method.

The following verifications can be supported by safety analysis:

Completeness of the relevant safety goals

Primarily safety goals are as follows: "Avoid that a possible malfunction of the item could possibly cause harm." Any malfunction can be structured in a System-FMEA as effects of systems failure. Any credible effect of a systems error could be considered as a malfunction that violates a safety goal. If all potential systems error or failure and their effects are considered and no effects lead to any other safety relevant effect (top-failure) than the defined Safety Goal, the completeness could be demonstrated.

Completeness of relevant functions within the boundary of the item

This analysis is based on the functional networks of the VDA FMEA. However, automated testing would be much more effective by using architecture tools. Checking may take place in any horizontal level of abstraction. Since a System-FMEA could be performed on any level of abstraction any completeness of functions within a element boundary could be analyzed on software-, hardware- component level and even within silicon, such as semiconductor. It is comparable with branch checks in SW-units, it analysis on a similar way that inputs and outputs within a boundary are complete connected. The basic principle of the analysis is to identify the signal chain, which was developed by Robert Lusser and had been described over 80 years ago.

Consistence and completeness of dedicated functions from a higher level derived to a lower level of horizontal abstraction. (Verification of function decomposition)

It is comparable with the analysis of completeness of functions within a boundary in previous chapter. This analysis does extend the analysis and compares the already approved on a higher level of horizontal abstraction with the same representation of the function on a lower level. Depending on the criterions which were added in the lower level of abstraction, their completeness could be also evaluated. It could be based on the function network of a VDA FMEA, but as well as in the previous analysis; better transparence could be achieved by architecture tools. A signal chain on system level could be compared with a signal chain which is allocated to software or hardware. In combination with previous analysis also the hardware-software-interface could be analyzed, due to separation of an element on higher level into 2 or more elements on lower level of abstraction. Further abstractions within the same level of abstractions could be analyzed for completeness and correctness, by adding information about environmental impacts, power supply, voltage, EMC, common usage of resources. Due to those verifications the analysis of dependent failure could be supported as described in chapter 5.1.9.

Consistency check of the interfaces (Verification of product decomposition)

The VDA FMEA by the structure networks, the interfaces for the entire product structure are described. Here there is the challenge that functional and technical interfaces are not always congruent. By comparing functional, logical and technical structure between each other and between structure and between their interfaces in different horizontal level of abstractions could provide information about completeness and consistency of those structures and their interfaces. Also here architecture tools and possible routines are much more effective than static structure within a VDA FMEA.

Completeness of the considered malfunctions (failure, error or fault modes)

Especially during deductive analysis, it is important to argue a certain completeness of considered malfunctions. Basically any characteristic of a function or an element could fail and so having potential impact to malfunctions. Any identified error of goods could be considered as an argument to add measure to improve measures during development and for implementing in the product to improve non-functional requirements such as safety, availability or reliability. Since we consider that system elements are always have to correct interact to perform a required function, error modes per functions could be defined. One way could be to apply law from DeMorgan, it converted negated "or" in "and" gates. A VDA FMEA failure analysis, which is seen as the third step of that

FMEA approach after product and function decomposition you determine for any function independent from the level of abstraction possible malfunctions. For the verification of the safety requirements, it is first of all necessary to determine completeness related to the allocated function. That means any required characteristics and any required technical behavior and their characteristics could deviate from their intended or required state. By pure information completeness of considered malfunction could be achieved that any information could be wrong. It is recommended by automated checker to consider in addition to that that the information could not be available at the required point. These could provide a completeness argument for the considered malfunctions. In a more deep analysis the following malfunctions could be considered:

- no function
- unexpected function (crosstalk from other systems)
- systematically falsified information or function (for example, signal drift)
- sporadically or improper function or unexpected information
- module or element was not executed addressed or considered
- function or element does not run continuously or is not considered continuous (uninterrupted operation is not, oscillations)
- Wrong Timing

These questions are the basics for the most deductive methods such as HAZOP and Fault Tree Analysis. In essence, they are comparable with the tables in Part 5, Annex D of ISO 26262, which are the basis for the diagnostic coverage. Even in Design-FMEA such analysis is considered to be evaluated sufficiently or necessary coverage of adequate design assurance measures.

Completeness of the considered single point malfunctions (failure, error, faults)

This is the classic domain of FMEA; here all possible malfunctions of an appropriate level to consider whether they can propagate to higher level up-to a safety goal.

Complete view of error combination up-to the order of 2 (e.g. double faults)

Multiple point errors always make high permutations based on their factors, therefore, even in a simple system the analysis of multiple faults is a challenge. By considering safety mechanisms as a barrier preventing errors from propagation, any fault could be considered as a single point fault related to the barrier or safety mechanism. For the safety goals higher than ASIL C, also fault combinations have to be controlled, depending on their probability of occurrence at the same time. If a safety mechanism is an independent measure to the dedicated safety related function, an error of the safety mechanism could not lead to a failure of the safety related function, so that these errors could be considered as a double fault. As a consequence any secondary independent function, that could not influence a safety goal by itself, have at least a distance of 2 related to their fault propagation, it means it is at least a double fault related to the considered safety goal. Due to classifying functions into secondary independent functions related to the safety goal, their errors could be considered as double failure.

Correctness of the safety goal itself

In case of considering completeness of hazardous events, the propagation of potential malfunctions of system or item to those hazardous events could be analyzed. In case of complete effect of any malfunction to the considered hazardous event it could be used as an argument. In the domain of event-tree-analysis (ETA) even the combination with relevant driving situations could be considered. Completeness could be argued in case of considering completeness of those driving situations.

4.5 Safety Validation

Validation is not in the scope of Safe. Therefore a mayor step before the assessment of functional safety has not been considered. It is a strong recommendation for future activities.

4.6 Functional Safety Assessment

ISO 26262 precise the requirements for Functional Safety Assessments in Part 4 chapter 10:

10 Functional safety assessment

10.1 Objectives

The objective of the requirements in this clause is to assess the functional safety that is achieved by the item.

10.2 General

The organizational entity with responsibility for functional safety (e.g. the vehicle manufacturer or the supplier, if the latter is responsible for functional safety) initiates an assessment of functional safety.

10.3 Inputs to this clause

10.3.1 Prerequisites

The following information shall be available:

- safety case in accordance with ISO 26262-2:2011, 6.5.3;*
- safety plan (refined) in accordance with 5.5.2, ISO 26262-5:2011, 5.5.2 and ISO 26262-6:2011, 5.5.2;*
- confirmation measure reports in accordance with ISO 26262-2:2011, 6.5.5;*
- audit report if available in accordance with ISO 26262-2:2011, 6.5.4; and*
- functional safety assessment plan (refined) in accordance with 5.5.5.*

10.3.2 Further supporting information

None.

10.4 Requirements and recommendation

10.4.1 *This requirement applies to ASILs (B), C, and D of the safety goal: for each step of the safety lifecycle*

in ISO 26262-2:2011, Figure 2, the specific topics to be addressed by the functional safety assessment shall be identified.

10.4.2 *This requirement applies to ASILs (B), C, and D of the safety goal: the functional safety assessment*

shall be conducted in accordance with ISO 26262-2:2011, 6.4.9 (Functional safety assessment).

10.5 Work products

10.5.1 *Functional safety assessment report resulting from requirements 10.4.1 and 10.4.2.*

The safety case is considered as an input of the Functional Safety Assessment, but the “Functional Safety Assessment Report” is an input for the Safety Case (further details about safety case generation see WP3). It shows that activities should be performed in parallel. After a successful run of a functional safety assessment, ISO 26262 defines the “Release for Series Production” in its chapter 11.

Due to permanent need of human interactions for analysis, verifications, design decisions, validations etc. within “Safe” only partially the “Functional Safety Assessment” could be considered. Some of the described methods for verification give already the hint, that for complete Functional Assessment a complete tailored safety Lifecycle need to be considered, including human influences.

5 Common Metrics for evaluation

For each work product, a metric *performance* will be setup rating how well the expectations given in the work product description have been met.

- Level 5:** beyond expectations described in the Full Project Proposal and evaluation criteria
- Level 4:** expectation from Full Project Proposal and good level evaluation criteria met
- Level 3:** expectations not fully met or some evaluation criteria not reached sufficient level but significant improvement achieved
- Level 2:** no significant improvement achieved or some evaluation criteria are rated incomplete
- Level 1:** negative impact (performance degraded) and all evaluation criteria are incomplete

This evaluation will be crossed with a metric *industrial interest* qualifying the relevance of the method (or tool or methodology, respectively) covered by the corresponding evaluation scenario.

- Level 4:** Interesting for evaluation scenario and ready for application in the field
- Level 3:** Interesting for evaluation scenario but needs to be slightly matured for application in the field
- Level 2:** Interesting for evaluation scenario but needs to be significantly matured for application in the field
- Level 1:** Not of interest for the specific evaluation scenario but interesting anyway for application in the field (not considered further for project evaluation – no detailed evaluation result available)
- Level 0:** Out of scope of evaluation scenario, not of interest for application in the field.

Thus, a graphical representation can be provided for each evaluated work product which gives an interpretation of the industrial potential of the latter.

		Performance				
		1	2	3	4	5
Interest	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
	0	0	0	0	0	0

6 Outlook to the final paper

Continuing in the story described so far, the final paper will introduce safety analysis techniques embedded in the EAST-ADL modeling layers. A main table offers the outcome of different safety modeling techniques using the defined EAST-ADL levels. Former work already mapped safety modeling techniques to the abstraction layers of EAST-ADL. But, a detailed mapping of the safety analysis methods had not been given before. Therefore, it is still unclear neither which safety analysis techniques are used in which abstraction layer of EAST-ADL nor which relations exist between those different modeling techniques.

The new approach discusses some of the related tools and techniques specifically proposed for automated safety analysis. Beside others the following two automatic analysis methods will be handled:

- **Automatic Fault Injection and Model Extension**
In a first step one defines failure modes. After that, the user can automatically inject the failures in the system model to create a new extended model. The extended system model adds degraded behavior to the original system corresponding to the failure modes defined. This model can then be used for safety assessment of the system.
- **Automated Fault Tree Analysis**
A significant advantage of an automated analysis tool is that it removes the burden of manually creating fault trees once the system and the fault model are specified. This ensures that the system and safety engineer work off the same models and assumptions.

7 References

- [1] AUTOSAR; www.autosar.org
- [2] Börcsök, Josef, Funktionale Sicherheit, Hüthig Verlag, Heidelberg, 2006.
- [3] EAST-ADL Association; www.east-adl.info
- [4] IEC 61508 (1998). International Electrotechnical Commission (IEC): Functional Safety of Electrical /Electronic / Programmable Electronic Safety-Related Systems.
- [5] ISO 26262 (2012) International Organization for Standardization Road Vehicles Functional Safety.
- [6] P. Liggesmeyer and M. Rothfelder. Improving System Reliability with Automatic Fault Tree Generation. In Proc. 28th International Symposium of Fault-Tolerant Computing (FTCS), 1998, Munich, Germany.
- [7] O. Ljungkrantz; Case study about ISO26262 in the EAST-ADL / Autosar context, Experiences with ISO26262-2013 conference, Munich, 2013.
- [8] MAENAD; www.maenad.eu
- [9] Place, Patrick R.H., Kang, Kyo C, Safety-Critical Software: Status Report and Annotated Bibliography, Technical Report, CMU/SEI-92-TR-5, Software Engineering Institute, Pittsburgh, 1993.
- [10] SAFE; www.safe-project.eu
- [11] Technische Universität München, SPES 2012 Deliverable D1.1.A, Initiale Modellierungstheorie, München, 2009. Download: <http://spes2020.informatik.tu-muenchen.de/results/D-1-1-A.pdf>
- [12] Technische Universität München, SPES 2012 Deliverable D1.2.A-1, Abstraction Layers - Motivation and Introduction of a System of Abstraction Layers for Embedded Systems, München, 2009. Download: http://spes2020.informatik.tu-muenchen.de/results/D-1-2-A-1_initial_abstraction_layers.pdf
- [13] ZVEI, - Zentralverband Elektrotechnik- und Elektronikindustrie e.V., Nationale Roadmap Embedded Systems, Frankfurt, 2009.

8 Acknowledgments

Many thanks to the project partners from the SAFE project.

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 E16095. The responsibility for the content rests with the authors.