Decision Hierarchy and Architectural Patterns

Description

This white paper gives “state of the art” and results regarding Decision Hierarchy and Architectural Patterns for ADS. It consists of excerpts from the final “state of the art” report and final result report in WP5 “Decision Hierarchy and Architectural Patterns” of the ESPLANADE (Efficient and Safe Product Lines of Architectures eNabling Autonomous DrivE) project. Qamcom was one several participants. The work was mainly financed by VINNOVA (Swedish Agency for Innovation) through the project ESPLANADE; which was run during 2016-2020.

Qamcom is now involved in a new research project: , SALIENCE4CAV - Safety lifecycle enabling continuous deployment for connected automated vehicles. This is a continuation of ESPLANADE . The focus of the projects has been to support a safe implementation of self-driving vehicles and contribute to one of SAFER's main research questions - how to verify and validate assisted and automated systems in cooperation. This third project, which started in early 2021, will continue the work on how to ensure safety and how to use existing safety standards, e.g. ISO26262. The focus will be to enable iterative development for safety-critical products, that is, development of products and enabling of upgrades much more often. Further topics include development of methods to use for the design and assurance of safety-critical automated driving systems (ADS) for connected automated vehicles (CAV), enabling the use of iterative development and continuous deployment for ADSs

Revision History

Iteration	Date	Paragraphs affected	Change information
1	2020-09-03	All	New white-paper format with selected contents from WP5 reports.

Authors

Name	Contact	Company
Anders Cassel	Anders.Cassel@qamcom.se	Qamcom Research and Technology AB
Carl Bergenhem	Carl.Bergenhem@qamcom.se	Qamcom Research and Technology AB

Contents

1. Introduction

One objective of the ESPLANADE was to propose and apply a systematic methodology for deriving a functional and logical architecture for automated driving systems (ADS). By using a systematic methodology, a good description of the function and system can be created that is consistent, complete, and provides the necessary information for carrying out the safety analyses that will produce the safety requirements.

The white paper presents the approach for defining a methodology to derive architectural structures for ADS. Accompanying examples illustrate the proposed design patterns. It is based on excerpts from [25].

2. Acronyms

ADAS	Advanced driver assistance system
ADS	Automated driving system
AI	Artificial intelligence
ATS	Automated transportation system
CPS	Cyber-physical system
E/E	Electric and electronic
HARA	Hazard analysis und risk assessment
JDVS	Joint driver-vehicle system
ODD	Operational design domain
RAS	Robotic and autonomous system
Reference architecture	Architectural guidelines and principles

3. State of the art of Automotive Architectural Structures and Design Patterns for automatic driving

In this section we give relevant State of the Art (SOTA). The section is based on excerpts from [24]. The intention is to provide an overview of the architectural solutions presented for automated vehicles and the reasoning and motivations behind.

Most Advanced Driver Assistance System (ADAS) architectures proposed in literature consists of three key functional components: Perception, decision and actuation/manipulation, see Figure 1, [18]. ADAS architectures presented in literature build upon this structure but with a large variety in the representation on a more detailed level and in the decomposition into smaller elements. Different architectural views (functional, logical and physical/technical) complements each other with information on how the architecture implements the ADAS functions [19].

The perception element refers mainly to the ability to collect information from the external environment. It utilizes various sensing sources and the extracts relevant data about the vehicle's surrounding for understanding the situation in which it is operating, see Figure 2.

The decision element processes the perception data to create awareness of the traffic situation, to identify potential threats, to plan the vehicle movement, and provide appropriate control demands.

The vehicle actuators execute the control demands and thereby brings the vehicle to the desired state or destination.

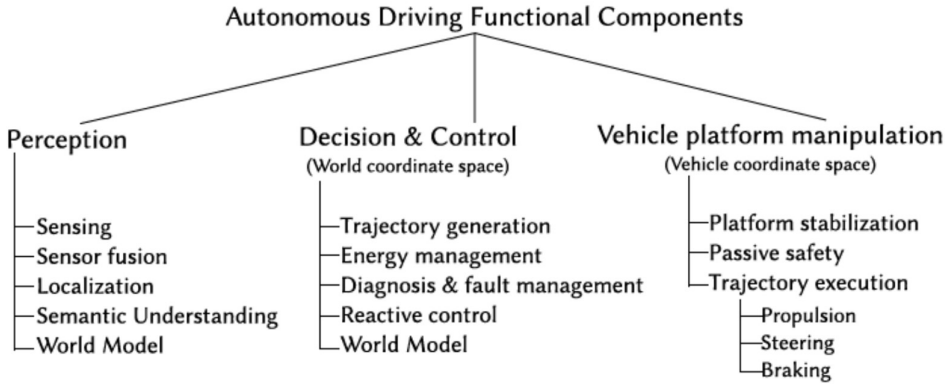

Figure 1. AD key functional components

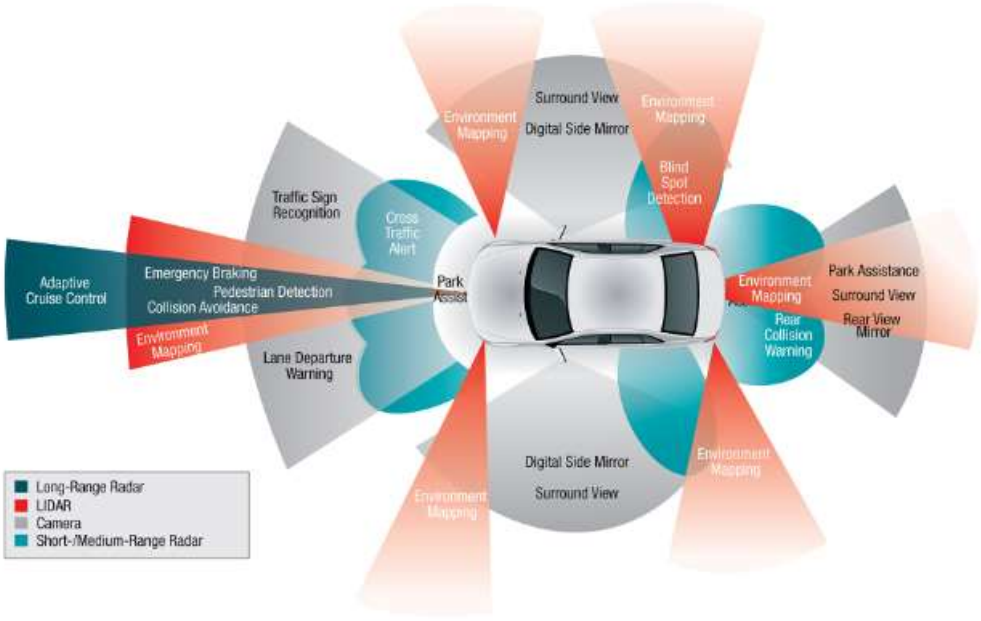

Figure 2. Typical sensor configuration used for various ADAS functions and AD functionality.

3.1. Automatic driving – functional aspects

For ADS and ADAS the driver and vehicle are a joint driver-vehicle system (JDVS) that to different extent interacts for controlling the movement of the vehicle. From a safety perspective, it is crucial to maintain control of the JDVS at every instant and in every situation. In traditional driving, the driver is responsible for the driving task in terms of perceiving the situation and making decisions on how to maneuver. The strategical and tactical decisions are the driver's responsibility while the vehicle controls the operational part. An ADAS supports the driver and contributes in a limited aspect to the control of the vehicle by replacing or augmenting the driver's actions to avoid critical situations. An

ADS operates without a human driver in the control loop and has the complete responsibility for the driving task and continuously operates the vehicle by issuing control demands (steering/braking). Thus, the responsibility split of the JDVS control is different between an ADS and an ADAS.

Transitions between automated driving and manual driving needs to be considered carefully. If a safe transition cannot be ensured, the system must be able to handle the situation by continuing to operate the vehicle while awaiting driver´s availability and readiness to take control. If needed the vehicle needs to be taken to a safe state/stop. Furthermore, while driving in automated mode, the driver interface needs to provide the passengers with appropriate information in order to assure a feeling of comfort and safety in all driving situations and, especially in the early stages of technology rollout, to increase acceptance towards automatic driving functions.

3.2. Automatic driving – safety aspects

While for ADAS functional safety relates mainly to commission failures (e.g. false brake interventions), both omission and commission failures are safety critical for an ADS. An ADS needs to be fault tolerant because the driver might to be able to take over control in case a system failure. The system needs to be able to continue operating, eventually with degraded functionality until a safe state is reached. Safe states can be a safe full stop of the vehicle or a successful handover to the driver.

An ADS needs to be able to detect failures and adapt its operational capability accordingly.

3.3. Automatic driving - architectural aspects

Reference architectures provide architecture principles and guidelines, without details and implementation specific solutions. Section 3.5 provides an example for a proposed reference architecture for commercial vehicles.

Adaptations of existing ADAS platforms have been made for demonstrating autonomous driving in specific situations and under certain restrictions and have successfully provided a proof of concept, However, the great difference in capability and safety needs between ADAS and ADS will impact the architecture more than just improvements and evolvements from existing ADAS platforms.

The work in ESPLANADE is primarily focused on the development of safety requirements on the decision-making and perception elements and how these affect the functional and logical architecture for an ADS.

A structured method for analysis of the ADS functions will be applied in order to propose an architectural design that meets the requirements on functional performance and safety. The primary aim is to define a functional and logical architecture in which the sub functions and interfaces between the different elements are clear. How to align the ADS decision hierarchy to the operational capability to assure that the dynamic driving task is executed in a safe way regardless of system state (normal or degraded mode) will also be considered.

3.4. Design Prerequisites for ADS

An automated driving system (ADS) architecture uses a similar overall architectural structure as an ADAS but the fundamental difference in terms of responsibility for the driving task impacts the architecture in terms of elements and functions for perception and decision making as well as other requirements and solutions for the driver interface.

3.5. A reference architecture for automated driving for commercial vehicles

The main purpose for trucks on the road is business-to-business (B2B) operations that require good operational margins. Automation allows decreasing operational costs and connectivity adds additional value by allowing a deeper integration of the vehicle fleet into a logistic system. Automated driving systems, connectivity and the introduction of electro mobility demands for moving away from today's traditional signal-based ECU-orientated architecture and the introduction of a service orientated architecture (SoA).

As an example for a reference architecture, this section is a summary of [20], which presents a reference architecture for commercial vehicles that facilitates extensibility and variability by separating software from physical hardware and introducing a horizontal layer of application software and thus allowing an easy integration of automatic driving functions.

Figure 3 illustrates the proposed reference architecture and we will consecutively explain the structure.

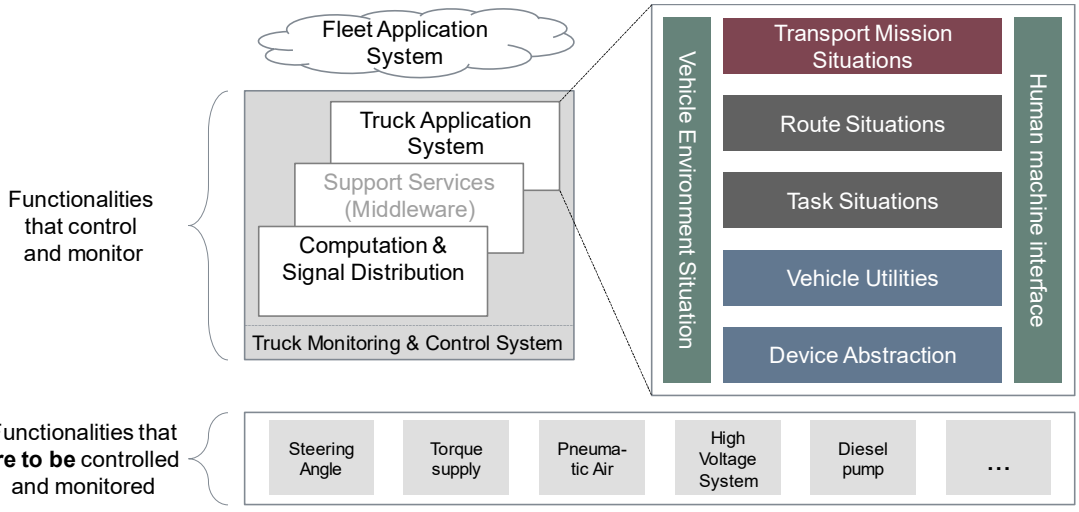

Figure 3: Reference architecture for commercial vehicles

A first step is the separation of functionalities that control and monitor from functionalities that are to be controlled and monitored. The functionalities that are to be controlled and monitored are related to physical process, for example the motor that adjusts the steering angle, a compressor providing pneumatic air or a power converter supplying a high voltage system. These functionalities include no or a very limited level of intelligence. All intelligent functionalities are part of the truck monitoring and control system.

A second step is the partitioning of the truck monitoring and control system into three major product line entities. The argumentation behind this step lies in the nature of the functionalities that might

be the conversion of an analogue value to a digital value, forwarding a signal to complex software services. The computation & signal distribution entity focuses solely on offering interfacing, routing, and computing capacities to other services. The truck application system hosts the various application software entities that define the higher intelligent functionalities of the vehicle. A major advantage of this partitioning is the acknowledgment that cycle-times for developing functionalities differ significantly between computation & signal distribution and the truck application system. Furthermore, the clear separation allows for a better refinement of safety requirements to either hardware or software [21]. Support services, or a kind of middleware, connect the two worlds.

The third step is to define a horizontally layered application layer for the truck application system. Each horizontal layer raises the abstraction level from physical devices to strategic mission planning.

Device abstraction and vehicle utilities define the operational functionalities. Device abstraction entities contain information and properties of actual mechatronic devices. They represent the functionalities that are to be controlled and monitored. Device abstraction entities do not have direct interaction. Instead, the vehicle utilities defined vehicle wide services that combine several device abstractions. For example, a "Window Cleaning" utility uses the "Wiper Motor" and "Washer fluid pump" device abstractions.

Task situations and route situations define the tactical functionalities. The purpose of the task situation layer is to offer automation and coordination for tasks that are traditionally performed by a human driver. The functionalities described in the route situation layer describe tasks with a longer planning horizon. Figure 4 illustrates how tactical functionalities are broken down into functionality domains (FD) that contain global tasks and functionality areas (FA) with finer defined tasks.

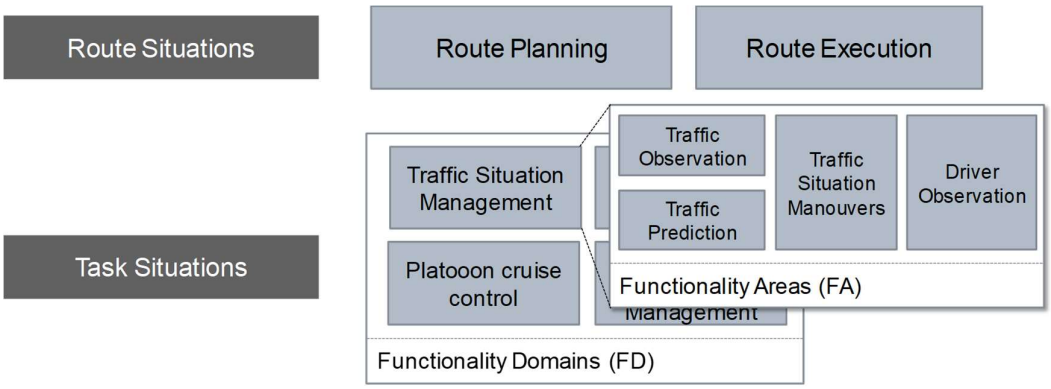

Figure 4: Break down of tactical functionalities into functionality domains and functionality areas

The transport mission situation layer contains the most abstract functionalities and forms the strategic level. The layer focuses on the transport mission and offers functionalities related to for example optimization of operation cost and delivery speed.

In a fourth step environmental awareness is added to the architecture. In contrast to the previous layers, the vehicle environmental situation layer is vertical and can therefore make use of software functionalities in any of the horizontal layers. For example, a functionality representing traffic jams requests status from traffic observation, current and average speed, but maybe also images from the camera system on the device abstraction level for real-time traffic observation.

In a fifth step a human machine interface (HMI) is added. Full autonomy of commercial vehicles will not happen directly, and a human operator will be present either locally in the vehicle or externally in a remote-control hub. The HMI is clearly separated from the other functionalities as an additional vertical layer. One reason for this is the versatility of a human driver. The driver can execute functionalities in all layers of the horizontal architecture, e.g. route planning functionalities, traffic situation management, speed- and trajectory controller, and headway illumination controller. User input controllers (UCI) define which control is offered to a human operator and when this control becomes available.

In a sixth step functionalities hosted in the cloud can be included in the architecture as an additional product line entity that leaves the single vehicle perspectives and includes a multiple vehicle perspective. Here it is called Fleet Application System and provides fleet statistics or even services hosted in other vehicles.

The introduced reference architecture for commercial vehicles combines a strict hierarchical style as described in [22] with a heterarchical style as for example described in [23]. The architecture is object- and component-oriented, especially at the lowest level close to hardware where vehicle utilities combine several device abstraction objects and inherit properties therefrom. Each object acts as a service and provides three interfaces: Operation control, operation capability and operation status. Data is exchanged by using these interfaces. This is fundamentally different to current practice where data is exchanged by parameter passing or provided as global data. An object-oriented architecture allows for a better control over and separation of safety related functionalities from non-safety relevant functions. Furthermore, the clear separations and clear hierarchy of functionalities in the architecture prevents unwanted behaviour such as wrong access to global variables or inconsistent decisions.

4. Architectural Structures and Design Patterns

4.1. Approach

This chapter presents a systematic approach for deriving a functional and logical architecture for an ADS by using function analysis. The starting point is the definition of the function mission including an analysis of the system functions and the motivation for the existence of each system function. The system functions are divided into sub-functions and the information exchange and dependencies between different sub-functions is explained.

When the function is clearly defined, the sub-functions and similar areas can be grouped into logical elements.

4.2. Function Analysis for functional architecture development

This section describes the concept of function analysis and motivates its use for the development of a functional architecture for ADS.

4.2.1. Introduction

When systems engineers design new products, Function Analysis is performed to:

- Refine the new product's functional requirements,
- map its functions to elements,
- guarantee that all necessary elements are listed,
- ensure that no unnecessary elements are requested, and
- understand the relationships between the new product's elements.

Function Analysis System Technique (FAST) is a method for Function Analysis that is used to analyze the functional structure used in systems design. It specifies a graphical representation and logical structure. The goal is to analyze the basic function(s), underlying functions and their relations. Basic questions are posed about the function: How? Why? When? A function is then expressed as an active verb + measurable noun. For example, the function of a light bulb would be to "illuminate area", and not, "light room".

4.2.2. Background

FAST was developed in the 1960s by Charles W. Bytheway and has its origin in Value Engineering [2]. In [1], Wixson gives a background to Function Analysis and describes the FAST method.

In [3], an ECSS (European Cooperation for Space Standardization) standard for Function Analysis is described. ECSS is a cooperative effort of the European Space Agency, National Space Agencies and European industry associations for the purpose of developing and maintaining common standards. The standard defines the requirements to perform function analysis and the information outputs of that analysis. It is intended to be applied conjointly for the management, engineering, and product assurance in space projects and applications.

In [4] Viola et al. gives an overview of function analysis in systems engineering and describes the methodology and application examples.

Chapter 5 of System Engineering Fundamentals provides additional information on function analysis [5].

4.2.3. Methodology (that includes function analysis)

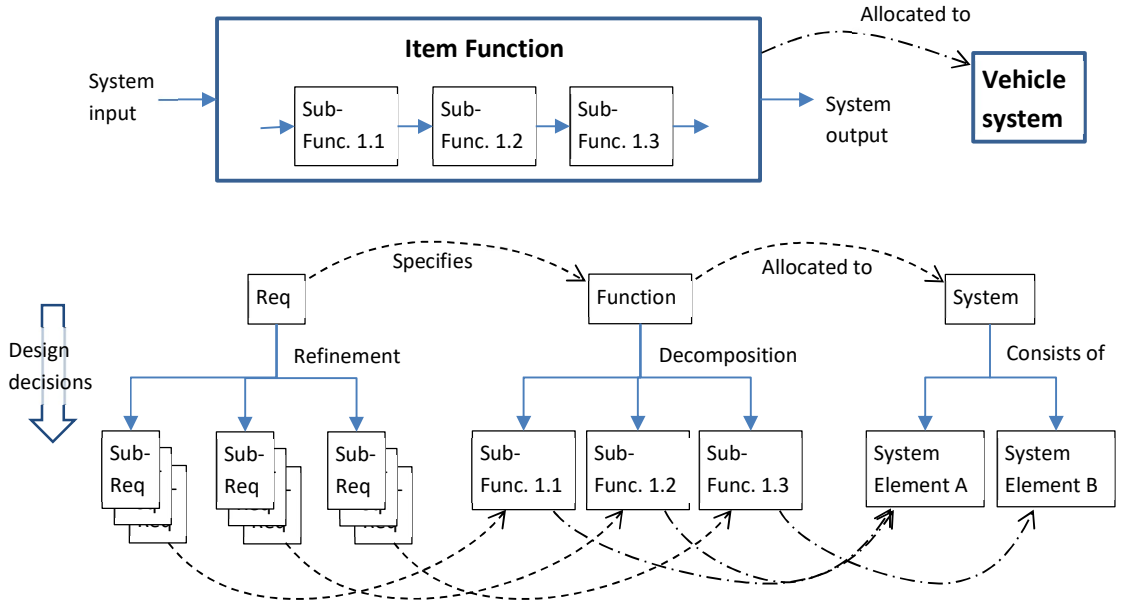

Figure 5: Requirement, function, element – Relation

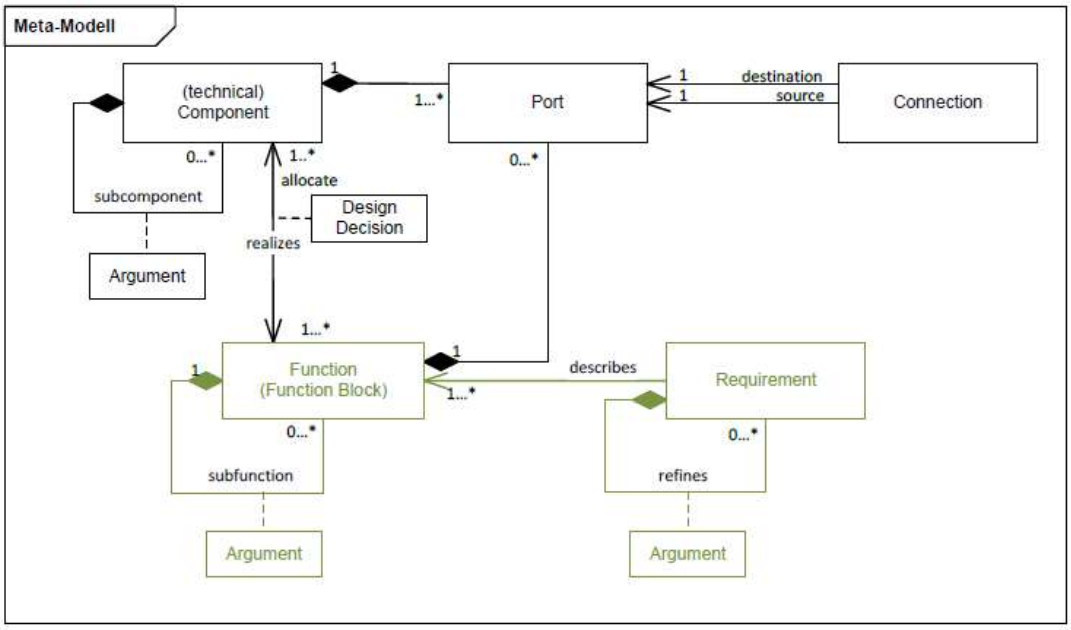

Figure 6: Meta-model of system architecture modelling, ref [17]

Figure 5 shows the relation between system requirements, system function and system (architecture). Each requirement specifying the complete system can be decomposed into a set of derived requirements, specifying sub-functions which are part of the Functional architecture. Functions are grouped into elements of the Logical architecture which are allocated to elements of the Structural/Physical architecture. The physical elements are the concrete solution realizing the functional and logical architecture.

- Requirements: The first step of the systems engineering process is to analyse the process inputs. A requirement at the top level describes use cases or the mission that an item shall perform. Safety related top level requirements are called safety goals. Analysis of requirements is performed to develop functional and performance requirements; that is, a use case is translated into a set of requirements that define what the system must do and how well it must perform.
- Function: Functions are analysed by decomposing higher-level functions identified through requirements analysis into lower-level functions. The performance requirements associated with the higher level are decomposed into requirements on lower level functions. The result is a description of the product or item in terms of what it does logically and in terms of the performance required. Function analysis and allocation allow for a better understanding of what the system must do, in what ways it can do it, and to some extent, the priorities and conflicts associated with lower-level functions. It provides information essential for optimizing the architecture of the solutions. A functional flow block diagram (FFBD) can be used to describe system requirements in functional terms, i.e. how the functions are related. Each function can also be decomposed into sub-functions and these also organised in a FFBD. This implies improved traceability.
- Design Synthesis: Design synthesis is the process of defining the item in terms of the elements and architecture which together make up and define the item. Depending on abstraction level this can have different names, e.g. preliminary architecture (functional level) or system design (system level). Each part must meet at least one functional requirement and any part may support many functions. The architecture is the basic structure for generating the specifications and baselines.

The activities and also outcomes can be described as depicted in Figure 7: Iterative development. The activities are confined to one abstraction level, e.g. system level. The outcome is an architecture and requirements. Also, there will be evidence created from the performed analyses, e.g. function analysis or safety analysis.

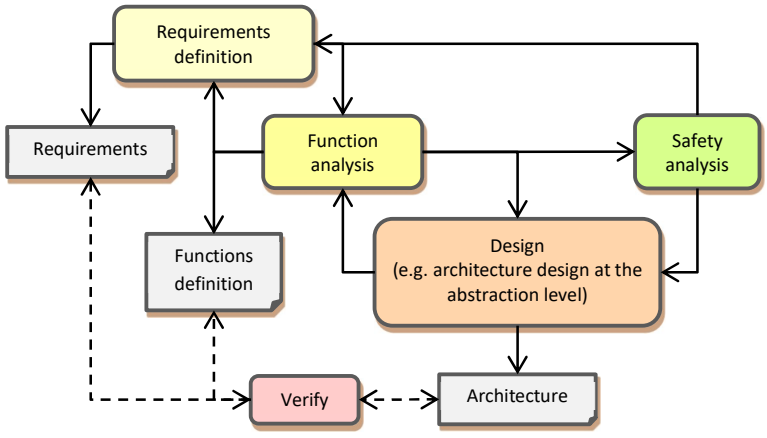

Figure 7: Iterative development

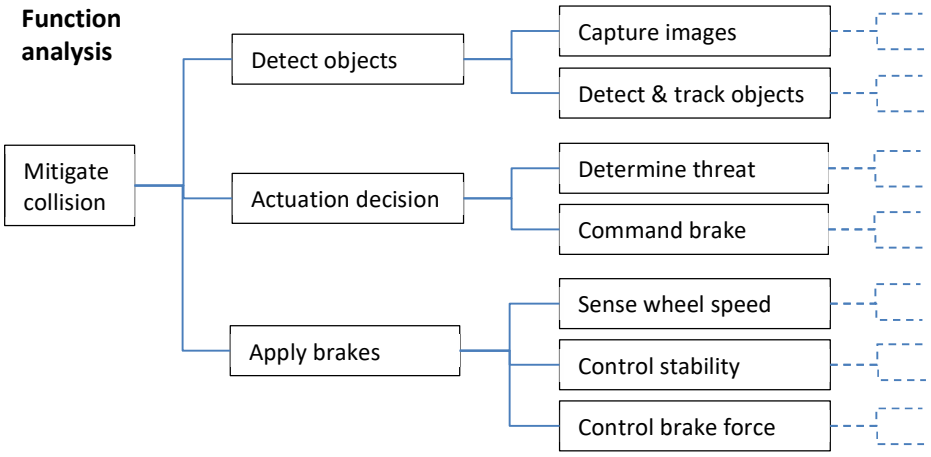

Figure 8: Example of applied Function Analysis

Figure 8 illustrates how function analysis is applied to a collision mitigation function. The figure shows a function tree that supports the top-level function “Mitigate Collision”. Each lower level (right to left in the figure) describes “how” the higher-level function is performed. Conversely, moving to the higher level (left to right in the figure) answers the question “why”. To find basic functions a function tree should be decomposed as far as possible, within the current level of abstraction. Each identified function will be described as a requirement and is also allocated to an element.

4.3. Safety architecture design patterns - overview

When designing a complex system, established and well-known design patterns are applied as basis for the architecture. In the safety-critical application domain, there are many safety design patterns that have been researched, e.g. [6], [7].

General requirements for ADS is that the ADS needs to be fail-operational and rely on redundancy to be practically feasible to implement. A few safety patterns are exemplified below that are suitable for ADS.

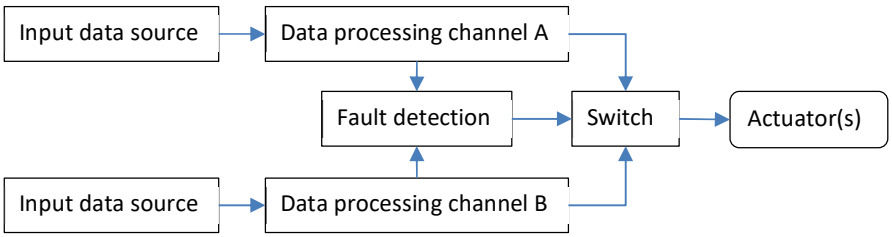

Figure 9. Heterogeneous redundancy pattern

The Heterogenous Redundancy Pattern is a diverse redundancy pattern including diagnosis (1oo2D) switching from the primary cannel A to secondary (backup) channel B in case a fault is detected. Main advantages: 1) The system is fully operational even in case of a single channel failure, 2) A single channel random fault does not lead to a system failure, 3) A single channel systematic fault does not lead to a system failure, 4) The system can detect a fault in a single channel.

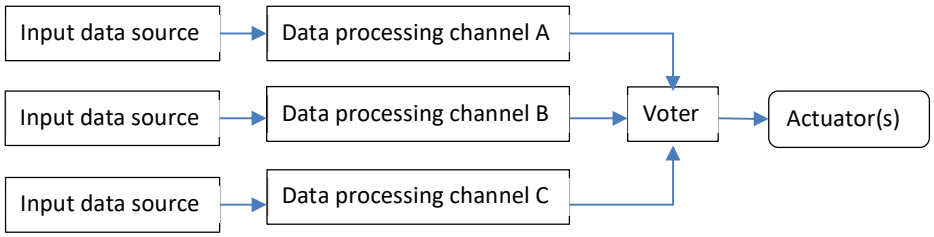

Figure 10. Triple modular redundancy pattern (TMR)

The Triple Modular Redundancy Pattern is a homogeneous triplex pattern where a majority voter 2-out-of-3 decides for the correct result (2oo3). Main advantages: 1) The system is fully operational even in case of a single channel failure, 2) A single channel random fault does not lead to a system failure.

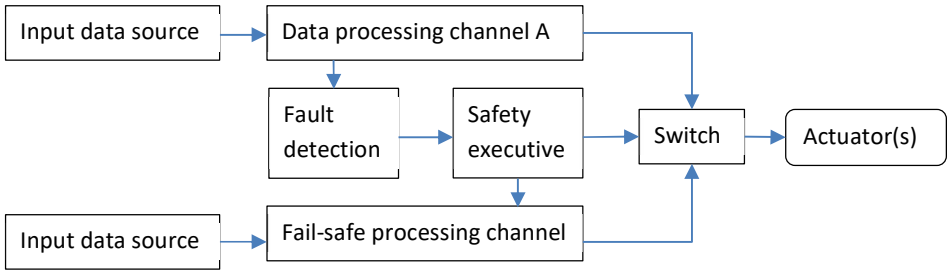

Figure 11. Safety executive pattern

The Safety Executive Pattern is a safety kernel simplex pattern where the primary channel A is diagnosed and in case a fault is detected a dedicated Safety executive is deploying the safety measures to switch over to the Fails-safe channel performing a complex emergency operation to enter a safe state. Main advantages: An emergency operation is performed if the primary channel failure is detected.

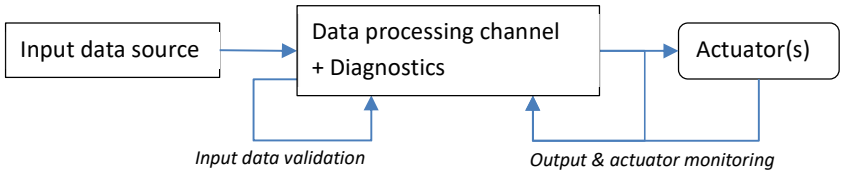

Figure 12. Protected single channel

The Protected Single Channel Pattern is a simplex channel including diagnosis covering input data validation and monitoring of output and actuators(s). A fail-safe state is entered if a fault or failure mode is detected. Main advantage: 1) Fail-safe state is entered if fault or failure mode is detected.

More pattern exists, both simpler and more complex, e.g. Monitor-Actuator pattern, M-out-of-N-D pattern.

Different safety design patterns are preferably combined in one way or the other in a real ADS application.

4.4. Deriving of safety requirements from safety goals

In this section the Safety Goals, i.e. top-level safety requirements, for a SAE level 4 ADS are assumed to have been defined in a HARA and the question is how to derive safety requirements from the safety goals.

ISO 26262 has two types of safety requirements; (1) Functional safety requirements (FSRs) defined during the Concept phase as part of the functional safety concept phase, and (2) Technical safety requirements (TSRs) defined during the system level product development phase. The FSRs specify implementation-independent safety behavior or implementation-independent safety measure, including safety-related attributes as ASIL and allocated to elements within the architecture. The TSRs are the requirements derived for implementation of associated FSRs. Consequently, the focus in this chapter will be on FSR and not on TSR.

4.4.1. The item

The objective with the HARA is to define safety goals for the autonomous vehicle. A starting point may be the comparison with models for a car driven by a human driver. A wide selection of research literature on decision hierarchies exists, starting with an overview from the 1980s in [8]. Using this as a starting point, the decision hierarchy can typically be divided into a strategic level, tactical/manoeuvring level and operational/control level, see Figure 13 from SAEJ3016 [9]. This may be complemented by a simplified human performance model adopted from [10], shown in Figure 14, and by a Dynamic Driving Task (DDT), adopted from [9], shown in Figure 15.

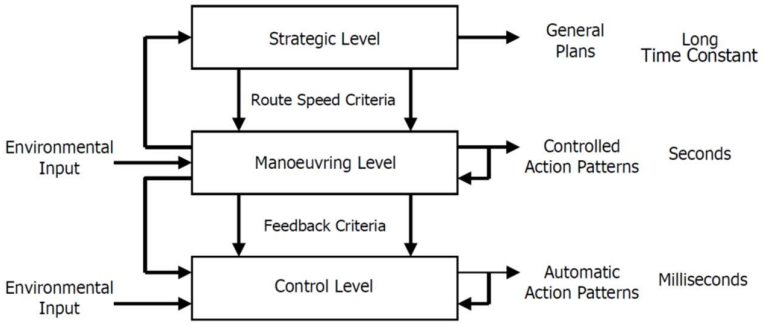

Figure 13 The hierarchical structure of the road user task with the performance structured at three, comparatively loosely, coupled levels [8].

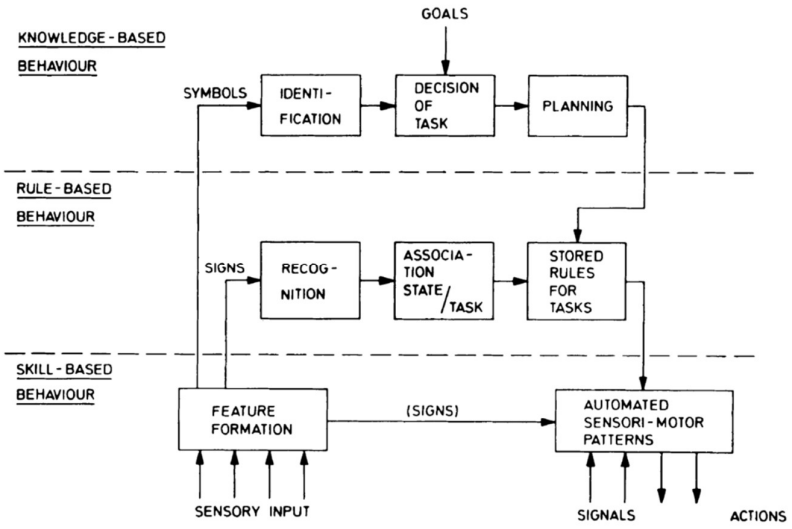

Figure 14. A simplified illustration of three levels of performance of skilled human operators [10].

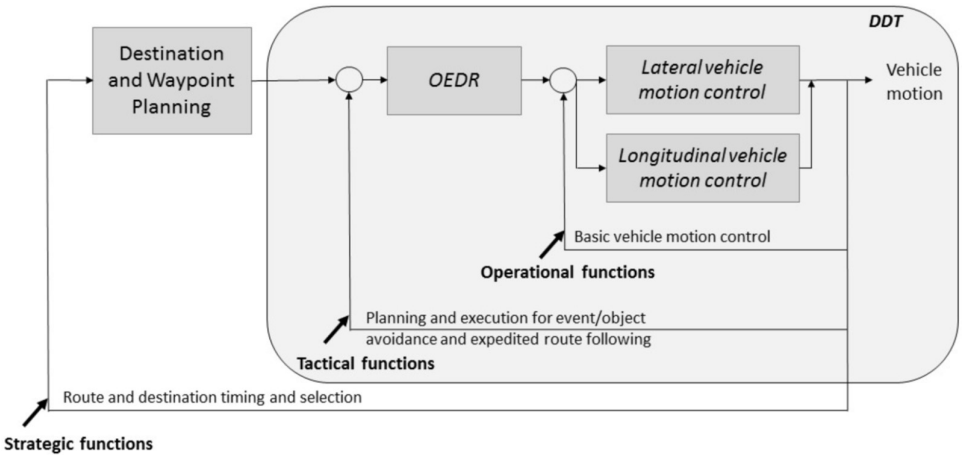

Figure 15. Schematic view of driving task showing Dynamic Driving Task (DDT) [9]. OEDR is an acronym for object and event detection and response. Not that it seems in the figure like OEDR is a tactical function only, but in the text, it stated that it is both on tactical and operational level.

A very generalized view of an ADS item is shown in Figure 16 which operates based in input information and control the vehicle motion.

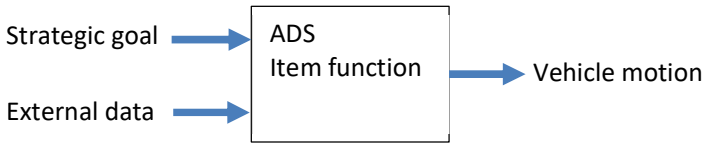

Figure 16. Generalized vies of an item.

An attempt to illustrate this is done in Figure 17 in which the approaches from Figure 13, Figure 14, Figure 15, and Figure 16 are combined. It also adds external data that include feature information from external sources.

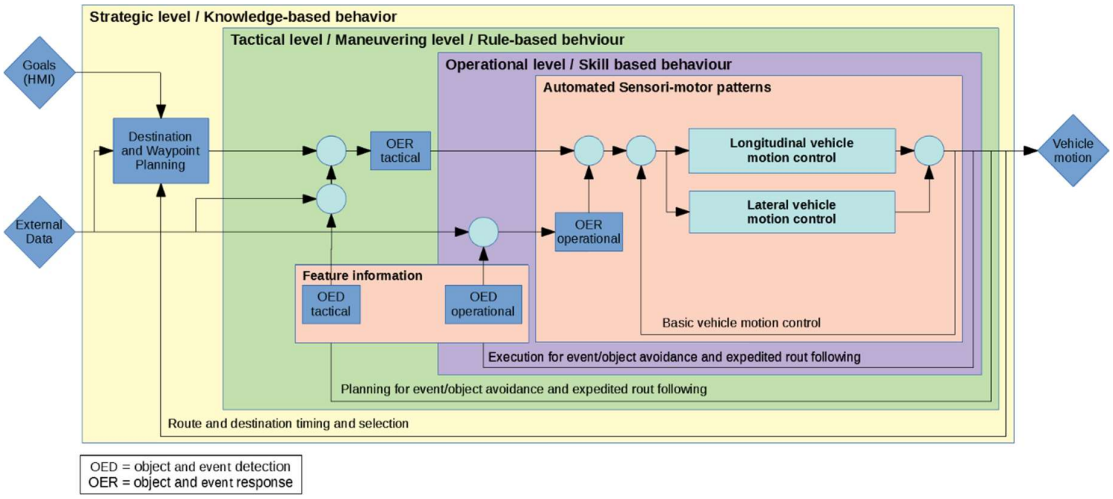

Figure 17. A schematic view of the ADS decision hierarchy.

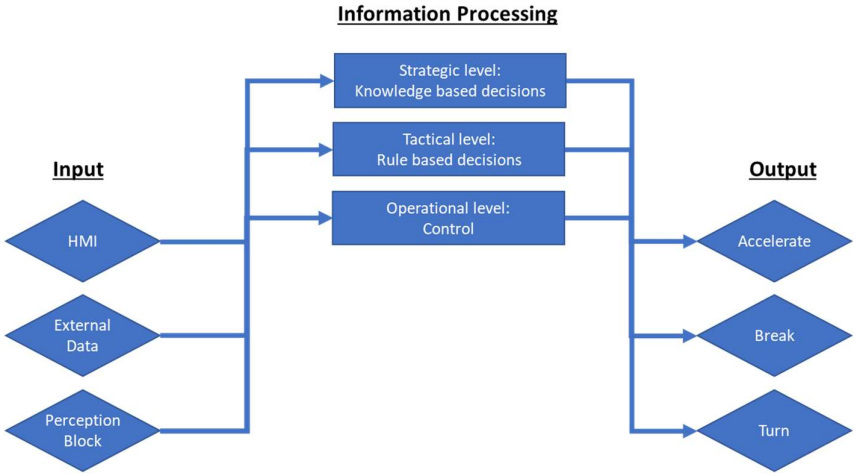

Figure 18. Proposed item view for ADS.

4.4.2. ADS decision hierarchy function analysis example

For ADS equipped vehicles, a list of safety goals are defined e.g. with the method described in ESPLANADE R4.2v4 [11]. Typically, safety goals imply avoiding collisions with various objects in vehicle path (e.g. vehicles, vulnerable road users, animals, and non-identified objects of a certain minimum size), and most likely, some restriction on not to leave the lane in certain ways.

Further the DDT is analyzed and broken down to functions in a functional structure by, e.g. applying the FAST method, example given in Figure 19. The specific functions are then allocated to the ADS hierarchical model, see Figure 20 and functional requirements are specified for each of the functions.

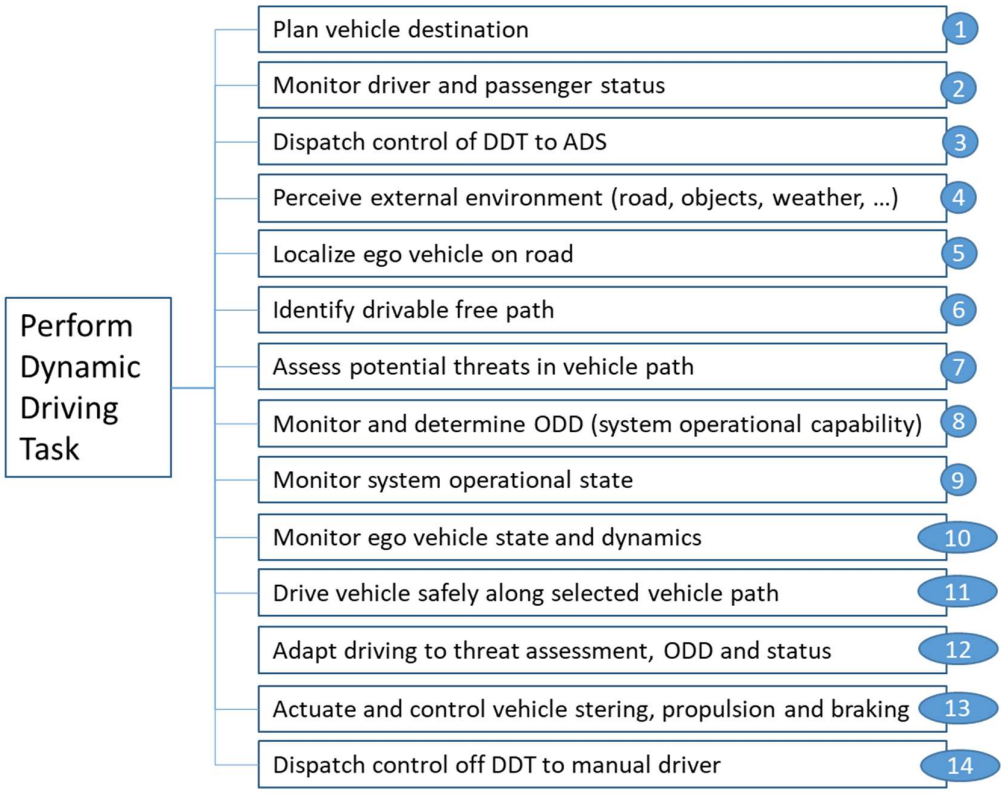

Figure 19. Function analysis of ADS DDT

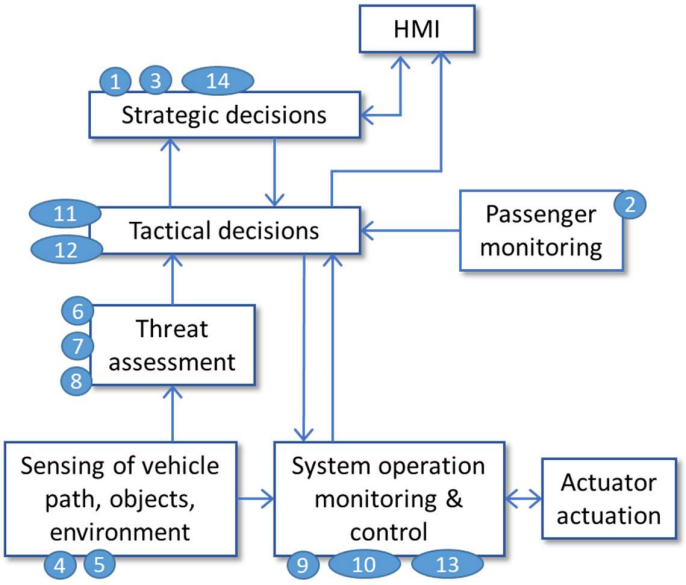

Figure 20. Allocation of ADS functions to the ADS decision hierarchy

4.4.3. ADS decision hierarchy – Preliminary FSC

4.4.3.1. FSC functional architecture

Based on the hierarchical model, a preliminary FSC architecture, Figure 21, is developed taking different design considerations into account. The FSC specifies FSRs as basis of safety analysis, covering the specific functionality, capability, information, and confidence each block must guarantee and the prevention of failure mode propagation. This can be done in different ways, but it is an important that an FSC remains implementable. It is not enough that the FSC is consistent and complete with respect to fulfilment of all safety goals, the FSRs also needs to be formulated in such a way that there is a realistic strategy for its implementation.

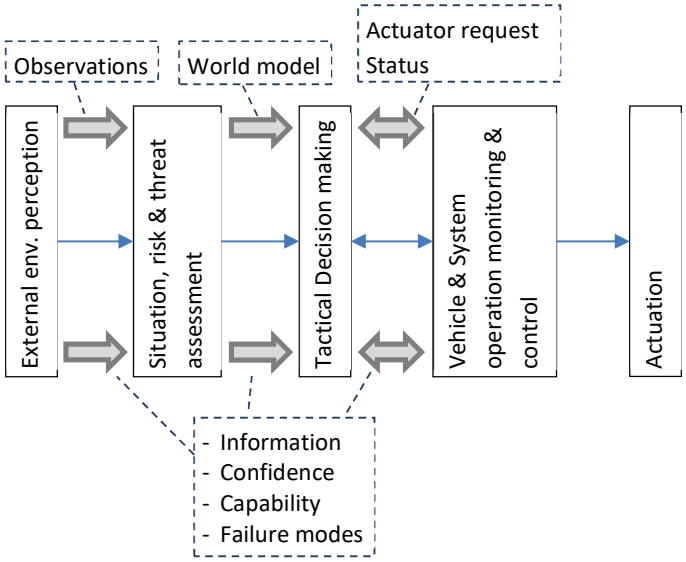

Figure 21. Preliminary FSC architecture – High level

Each functional block of preliminary FSC architecture are preferably described using a generic design pattern according to Figure 22.

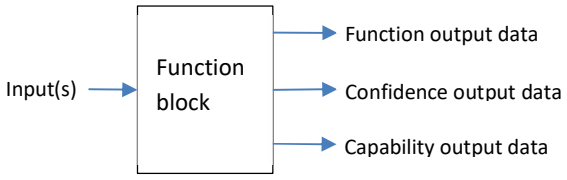

Figure 22. Generic FSC function block design pattern

Where the

- Function output data is the output response of the requested functionality.
- Confidence output data is the confidence level of that the function output data is within the specified range of values. This is required as input to the Tactical decision making.
- Capability output data is the current capability of the system to provide the requested functionality. This includes diagnosis of the function block for internal failures, plausibility

checks of input data and evaluation if external conditions is affecting the ability to provide the functionality within the defined ODD.

A safety analysis is carried out identifying the safety relevant failure modes of the item function and its decomposed functions based on a simplistic error model, Figure 23. The nominal response of each function is defined by the functional requirements. A system fails when characteristics of the requested functionality can't be provided. Failure modes are identified by applying FMEA method using guide words such as *omission/commission*, *too early/late*, *value too much/little*. Faults can be categorized in *input faults*, *internal faults* and *external noise sources*.

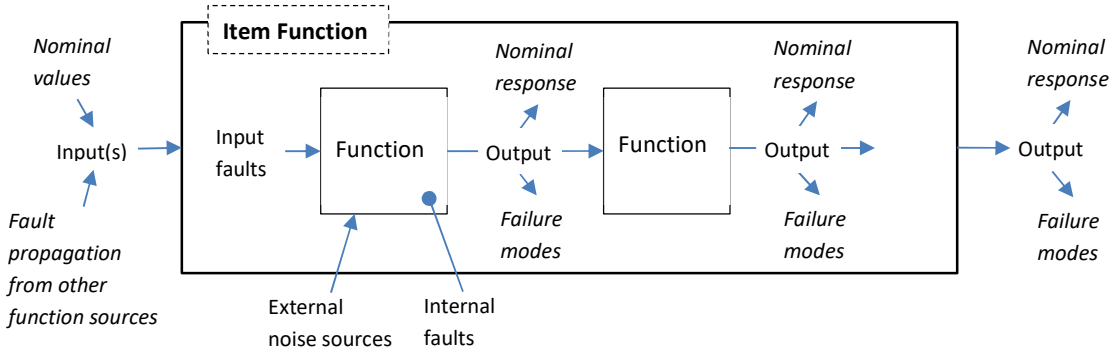

Figure 23. Error model

A failure is caused by either internal faults, input faults propagated from other function sources or external noise sources.

In the ADS example, functions 4, 6, 9 and 12 is used as examples to derive functional requirements, see Table 1 below.

Table 1. Example ADS functions.

Functional block	Function	Functional requirements: "The ADS shall …"
External environmental perception	4. Perceive surroundings (road, objects, weather, …)	4.1 Provide detected lanes, road edges, free space and surrounding objects with sufficient confidence. 4.2 Provide position & motion of ego vehicle. 4.3 Provide information of current external environmental conditions 4.4 Provide information of the perception capability of providing the requested functionality
Threat assessment	6. Identify drivable free path	6.1 Create a real-world model with sufficient confidence of lanes, free space and surrounding objects. 6.2 Localize ego vehicle in relation to real world model. 6.3 Provide information of the threat assessment capability of providing the requested functionality
System operation monitoring & control	9. Control vehicle operation and monitor system status	9.1 Control the actuation of propulsion, steering and braking for the vehicle

		operation according to selected path and speed 9.2 Provide information of current capability of the system elements to provide the requested functionality
Tactical decision making	12. Adapt driving to threat assessment, ODD and status	12.1 Determine safe vehicle path free of objects. 12.2 Determine required safe distance to objects. 12.3 Operate the vehicle to maintain safe distance within its current capability

Assume the following hazard identified in the hazard analysis "To short distance kept to object, based on current speed and capability of braking and steering" and the corresponding SG: "Assure safe minimum distance in accordance with selected path, current speed and driving conditions; ASIL D".

Failure modes of the system which could lead the hazardous event shall then be identified. Example of relevant failure modes are identified by FMEA method, see Table 2 below.

Table 2. Example failure modes.

Functional requirements: "The ADS shall …"	Relevant failure modes
4.1 Provide detected lanes, road edges, free space and surrounding objects with sufficient confidence.	1. Existing objects not detected (False negatives) 2. Non-existing objects detected (False positives) 3. Incorrect position of detected objects 4. Incorrect distance to detected objects 5. Incorrect speed of detected objects 6. Confidence level of de detection is reported to be higher than true confidence level of the detections 7. … <more failure modes>
4.2 Provide position & motion of ego vehicle.	1. Incorrect position of ego vehicle 2. Incorrect distance to ego vehicle 3. Incorrect speed of ego vehicle 4. … <more failure modes>
4.3 Provide information of current external environmental conditions	1. Environmental conditions are reported to be better than the actual real conditions 2. … <more failure modes>
4.4 Provide information of the perception capability of providing the requested functionality	1. Reported capability is above its actual capability 2. … <more failure modes>
6.1 Create a real-world model of lanes, free space and surrounding objects with sufficient confidence.	1. Existing objects not existing in world model 2. Non-existing objects present in world model 3. Objects incorrectly positioned in world model 4. Incorrect object distance in world model 5. Incorrect object speed in world model 6. … <more failure modes>
6.2 Localize ego vehicle in relation to real world model.	1. Incorrect position and speed of Ego vehicle in world model 2. … <more failure modes>
6.3 Provide information of the threat assessment capability of providing the requested functionality	1. Reported capability is above its actual capability 2. … <more failure modes>
9.1 Control the actuation of propulsion, steering and braking for the vehicle operation	1. Incorrect actuation: a. Too high propulsion torque

according to selected path and requested propulsion torque and brake force	b. Too less brake force c. Erroneous steering angle 2. … <more failure modes>
9.2 Provide information of current capability of the system elements to provide the requested functionality	1. Reported system capability is above its actual capability, e.g. relevant internal system faults are not detected 2. … <more failure modes>
12.1 Determine safe vehicle path free of objects.	1. Selected path is not free from objects 2. Erroneous path (leading to a potential hazard) 3. … <more failure modes>
12.2 Determine required safe distance to objects.	1. Determined safe distance to objects is too short (to avoid a potential hazard) 2. … <more failure modes>
12.3 Operate the vehicle (request propulsion torque, brake force and steering angle) to maintain safe distance within its current capability	1. Incorrect request of propulsion torque, brake force and steering angle to maintain safe distance 2. Vehicle is operated outside its current capability defined by ODD or reported system status 3. … <more failure modes>

4.4.3.2. Specification of FSR

Based on the failure mode analysis, corresponding FSRs are then specified for each of the failure modes. Several methods have been proposed in related work, e.g. [12], [13], how to formulate FSRs. For the purpose of this study, a few example FSRs are presented.

The FSR are specified and formulated applying a generic strategy as follows:

- To assure the nominal functional response to an acceptable level, i.e. to prevent the failure mode, or reduce the likelihood of the failure mode to an acceptable level.
    - Typical requirement pattern:

      “The <entity> shall assure <property> within a <range/min/max> of <value> <unit> under the condition <specific condition>; ASIL X”

- Detect the presence of the failure mode and make an appropriate failure reaction. A failure reaction should be either 1) transition to a fall-back function, 2) transition to a degraded mode, 3) perform an emergency operation, or 4) transition to a safe state.
    - Typical requirement pattern:

      “The <entity> shall detect <failure mode> of <property> exceeding a <range/min/max> of <value> <unit>; ASIL X”

      “If <failure mode> is detected then the <entity> shall <enter/perform> <state/action> within <reaction time> <unit>; ASIL X”

Specific for ADS is that safe state is required to be fail-operation when in AD mode. This affects how the FSRs are formulated for preventing and detection of failure modes. The strategy for fail-operational, e.g. strategy for fall-back function, degraded mode, emergency operation and safe state,

needs to be defined as part of the AD use-case and functional requirements. A simplified fail-operational strategy is defined for this study.

Example FSR covering some of the failure modes in Table 2:

Table 3. Example FSRs.

Relevant failure modes	FSR	Rationale
4.1.1 Existing objects not detected (False negatives)	The Perception shall assure detection of existing objects, for each object class defined in the scene catalogue, to a minimum probability of Pd % at a minimum distance of D meter under the conditions defined by the ODD; ASIL D	Objects must be detected in order to determine a drivable path free of objects to assure the selected tactical decision.
4.1.2 Non-existing objects detected (False positives)	No FSR specified	Non-existing objects is not violating the SG: "Assure safe minimum distance …"
4.1.3 Incorrect position of detected objects	The Perception shall assure the detected object position error, for each object class defined in the scene catalogue, to a maximum X % compared with existing object position at a minimum distance of D meter under the conditions defined by the ODD; ASIL D	Object position must be within the error tolerance in order to maintain a safe distance to objects on the border of the safe path to assure the selected tactical decision.
4.1.6 Confidence level of de detection is reported to be higher than true confidence level of the detections	The Perception shall assure the reported confidence level is not higher than the true confidence level of the detections; ASIL D	The confidence level shall not be overrated compared with true value to assure that the tactical decision is taken with high confidence.
4.4.1 Reported capability is above its actual capability	The Perception shall assure the reported capability is not higher than the true capability of providing the perception functionality; ASIL D	The capability shall not be overrated compared with true capability to assure that the selected tactical decision is based on valid data.
9.1.1. Incorrect actuation a. Too high propulsion torque b. Too less brake force c. Erroneous steering angle	a. The System operation monitoring & control shall assure that actual propulsion torque is provided within +/-X % of the requested torque to the drive line system; ASIL D b. … c. …	The actuation of torque, brake force and steering angle must be controlled and monitored in order to assure that the vehicle operation are within the limits to assure driving according to the selected path and speed.
9.2.1. Reported system capability is above its actual capability, e.g. relevant internal system faults are not detected	The System operation monitoring & control shall assure the reported capability is not higher than the true capability of providing the perception functionality; ASIL D	The capability shall not be overrated compared with true capability to assure that the selected tactical decision is based on the actual capability of the control system.
12.2.1 Determined safe distance to objects is too short (to avoid a potential hazard)	The safe distance to objects shall be determined implying a maximum impact speed depending on their vulnerability. ASIL D. Maximum impact speed of objects is: • Vehicles: XX km/h	The safe distance is adapted to the vulnerability of the objects on the border of the safe path, limiting the maximum impact speed in case of a potential hazardous event.

		• Vulnerable road users: YY km/h • …	
12.3.1	Incorrect request of propulsion torque, brake force and steering angle to maintain safe distance	The Tactical decision making shall adapt vehicle operation according to the selected path and speed, assuring the requested propulsion torque, brake force and steering angle are within the tolerance margin +/-X% to maintain safe distance; ASIL D	The vehicle operation must be operated according to the tactical decisions
12.3.2	Vehicle is operated outside its current capability defined by ODD or reported system status	a. The Tactical decision making shall assure that the ADS is operated within its capability; ASIL D b. The Tactical decision making shall detect whether the capability status of ADS functional blocks is Capable/Not capable; ASIL D c. If the capability of the External environmental perception or the Threat assessment functional blocks is detected to be Not capable then the Tactical decision shall enter the degraded mode DM within reaction time of XXX ms: ASIL B d. If the capability of the System operation monitoring & control or the Tactical decision making functional blocks is detected to be Not capable then the ADS shall perform the emergency operation EO within reaction time of XXX ms; ASIL B e. Degraded mode DM shall assure DDT fallback operation until minimal risk condition is reached; ASIL B f. Emergency operation EO shall assure minimal risk condition where the vehicle come to a safe stop; ASIL B.	Continuous monitor of the capability is required to assure that the tactical decisions are valid and can be actuated accordingly. If the system is diagnosed to be not capable then a degraded mode must be entered or an emergency operation must be performed. A HARA is assumed to be performed for both degraded mode DM and emergency operation EO with resulting SG of maximum rating of ASIL B due to that the probability of exposure that capability is Not capable and the need to operate in DM or EO is low.

A strategy for defining FSRs assuring that the DDT is safe by adapting to the capability of the ADS is further discussed in [15].

4.4.3.3. FSC logical and physical architecture

An example logical and physical architecture is developed based on the preliminary FSC in Figure 21 for the purpose of this study. The architecture is much simplified and several vital elements are purposely left out. There is no more intention with the example architecture then to demonstrate the method of applying safety architecture design patterns, further deriving FSRs and the allocation to architectural elements.

During the process deriving FSRs, ASIL decomposition (according to ISO 26262 [16]) is used to limit the required ASIL D to a lower level ASIL, applicable for each of the specific physical elements, in order to reduce development effort.

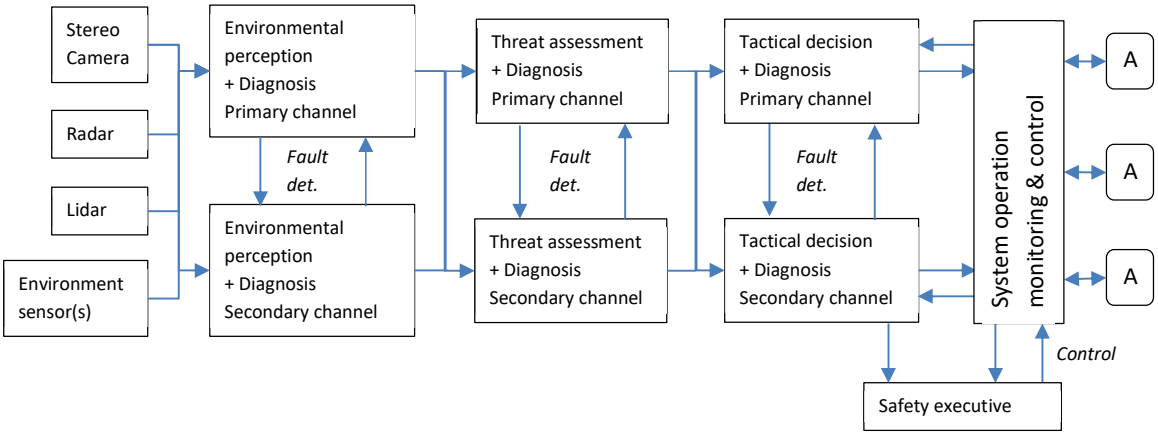

Figure 24. Elements of the FSC logical architecture

The FSC logical architecture applies a combination of Heterogenous Redundancy Pattern, Safety Executive Pattern, Protected Single Channel and Monitor-Actuator design patterns.

The sensor elements Stereo Camera, Radar, Lidar and Environmental sensor(s) are configured as protected single channels, providing redundant information of the environment to the Environmental perception element.

The elements of Environmental perception, Threat assessment and Tactical decision are configured in two diverse redundant processing channels, where the

- Primary channel is performing two basic tasks:
    - The more complex task of executing the complete functionality of environment perception, threat assessment and tactical decisions providing vehicle operation actuation requests to the System operation monitoring control element.
    - Monitor and detect faults in each logical element of the Secondary channel and System operation monitoring & control element.
- Secondary channel is performing three basic tasks:
    - Monitor and verify the functional output of each logical element in the Primary channel.
    - Monitor and detect faults in each logical element of the Primary channel and System operation monitoring & control element.
    - Perform degraded mode DM executing the DDT fallback operation in case a fault is detected in the Primary channel.
- Each of the Primary and Secondary channels are configured as protected single channel, checking its inputs and monitoring its outputs.

The System operation monitoring & control element is configured as a protected single channel, receiving information (e.g. actuation requests capability status, etc) from both the Primary and Secondary channel. The element is performing three basic tasks:

- Checking the inputs from the Primary channel versus the inputs from the Secondary channel verifying that the data (e.g. actuation requests capability status, etc) are within acceptable range.
- Control and monitors the operation of the actuators.
- Monitors the vehicle system regarding its capability.

The combination of Secondary channel, System operation monitoring & control and Safety executive elements are configured as a Safety Executive Pattern, where the Safety executive checks the combined capability of the of Secondary channel and System operation monitoring & control, and if a fault is detected, Safety executive take over control and an emergency operation is performed to reach a fail-safe state.

The enhanced FSC logical architecture enables further derivation of the FSRs. The basic method is to make ASIL decompositions that are in line with each of the refined logical elements. The ASIL decomposition used in this example is applying the following decomposition:

ASIL D = ASIL B(D) + ASIL B(D)

A few FSRs from Table 3 are derived and exemplified Table 4 below (Note. The same requirement patterns applies as described previously). Further the decomposed FSR is allocated to elements of the logical architecture.

Table 4. Derived FSR

FSR ID	FSR	Decomposed FSR	Allocation
4.1.1	The Perception shall assure detection of existing objects, for each object class defined in the scene catalogue, to a minimum probability of Pd % at a minimum distance of D meter under the conditions defined by the ODD; ASIL D	The Perception shall assure detection of existing objects, for each object class defined in the scene catalogue, to a minimum probability of Pd % at a minimum distance of D meter under the conditions defined by the ODD; ASIL B	Environmental perception Primary channel
		The Perception shall verify detection of existing objects, for each object class defined in the scene catalogue, to a minimum probability of Pd % at a minimum distance of D meter under the conditions defined by the ODD; ASIL B	Environmental perception Secondary channel
		In case verification of detection of existing objects fails, the Confidence level shall be decreased by a factor of X%; ASIL B	Environmental perception Secondary channel
4.1.6	The Perception shall assure the reported confidence level is not higher than the true confidence level of the detections; ASIL D	The Perception shall assure the reported confidence level is not higher than the true confidence level of the detections; ASIL B	Environmental perception Primary channel
		The Perception shall verify the reported confidence level is not higher than the true confidence level of the detections; ASIL B	Environmental perception Secondary channel
		In case verification of reported confidence level fails, the Confidence level shall be decreased by a factor of X%; ASIL B	Environmental perception

			Secondary channel
4.4.1	The Perception shall assure the reported capability is not higher than the true capability of providing the perception functionality; ASIL D	The Perception shall assure the reported capability is not higher than the true capability of providing the perception functionality; ASIL B	Environmental perception Primary channel
		The Perception shall verify the reported capability is not higher than the true capability of providing the perception functionality; ASIL B	Environmental perception Secondary channel
		In case verification of the reported capability fails, the Capability shall be reported as Not capable; ASIL B	Environmental perception Secondary channel

By following this approach, all ASIL D requirements of the functional elements Environmental perception, Threat assessment and Tactical decision can be derived in to ASIL B, except the System operation monitoring & control element which requires ASIL D to sum the two ASIL B(D) path together. All derived FSRs are now allocated to the elements of the logical architecture.

A concern not considered in the developing this simplified logical architecture and corresponding derivation of FSRs, are the determination and specification of sufficient redundancy of the combined sensor information required for the perception functionality. The concern of specification of safety requirements for perception in a multi sensor system is discussed in the related "ESPLANADE"—paper [14]. Thus the derivation of safety requirements for the sensor elements has not been addressed, and for this study it's assumed for simplistic reasons, that sensor elements are of maximum ASIL B.

A physical architecture is developed realizing the logical architecture. Different design options and trade studies must be evaluated in order to come to an acceptable architectural solution. For the purpose of this study, separate physical sensor elements, a centralized ADS controller, a Vehicle main system controller and a separate Emergency operation controller has been defined according to Figure 25.

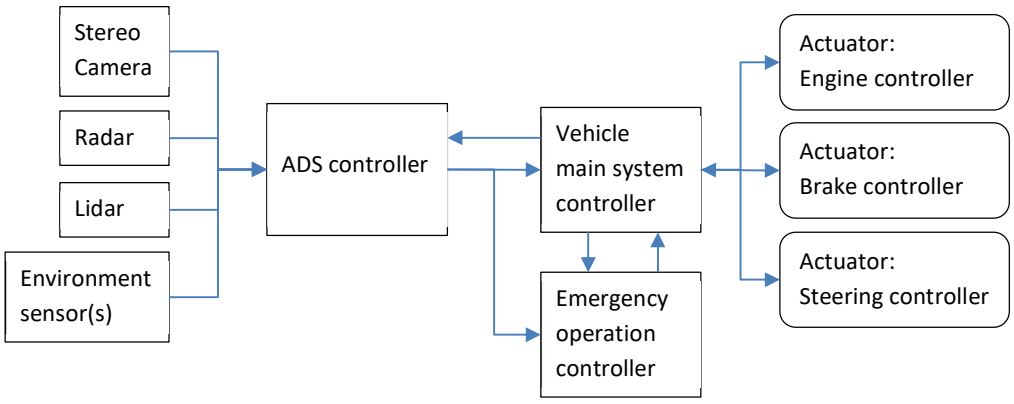

Figure 25. Elements of the FSC physical architecture

The elements of the logical architecture are allocated to the elements of the physical architecture according to Table 5.

Table 5. Allocation of architectural elements

Logical element	Physical element	Maximum ASIL level
Stereo Camera	Stereo Camera	ASIL B
Radar	Radar	ASIL B
Lidar	Lidar	ASIL B
Environmental sensor(s)	Environmental sensor(s)	ASIL B
Environmental perception – Primary channel	ADS controller	ASIL B
Threat assessment and – Primary channel	ADS controller	ASIL B
Tactical decision – Primary channel	ADS controller	ASIL B
Environmental perception – Secondary channel	ADS controller	ASIL B
Threat assessment and – Secondary channel	ADS controller	ASIL B
Tactical decision – Secondary channel	ADS controller	ASIL B
System operation monitoring & control	Vehicle main system controller	ASIL D
Safety executive	Emergency operation controller	ASIL B

5. Conclusions and future work

The objective to propose and apply a systematic methodology for deriving a functional and logical architecture for automated driving systems (ADS) has been demonstrated. By using a systematic methodology, a function analysis defining the nominal function, perform safety analysis identifying safety relevant failure modes. Based on the SG and the corresponding failure modes, FSRs can be established. Applying a structed method of using safety design patterns a logical architecture can be developed that fulfills the FSRs. FSRs can be further derived using ASIL decomposition, reducing the required ASIL. Example FSRs has been specified which are allocated to elements of a logical and physical architecture.

References

[1] Wixson, J. R. (1999, June). 2 Function Analysis and Decomposistion using Function Analysis Systems Technique. In INCOSE International Symposium (Vol. 9, No. 1, pp. 800-805).

[2] https://en.wikipedia.org/wiki/Value_engineering, Accessed 2018-06-13

[3] ECSS-E-10-05A, SPACE ENGINEERING: FUNCTIONAL ANALYSIS (13 APR 1999).

[4] Viola, N., Corpino, S., Fioriti, M., & Stesina, F. (2012). Functional analysis in systems engineering: Methodology and applications. In Systems engineering-practice and theory. InTech.

[5] Systems Engineering Fundamentals, United States Government US Army, 2001 ISBN-13: 978-1484120835

[6] ARMOUSH, A. Design Patterns for Safety Critical Embedded Systems. PhD Thesis. ISSN 0935–3232, RheinlandWestfalen Technische Hochschule (RWTH), Aachen, 2010

[7] Preschern, C., , Kajtazovic, N., Kreiner, C. 2015. Building a Safety Architecture Pattern System. EuroPLoP '13: Proceedings of the 18th European Conference on Pattern Languages of Program, Article 17.

[8] Michon, John A. 'A Critical View of Driver Behavior Models: What Do We Know, What Should We Do?' In Human Behavior and Traffic Safety, edited by Leonard Evans and Richard C. Schwing, 485–524. Boston, MA: Springer US, 1985. https://doi.org/10.1007/978-1-4613-2173-6_19.

[9] SAE. 'SAE_J3016_Taxonomy and Definitions for Terms Related to Driving Automation Systems.Pdf', n.d.

[10] Rasmussen, J. 'Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models'. IEEE Transactions on Systems, Man, and Cybernetics SMC-13, no. 3 (May 1983): 257–66. https://doi.org/10.1109/TSMC.1983.6313160.

[11] ESPLANADE Deliverable R4.2v4 - Methods for hazard analysis and risk assessment - Project results

[12] Teemu Tommila, Antti Pakonen. Controlled natural language requirements in the design and analysis of safety critical I&C systems. VTT-R-01067-14

[13] Kristian Beckers, Isabelle Côté, Thomas Frese, Denis Hatebur, Maritta Heisel. Systematic Derivation of Functional Safety Requirements for Automotive Systems. Published in SAFECOMP 2014

[14] Cassel, A., Bergenhem, C., Christensen, O.M., Heyn, H.-M. et al., "On Perception Safety Requirements and Multi Sensor Systems for Automated Driving Systems," SAE Technical Paper 2020-01-0101, 2020, doi:10.4271/2020-01-0101

[15]Johansson, R., Alissa, S., Bengtsson, S., Bergenhem, C., Bridal, O., Cassel, A., Chen, D.J., Gassilewski, M., Nilsson, J., Sandberg, A. and Ursing, S., 2017, September. A strategy for assessing safe use of sensors in autonomous road vehicles. In International Conference on Computer Safety, Reliability, and Security (pp. 149-161). Springer, Cham.

[16]ISO. ISO 26262:2018 Road vehicles – Functional safety, 2018.

[17]Design of the AMASS tools and methods for architecture driven assurance (AMASS, D3.3, 2018)

[18]A functional architecture for autonomous driving' by Sagar Behere, Martin Törngren, KTH in *Workshop on Automotive Software Architectures (WASA) 2015.* [Online] Available: https://sagar.se/files/wasa2015.pdf.

[19]'DeSyRe: Decomposition of Systems and their Requirements' by Birgit Penzenstadler, doctors thesis, Technischen Universität München, 2010

[20]A. Magnusson, L. Laine och J. Lindberg, "Rethink EE Architecture in Automotive to facilitate Automation, Connectivity, and Electro Mobility," *Proceedings of ICSE-SEIP,* 2018.

[21]International Organization for Standardization, Road Vehicles - Functional Safety, Geneve, 2016.

[22]National Institute of Standards and Technology, "A Reference Model Architecture for Unmanned Vehicle Systems Version 2.0," 4D/RC, Gaithersburg, 2002.

[23]V. Kimon, D. Gracanin, M. Matijasevic, K. Ramesh och G. Demetriou, "Control Architecture for Autonomous Underwater Vehicles," *IEEE Control Systems,* vol. 17, nr 6, pp. 48-64, 1997.

[24]ESPLANADE Deliverable R5.1v5 – "Methods for allocating safety requirements on decision elements - State of the art", Anders Thorsén RISE, Jonas Nilsson Zenuity, Susanna Leanderson Olsson Autoliv, Håkan Sivencrona Zenuity/Qamcom, Hans-Martin Heyn Volvo AB, Kenneth Östberg RISE

[25]ESPLANADE Deliverable R5.2v4 – "Methods for allocating safety requirements on decision elements – project results", Anders Cassel Qamcom, Carl Bergenhem Qamcom, Olle Bridal Volvo AB, Anders Thorsén RISE Research Institutes of Sweden, Susanna L Olsson Veoneer, DeJiu Chen KTH