MAENAD

Grant Agreement 260057

Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles

Report type	**Deliverable D2.2.1**
Report name	**Design methodology: Methodology description for embedded systems development with EAST-ADL**

Dissemination level	PU
Status	Intermediate
Version number	1.0
Date of preparation	2012-01-30

Authors

Editor	E-mail
J. Fiedler	CON (jens.fiedler@continental-corporation.com)

Authors	E-mail
R. Librino	4SG (renato.librino@4sgroup.it)
H. Lönn	VTEC (henrik.lonn@volvo.com)
F. Stappert	CON (friedhelm.stappert@continental-corporation.com)
F. Tagliabò	CRF (fulvio.tagliabo@crf.it)
S Torchiaro	CRF (sandra.torchiaro@)crf.it
S. Voget	CON (stefan.voget@continental-corporation.com)

The Consortium

Volvo Technology Corporation (S)	4SG(I)	Centro Ricerche Fiat (I)
Continental Automotive (D)	Delphi/Mecel (S)	CEA LIST (F)
MCO (SF)	Systemite (S)	PAR (F)
Kungliga Tekniska Högskolan (S)	Technische Universität Berlin (D)	University of Hull (GB)

Revision chart and history log

Version	Date	Reason
0.1	2010-12-07	First internal release
0.1	2011-01-19	Edited by VTEC
0.5	2011-06-20	Second internal release
0.9	2011-12-14	Inclusion of the methodology
1.0	2012-01-30	Intermediate release

List of abbreviations

Abbreviation	Description
FEV	Fully Electric Vehicle
V&V	Validation & Verification
EPF	Eclipse Process Framework
BPMN	Business Process Model and Notation

Table of contents

1 Introduction

During the ATESST2 project the EAST-ADL methodology has been defined, to give guidance on the use of the language for the construction, validation and reuse of a well-connected set of development models for automotive systems.

The aim of the MAENAD project is to extend the EAST-ADL methodology for the engineering of FEV.

The following aspects will be addressed by the methodology:

- specific requirements in FEV engineering and specific applicable standards (e.g. high voltage, flammability of batteries, high current switching);
- Application of safety concepts in FEV as defined in ISO 26262, supported by EAST-ADL and novel techniques for automated fault tree analysis and FMEA;
- Application of automated techniques for ASIL decomposition;
- Application of new concepts for V&V, e.g. using behavioral simulation, fault simulation and fault injection;
- Introduction of new concepts for overall safety assessment, providing sufficient evidence of application of ISO 26262 concerning the design process and the relevant work products, including requirements capturing and modeling, completeness of safety analysis, of the safety case, and of the V&V

To define a detailed methodology based on EAST-ADL for engineering of FEV systems, using a seamless integrated approach compliant with ISO 26262 Functional Safety requirements, the following steps will be performed:

- Review of the already existing EAST-ADL methodology in terms of compliance with the last version of ISO 26262 (FDIS – Final Draft International Standard). Moreover the ISO26262 activities and work products not yet included in the EAST-ADL methodology, especially related to Functional Safety Concept and to Technical Safety Concept (ISO 26262, part3 and 4), will be identified.
- EV standards & regulations analysis: the requirements coming from EV standards and regulations will be analyzed in detail to identify the requirements to be considered relevant for MAENAD approach.
- Integration of ISO 26262 concepts and EV needs into the EAST-ADL methodology

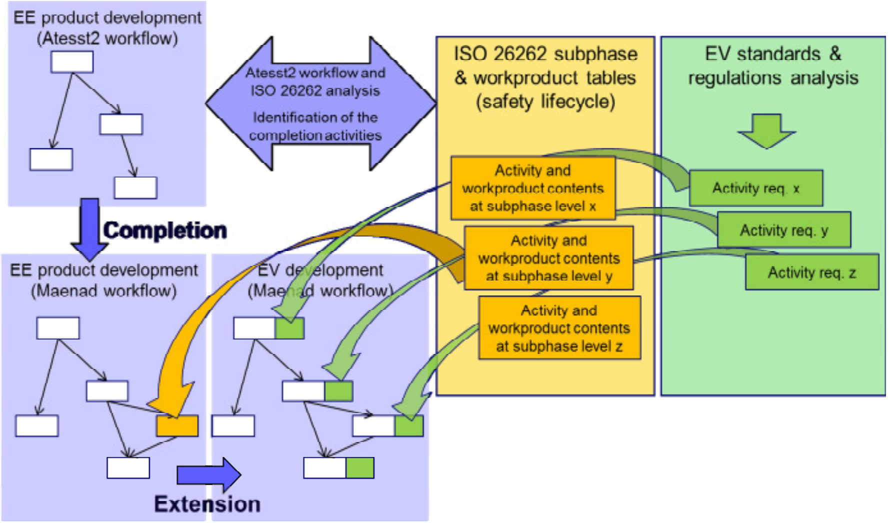

Figure 1 – Illustration of the methodology modeling process

The methodology is based on a set of elementary work tasks which are performed by a set of actors (roles) and produce a set output artifacts from a set of input artifacts. These tasks are structured into disciplines and then presented to the end user by a set of views. This leads to a highly linked network of methodological activities in which an end user can easily navigate to get information and guidance on the use of the language for particular development tasks.

Technically, modeling of the methodology itself has been done by means of the Eclipse process framework (EPF, www.eclipse.org/epf/). We are also investigating using Business Process Model and Notation (www.omg.org/spec/BPMN/2.0/). The preliminary MAENAD methodology was captured in this notation.

The MAENAD methodology is intended to be a composable methodology where activities and work products related to different aspects of development are documented separately. Examples of aspects are safety, electrical, variability and timing. This is manifest as “swimlanes” in the preliminary methodology, a concept that is being refined and validated.

The tooling used for methodology modeling allows publishing an html export as main methodological artifact for the end user.

2 Overall design process

Given the complexity of the development activities in automotive embedded software development, it is mandatory to structure the methodology so as to enable a relatively fast and easy access to the EAST-ADL language for a small kernel of essential development activities which can then be seamlessly extended to a comprehensive treatment of the language including more specialised development activities which may not necessarily be used in any development project. Hence the methodology is structured into two major components. This structuring is analogous to the structuring of the EAST-ADL language itself.

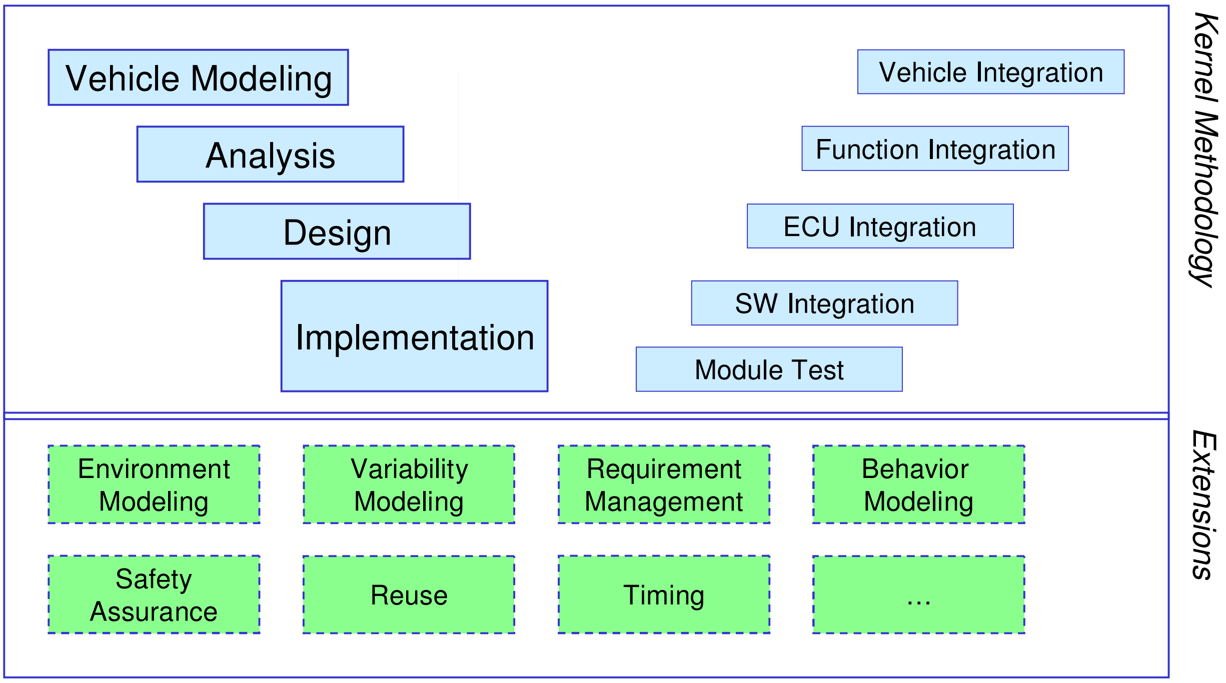

Figure 2 – EAST-ADL Structure

The main component, the kernel development part, comprises a top-down description of the central constructive phases of automotive embedded software development:

- **Vehicle Modeling:** The analysis of external requirements resulting in the construction of a top-level vehicle feature model together with the definition of necessary or intended feature configurations. In addition, for each feature a set of requirements is specified at vehicle level.
- **Analysis:** The creation of a functional analysis model specifying a solution of the requirements without concern about implementation restrictions of automotive series development. The analysis model is a logical representation of the system to be developed and its environment, and the boundary of the system to its environment. All the modeling in this phase will be on a logical behavior level, i.e. it will make no distinction between HW and SW or about the implementation of communication. Behavior may be specified in detail by executable models.
- **Design:** The creation of a functional design model specifying a solution to the requirements in terms of efficient and reusable architectures, i.e. sets of (structured) HW/SW components and their interfaces, a hardware architecture, and a mapping from functional components to HW/SW components. The architecture must satisfy the constraints of a particular development project in automotive series production.

- **Implementation:** The HW/SW implementation and configuration of the final solution. This part is mainly a reference to the concepts of AUTOSAR which provides standardized specifications at this level of automotive software development. However, the use of AUTOSAR concepts is not mandated by the methodology. Other, in particular more traditional implementation concepts can be used in this phase while leaving the other phases unchanged.

The core methodology is extended into a comprehensive methodology for automotive development projects by adding three additional and orthogonal activities to each of these phases:

- Specification of V&V cases to be executed and evaluated during the corresponding integration phase. V&V cases are most typically test cases, but can also include reviews etc.
- Verification of the model on a given abstraction level to the requirements of the model at the abstraction level directly above.
- V&V activities on the model artifacts of a given level itself, i.e. peer reviews, consistency checks, check of modeling guidelines etc.

Automotive software development also has to go through a number of integration and testing phases. While the methodology tries to be comprehensive handling the construction phases, the integration activities are only covered inasmuch they involve V&V activities and the relation to V&V-artifacts defined in the construction phases.

The second main component of the EAST-ADL methodology consists of a set of complementary loosely-coupled extensions to the core development part. Each of these extensions may be used as an add-on to the core activities. Extensions can of course also be combined depending on project needs. The following extensions are currently included:

- **Environment Modeling:** Modeling of the (typically analog or discrete-analog) environment of the system to be developed.
- **Requirements and V&V:** Detailed handling of complex requirements and V&V artifacts.
- **Safety Assurance:** Development of Safety-critical systems
- **Timing:** Detailed handling of timing requirements and properties.
- **Variability Modeling:** Detailed handling of variability modeling.
- **Behavior modeling:** Detailed handling of behavioral modeling
- EV specific modeling: Detailed handling of FEV relevant issues.

3 EAST-ADL Methodology analysis

In a first step, the existing EAST-ADL methodology, as developed in the ATESST2 project, has been analysed. The normative regulation for functional safety in the automotive domain, ISO 26262, describes the phases Safety Management and Safety Development. These phases are structured in sub-phases. Each sub-phase comprises work products, tools and the responsible role. Both process models were compared regarding functional safety. A common view is established in the appended Sheet (Excel-File). The following tables show an excerpt of this analysis.

ISO Part	Phase	Sub-phase	Work Products	Tools	Activity Responsible	LINK to Product Development Work Flow (EAST-ADL Based)	EAST-ADL artifacts
Part 2	Safety management	Overall safety management	Organization-specific rules and processes for functional safety	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of competence	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of quality management	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
		Safety management during the concept phase and the product development	Safety plan	Compatible with traceability requirements, change mangement, configuration…	Safety Manager	Vehicle Level Analysis Level Design Level	VVCase for detailed activities, SafetyCase structure for overall information structure, Requirements with Satisfy relation to Ground to detail the evidence required.
			Project plan (refined)	Not applicable	Project Manager		N/A
			Safety case	Compatible with traceability requirements, change mangement, configuration…	Safety Manager	EAST-ADL Quality+Safety Process > Analysis Phase > Functional Safety Requirements EAST-ADL Quality+Safety Process > Design Phase > Functional Safety Requirements EAST-ADL Quality+Safety Process > Analysis Phase > Safety Goals EAST-ADL Quality+Safety Process > Analysis Phase > Perform Risk Assessment Validation > Safety Goals EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Safety Goals	SafetyCase Functional Safety Requirements Safety Goals
			Functional safety assessment plan	Compatible with traceability	Safety Manager		N/A
			Confirmation measure reports	Compatible with traceability requirements, change	Safety Manager		Warrant.Evidence
		Safety management after the item's release for production	Evidence of field monitoring	Compatible with traceability requirements, change mangement, configuration…	*Persons appointed to maintain functional safety after release for production*		Warrant.Evidence

Table 1: EAST-ADL methodology elements for ISO26262 Part 2

ISO Part	Phase	Sub-phase	Work Products	Tools	Activity Responsible	LINK to Product Development Work Flow (EAST-ADL Based)	EAST-ADL artifacts
Part 2	Safety management	Overall safety management	Organization-specific rules and processes for functional safety	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of competence	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
			Evidence of quality management	Not applicable	*Product Liability Manager*	Vehicle Level Analysis Level Design Level	N/A
		Safety management during the concept phase and the product development	Safety plan	Compatible with traceability requirements, change mangement, configuration…	Safety Manager	Vehicle Level Analysis Level Design Level	VVCase for detailed activities, SafetyCase structure for overall information structure, Requirements with Satisfy relation to Ground to detail the evidence required.
			Project plan (refined)	Not applicable	Project Manager		N/A
			Safety case	Compatible with traceability requirements, change mangement, configuration…	Safety Manager	EAST-ADL Quality+Safety Process > Analysis Phase > Functional Safety Requirements EAST-ADL Quality+Safety Process > Design Phase > Functional Safety Requirements EAST-ADL Quality+Safety Process > Analysis Phase > Safety Goals EAST-ADL Quality+Safety Process > Analysis Phase > Perform Risk Assessment Validation > Safety Goals EAST-ADL Quality+Safety Process > Vehicle Phase > Perform Safety Analysis > Safety Goals EAST-ADL Quality+Safety Process > Design Phase > Safety Goals	SafetyCase Functional Safety Requirements Safety Goals
			Functional safety assessment plan	Compatible with traceability	Safety Manager		N/A
			Confirmation measure reports	Compatible with traceability requirements, change	Safety Manager		Warrant.Evidence
		Safety management after the item's release for production	Evidence of field monitoring	Compatible with traceability requirements, change mangement, configuration…	*Persons appointed to maintain functional safety after release for production*		Warrant.Evidence

Table 2: EAST-ADL methodology elements for ISO26262 Part 3

FEV specific standards

Further on, standards concerning FEV are analyzed in order to identify the requirements that should be considered relevant to MAENAD, especially those regarding E/E addressing functionality, safety, communication, thus excluding mechanics, environmental conditions, EMC, operational procedures not related to the design phase.

The following normative standards concerning FEV are used:

- SAE – J2289 Electric-Drive Battery Pack System: Functional Guidelines.
- ISO 6469-1 Electrically propelled road vehicles – Specific requirements for safety – Part 1: On board energy storage
- ISO 6469-2 Electric road vehicles – Safety specifications – Part 2: Vehicle operational safety means and protection against failures
- ISO 6469-3 Electric road vehicles – Safety specifications – Part 3: Protection of persons against electric hazards
- R.116 and subsequent amendments

The identified requirements are evaluated to define further requirements, which should be captured in MAENAD, in terms of:

- system description and modeling requirements
- methodological requirements for system design

The SAE – J2289 collects a set of requirements for the Electric-Drive Battery Pack System. These requirements are mapped in MAENAD to system description and modeling. Further requirements for the design methodology are derived.

In detail there are requirements for

- Modes and associated electrical modes
- Key on – Discharge
- Key on – Regen Operation
- Key on – Charge
- Key-Off Parked Off Plug Operating
- Parked Off Plug IDLE/Storage Operation
- Traction Wiring and Connectors Sensor Wiring
- Contactors/Disconnects
- Electrical Isolation
- Discharge Management – Performance Limits
- Charge Management
- Key-On Startup Diagnostics and Warning
- Key-On Running Diagnostics and Warning
- Service Diagnostics
- Multiplex Communication Interface
- Toxic Emissions

- Flammable Gasses

The ISO 6469-1, Part 1, collects requirements for “On board energy storage”. A detailed list is given in the following enumeration:

- The measurement of the isolation resistance of the RESS shall include auxiliary components located inside the RESS housing, e.g. monitoring or temperature-conditioning devices and liquid fluids (if any).
- Heat generation under any first-failure condition, which could form a hazard to persons, shall be prevented by appropriate measures, e.g. based on monitoring of current, voltage or temperature.
- RESS over-current interruption: If a RESS system is not short-circuit proof in itself, a RESS over-current interruption device shall open the RESS circuit under conditions specified by the vehicle and/or RESS manufacturer, to prevent dangerous effects for persons, the vehicle and the environment.

The ISO 6469-2, Part 2, collects “Vehicle operational safety means and protection against failures”. The following enumeration is given:

- Electric road vehicles - Safety specifications - Part 2: Functional safety means and protection against failures
- Operational safety -Connection of the vehicle to an off-board electric power supply
- Operational safety – Driving - Indication of low energy content of RESS
- Operational safety - Driving backwards
- Operational safety – Parking
- Protection against failures

The ISO 6469-3, Part 3 focuses “Protection of persons against electric hazards”. Also Safety requirements are described regarding:

- Measures and requirements for protection of persons against electric shock - Protection under first failure conditions
- Measures and requirements for protection of persons against electric shock - Alternative approach for protection against electric shock
- Measures and requirements for protection of persons against electric shock - Isolation resistance requirements
- Measures and requirements for protection of persons against electric shock - Requirements of potential equalization
- Requirements for vehicle charging inlet - Voltage decrease requirement
- Requirements for vehicle charging inlet - Grounding and isolation resistance requirement for charging inlet

4 MAENAD Methodology

The MAENAD methodology is modelled in BPMN2.0 using the open source Eclipse based tool ADONIS. The methodology itself is attached as an HTML export from the ADONIS tool to this document as an appendix. This chapter describes the structure and basic modelling principles of the methodology.

4.1 Methodology modelling principles

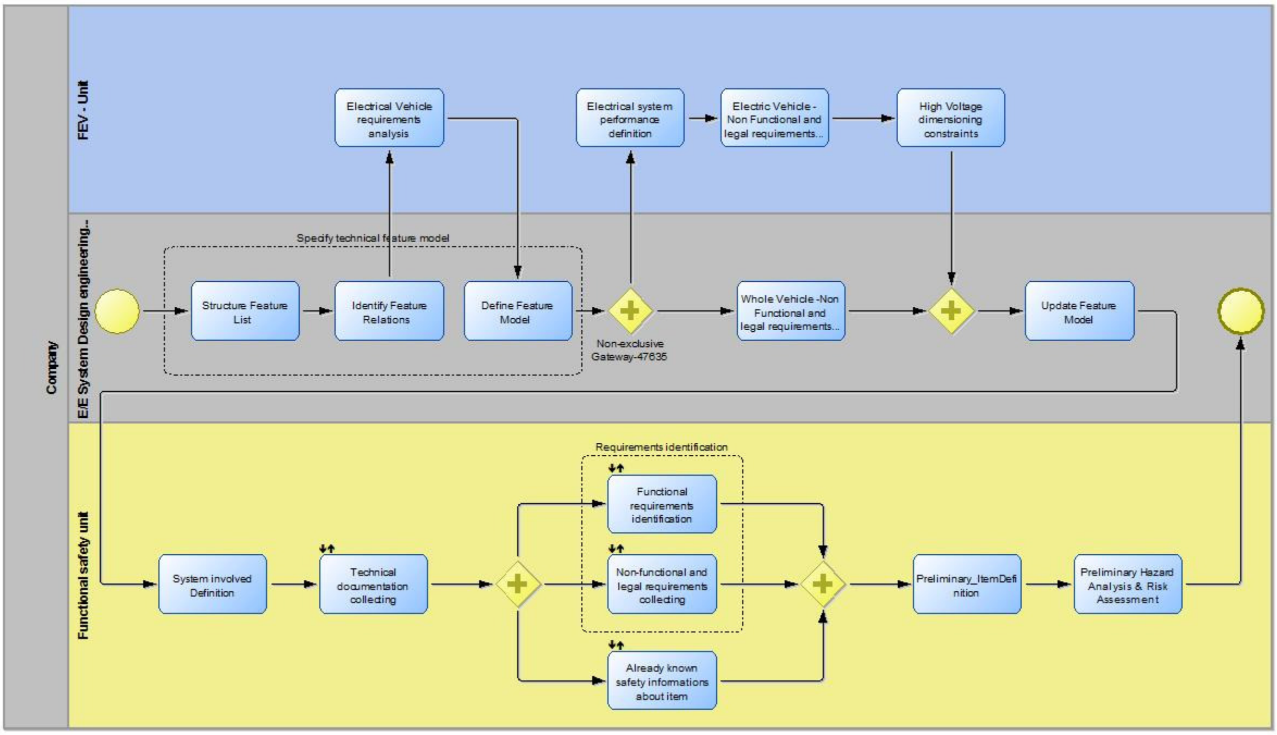
Company
FEV - Unit
E/E System Design engineering..
Functional safety unit
Electrical Vehicle requirements analysis
Electrical system performance definition
Electric Vehicle - Non Functional and legal requirements...
High Voltage dimensioning constraints
Specify technical feature model
Structure Feature List
Identify Feature Relations
Define Feature Model
Non-exclusive Gateway-47635
Whole Vehicle -Non Functional and legal requirements...
Update Feature Model
Requirements identification
Functional requirements identification
Non-functional and legal requirements collecting
Already known safety informations about item
System involved Definition
Technical documentation collecting
Preliminary_ItemDefinition
Preliminary Hazard Analysis & Risk Assessment

Figure 3: Example from methodology to illustrate the methodology modeling principles

The methodology is modeled in “swimlanes”. The core development methodology leading a developer through the EAST-ADL language is modeled in the “E/E System Design Engineering – Conventional Vehicle Unit” lane. Specific aspects that extend the core methodology are separated to additional lanes, e.g. in the example “FEV-Unit” and “Functional safety unit”.

Such additional lanes can be included or excluded in a process regarding the needs of a specific project. E.g. in case of a FEV vehicle, at least the FEV-Unit would be added to the process. In case of creation of a process instance,

1. The appropriate swimlanes have to be selected, and
2. The links that go beyond the borders of two or more swimlanes have to be resolved.

Example for case 2: Let’s consider in Figure 3 the process chain “Identify Feature Relations” -> “Electrical Vehicle requirements analysis” -> “Define Feature Model”. In case a conventional vehicle is developed, the FEV-Unit swimlane is not selected. As a consequence the considered path has to be reduced to “Identify Feature Relations” -> “Define Feature Model”.

Notations used from BPMN:

Yellow circle: start or end node of a methodology area.

Blue rectangle: action

Yellow rhombus: start or end of a branch

4.2 Basic structure

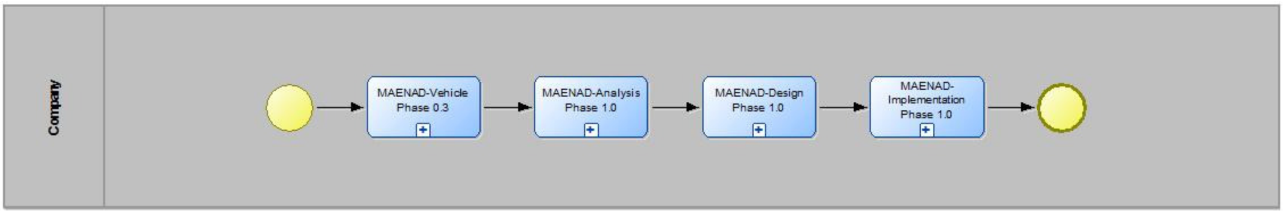

Figure 4 – Upper level of MAENAD methodology

Figure 4 gives an overview about the most abstract view on the methodology. The methodology follows one to one the abstraction level principle of the EAST-ADL language, starting from the most abstract level, the vehicle phase, to the most concrete level, the implementation phase.

As usual the methodology shows a forward process oriented view only. Iterations are not illustrated in the methodology. This is reserved to the process instantiation in a concrete project.

5 Definition of the Design Methodology of FEVs

As reported in D2.1.1, a process was followed to define the requirements related to FEV development, in order:

- to verify the capability of the current version of EAST-ADL2 to cover the needs related to specific characteristics of FEVs, and to extend its features if necessary; and similarly,
- to verify the capability of the analysis tools and to give inputs to adapt or, possibly, create specific tools to perform the necessary analyses;
- to define an extension of the basic E/E system development methodology resulted from ATTEST2, in order to help designers to perform the development activities required by the standards and the regulations, or those compliant to best practices or engineering needs for EV development.

Therefore, through a sequence of activities according to a bottom-up approach, three categories of requirements have been defined: language requirements, analysis requirements, and methodology requirements.

The requirements defined have been reported in an Excel sheet and, subsequently, in Enterprise Architect, to comply with the method followed for the collection of MAENAD requirements, thus allowing better traceability, uniform categorization, assignment to WPs.

The following table is an excerpt of the Excel file and includes only the methodology requirements.

Reference are given to the requirement codes used in EA; the field “subject” has been introduced to better identify the related engineering topic and to establish a link with the language and analysis requirements related to the same topic.

It has to be pointed out that in the following table some language requirements are referred to a specific standard or regulation. However, the requirements, in some cases, can be referred to similar standards (not mentioned here, but only in the Excel sheet, which gives a more global view of the analysis conducted to define the requirements).

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
4SG 7	EV safety standards/ ISO 6469-1	Insulation	- Deployment of insulation resistance - Addressing insulation monitoring system - Hazard analysis and risk assessment concerning insulation monitoring - Design issues concerning recharging (grounding, communication) - Test planning concerning insulation - Production, operation and maintenance requirements during design phase (ISO 26262-4)	4SG78
		Heath generation	Designing a monitoring system to prevent dangerous effects to persons, in the case of failures producing heat generation	4SG79
		RESS over-current interruption	- Designing an over-current interruption device - Hazard analysis in the case of short circuit of RESS - Planning of short circuit test	4SG82
4SG 8	EV safety standards/ ISO 6469-2	Connection of the vehicle to an off-board electric power supply	Designing a means to make impossible to move the vehicle when connected to off-board electric power supply and charged by the user	4SG83
		Indication of reduced power	Designing a warning to signal to the driver that the propulsion power is reduced, in the case this is done	4SG84
		Driving backwards	Designing means to prevent unintentional switching in reverse when the vehicle is in motion (two options are available)	4SG85
		Parking	Designing a warning to indicate whether propulsion is in the driving–enable mode, when user leaves the vehicle. Designing a safety mechanism to prevent unexpected movements.	4SG86
		Protection against failures	In functional safety development, include unintended acceleration, deceleration and reverse motion as hazards to be prevented or minimized.	4SG87

First level user requirements		Second level user requirements Design methodology requirements		
Code	Title	Subject	Requirement description	Code
4SG 9	EV safety standards/ ISO 6469-3 Protection of persons against electric hazards	Protection of persons against electric shock	Designing mechanical and electronics means according to the standard. Verification planning for measures protection (design verification, test plan)	4SG88
		Alternative approach for protection against electric shock	Conduct an appropriate hazard analysis with respect to electric shock and establish a set of measures which give sufficient protection against electric shock	4SG89
		Isolation resistance requirements	Assignment of insulation resistance to high voltage components as to achieve the overall insulation resistance (dc, ac cases).	4SG90
		Requirements of potential equalization	Designing insulation barriers and bonded conductive equalization barriers. Planning verification of barriers, including bond testing.	4SG92
		Charging inlet disconnection	Designing charge system, as to ensure voltage decrease of inlet according to time requirements. Verification by simulation, analysis and testing.	4SG94
		Grounding and isolation resistance requirement for charging inlet	Designing charging system as to meet insulation requirements in the case of ac and ac inlet.	4SG95
4SG 16	EV safety standards/ EN 61851	Types of EV connection	- Define the charging system according to one of the 4 charging modes. - Define the control pilot mandatory and optional functions (modes 2-4), including charging operation states.	4SG96
		Protection against electric shock	Define and provide measures to prevent electric shock both in normal service and in case of fault.	4SG97
		Stored energy – discharge of capacitors	Design the EV voltage input in such a way to control the voltage decay after EV disconnection	4SG99
		Detection of the electrical continuity of the protective conductor	Design a monitoring system to detect the electrical continuity of the protective conductor during charging modes 2, 3 and 4.	4SG100

First level user requirements		Second level user requirements Design methodology requirements		
Code	Title	Subject	Requirement description	Code
		Dielectric withstand voltage	Design the on board charging equipment as to withstand the test voltage at any input connection (2U +1000 V, min. 1500 V ac). Design all vehicle equipment as to withstand a test voltage of 4kV between ac or dc input and low voltage inputs (if any).	4SG101
		Electric vehicle insulation resistance	Verify the insulation resistance (by analysis and testing). Minimum required: 1 Mohm.	4SG102
		Drive train interlock	Design a system to detect the connection of the mobile connector or that the plug and the cable have been stored in the vehicle. The system shall also inhibit the drive train	4SG103
4SG 18	EV safety standards/ J2289	Vehicle operational modes	- Defining the vehicle operational modes - Justify possible discrepancies	4SG104
		Key-on discharge	- Assessment of battery capability to match the vehicle demand (range, supply of auxiliary equipment) - Designing means to detect and limit the overdischarge of individual cells - Providing fault protection devices (fuses, fast contactors)	4SG107
		Key-on Regen operation	- Assessing the compliance of the voltage with the limits during regeneration - Providing design means to avoid drive component overvoltage occurrence during regeneration - Verifying the compliance with current and voltage profiles - Providing design means to limit battery current and voltage during regeneration according to the specified profiles	4SG110
		Key on – Charge	- Verifying that all charge system components match w.r.t. electrical characteristics - Designing charge algorithm with the battery supplier	4SG113
		Key-Off Parked Off Plug Operating	- Providing energy management to prevent excessive discharge due to vehicle equipment operating in key-off mode	4SG116

First level user requirements		Second level user requirements Design methodology requirements		
Code	**Title**	**Subject**	**Requirement description**	**Code**
			- Verify energy behavior in key-off mode by simulation/calculation - Designing charge algorithm with the battery supplier	
		Parked Off Plug IDLE/Storage Operation	Designing a battery disconnect system for operation during storage or maintenance	4SG118
			- Designing contactor operation as to be deactivated in the case of crash or isolation fault - Designing disconnect system for added safety during service or by first responders during accidents.	4SG119
		Discharge management - Performance limits	Designing BMS to protect for over-temperature, under-temperature, over-current	4SG121
		Charge management	Design communication in compliance with SAE J1772, SAE J1773, and SAE J2293	4SG122
		Key-on startup diagnostics and warning	Design key-on running diagnostics and warning procedures	4SG124
		Service diagnostics	Design service diagnostics	4SG125
		Toxic emissions Flammable gasses	Consider toxic emissions and flammable gasses caused by battery damages	4SG126
4SG 72	FMVSS No. 114 Theft protection	Key-locking device	Design the key-locking system to prevent the activation of the motor and steering or self-mobility (or both)	4SG128
		Parking function	- Design the operation of key-locking system according to the standard (see interaction with park command). - Verify (by calculation and testing) that the maximum movement of the vehicle when locked is less than the max. allowable limit.	4SG129
4SG 73	FMVSS No. 102 Transmission shift lever sequence, starter interlock, and transmission braking effect		Designing the shift lever according to the sequence position and rotation requirements	4SG130
4SG 75	R 116 Theft protection	Locking device	Designing devices to prevent unauthorized use (deactivation of engine in combination with a system to lock	4SG131

First level user requirements		Second level user requirements Design methodology requirements		
Code	Title	Subject	Requirement description	Code
			other vehicle functions, see regulation)	
		Locking function	Conduct functional safety analyses to cover the devices intended to prevents unauthorized use	4SG132
4SG 71	FMVSS No. 135 Passenger car brake systems	Regenerative braking system	- Plan the analysis and the development of braking system according to the operation mode of the RBS: control of RBS by ABS (if RBS is always active, also in neutral without any means to disconnect it by the driver, RBS is part of the service braking system). - Item definition: consider the interactions between RBS and ABS (w.r.t. interfacing and system definition in ISO 26262)	4SG133
		Diagnostics and warning	- Include diagnostics task related to RBS, in order to transmit information to the visual warning indicator - Design proper warning in the case of failure of brake power supply, reduced SoC, RBS failure	4SG135
		Braking performance	Plan a braking test in depleted battery state-of-charge condition	4SG137
4SG 19	EV performance standards/ ISO 8715	Performance testing - Test conditions and procedures	Include the simulation of vehicle performance according to test conditions and test procedure requirements Include vehicle performance testing according to test condition and test procedure requirements	4SG141
4SG 20	EV performance standards/ ISO 8714	Energy and range testing - Test conditions and procedures	Include the simulation of vehicle performance according to test conditions and test procedure requirements Include vehicle performance testing according to test condition and test procedure requirements	4SG145
4SG 23	EV performance standards/ ISO 12405-2	Test sequence - Test conditions	- Simulate vehicle performance according to test conditions requirements(when applicable) - Test vehicle performance according to test conditions requirements	4SG148
4SG 74	SAE J2777 Conductive charge coupler	Control pilot	Design the communication according to the standard (charging station status, power level, fault conditions)	4SG151

First level user requirements		Second level user requirements Design methodology requirements		
Code	Title	Subject	Requirement description	Code
		Proximity detection	Design the management of the connector detection signal: to start charge control, to engage drive train interlock, to reduce charge load during disconnection	4SG152
		Charge management	Design the charging state machine according to the standard, including safe states in the case of fault.	4SG153
		Charge status indicator	Define the charge status indicator, including diagnostic functions.	4SG154
4SG 70	R 13H Braking	Phasing of braking sources (B category)	If the RBS is part of service brake, design the braking inputs, compensating the variations of the regenerative braking and ensuring breaking action in all wheels.	4SG156
		Integration with ABS	Include a development task to define and manage the interaction between ABS and RBS.	4SG157

Table 3: Methodology Requirements

In order to define some FEV design processes addressing the different subjects covered by the standards and regulations, the design methodology requirements have been analyzed, so as to identify the design activities that shall be performed according to the standards and regulations. The design phases considered are related only to E/E systems, but include also the planning of test activities, whenever the planning should be performed during the design phase, also according to ISO 26262.

The following processes have been defined:

- Design of an insulation monitoring system
- Design the Regenerative Energy Storage System
- Design of the Regenerative Braking System
- Design of conductive charge coupling
- Design of the vehicle operation modes
- Design of theft protection system

The figures hereafter show the highest level representation of the design processes, while the detailed description at lower level in terms of subprocesses and activities is reported using the tool Adonis. In order to provide a useful guideline to FEV designers, the textual description of the subprocesses and of the activities include the reference to the standards and regulations. It should be pointed out that some activities refer to more than one standards or regulations. Designers should identify the applicable standards and regulations according to the specific system under development or the legislative constraints.

It has to be pointed out that all the above design processes are FEV specific. The last one (Design of theft protection system) is also EV specific, because it is intended to ensure the safety of EVs by preventing the unauthorized use of FEVs, which can be dangerous.

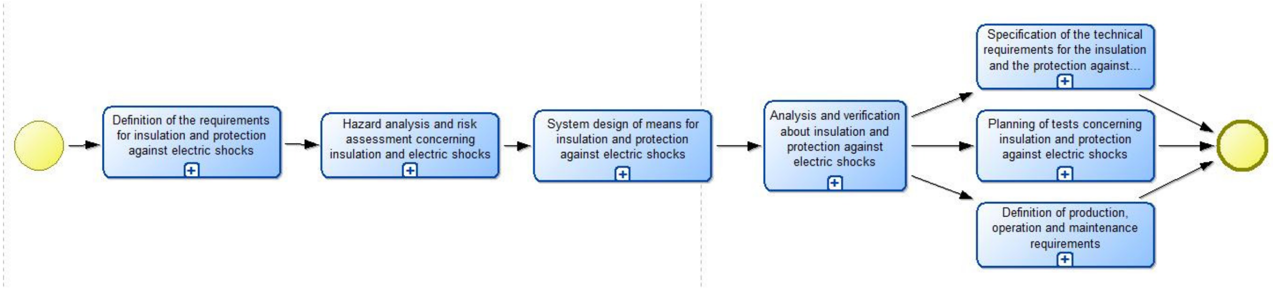
Definition of the requirements for insulation and protection against electric shocks
Hazard analysis and risk assessment concerning insulation and electric shocks
System design of means for insulation and protection against electric shocks
Analysis and verification about insulation and protection against electric shocks
Specification of the technical requirements for the insulation and the protection against...
Planning of tests concerning insulation and protection against electric shocks
Definition of production, operation and maintenance requirements

Figure 5 – Design of an insulation monitoring system

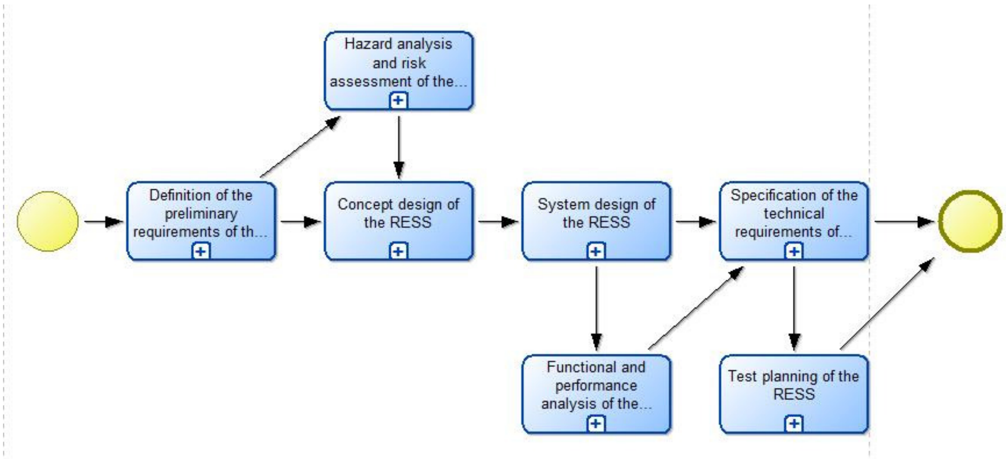
Hazard analysis and risk assessment of the...
Definition of the preliminary requirements of th...
Concept design of the RESS
System design of the RESS
Specification of the technical requirements of...
Functional and performance analysis of the...
Test planning of the RESS

Figure 6 – Design the Regenerative Energy Storage System

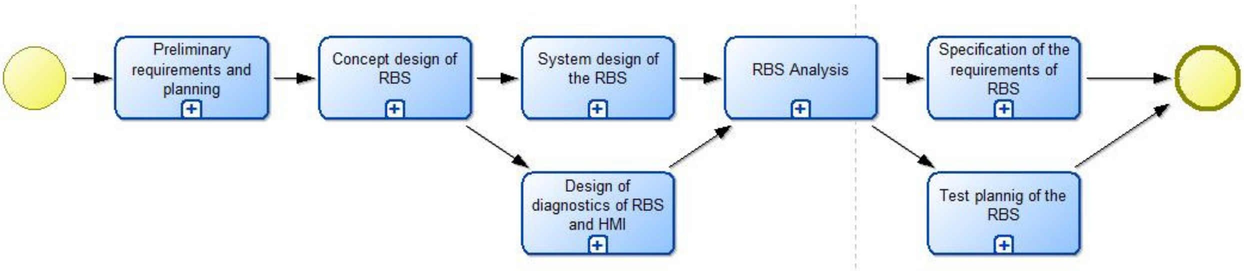
Preliminary requirements and planning
Concept design of RBS
System design of the RBS
RBS Analysis
Specification of the requirements of RBS
Design of diagnostics of RBS and HMI
Test plannig of the RBS

Figure 7 – Design of the Regenerative Braking System

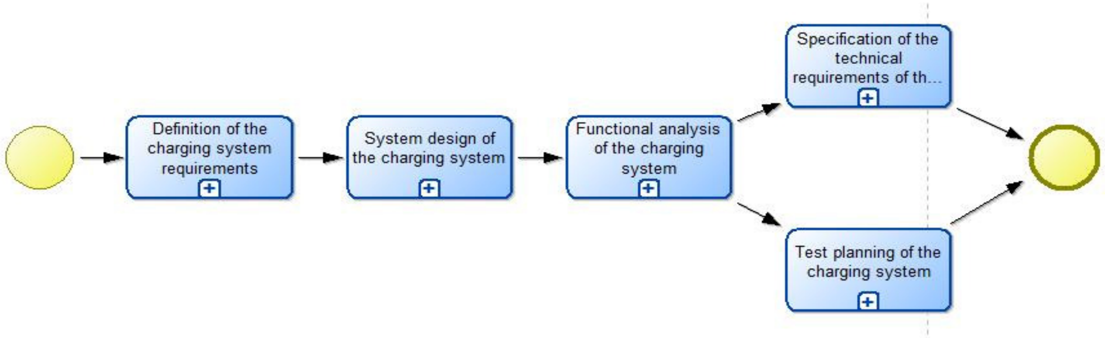
Definition of the charging system requirements
System design of the charging system
Functional analysis of the charging system
Specification of the technical requirements of th...
Test planning of the charging system

Figure 8 – Design of conductive charge coupling

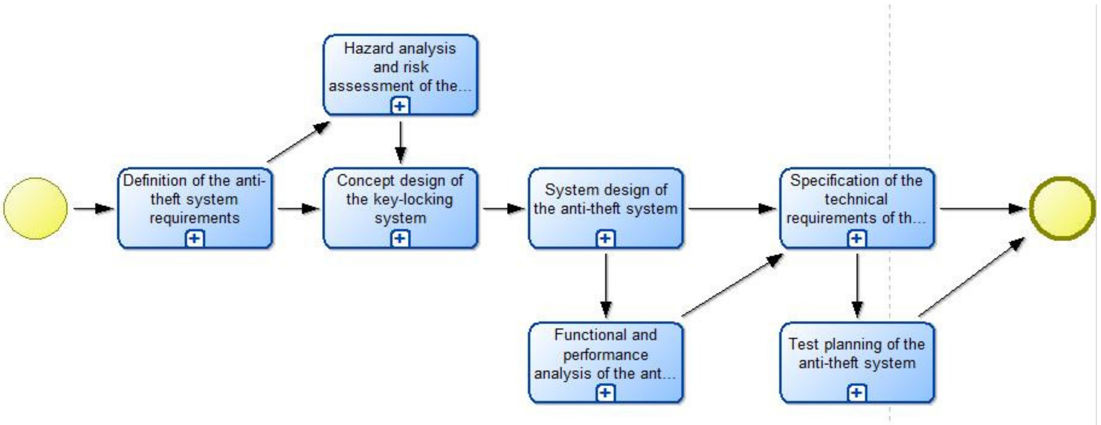
Hazard analysis and risk assessment of the...
Definition of the anti-theft system requirements
Concept design of the key-locking system
System design of the anti-theft system
Specification of the technical requirements of th...
Functional and performance analysis of the ant...
Test planning of the anti-theft system

Figure 9 – Design of theft protection system

To complete the definition of the FEV design methodology, two further steps are envisaged:

- The allocation of the above processes in the general FEV development process, which will be represented on a separate swimlane linked to the general development process (GMP model), i.e. by identifying the appropriate development phases (vehicle level, analysis level, design level, implementation level).
- The refinement of the FEV design processes, in order to structure it as much as possible in compliance with the general activity sequence that is the reference structure of each GMP phase.

6 References

[1] ATESST2 Deliverable D5.1.1 Methodology guideline when using EAST-ADL2, June 2010.

[2] ISO 26262: Road Vehicle – Functional Safety standard

[3] Maenad_Deliverable_D2.2.1_Appendix.zip: Integrated MAENAD Methodology

7 Appendix ISO26262 Requirements

7.1 Vehicle level modeling

7.1.1 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 4.4.1:

"*The functional requirements of the item as well as the dependencies between the item and its environment shall be available. This information includes the following:*

a) Purpose and functionality of the item;

b) Non-functional requirements, e.g. operational and environmental requirements and constraints, if available;

c) Legal requirements (especially laws and regulations), national and international standards, if already known."

Recommendation: This ISO clause is concerned with an early development phase in which the starting point for the functional safety work is defined in the form of an item definition. The topics addressed in the clause should mainly be included in the modeling at the vehicle level, although some specific aspects might be more appropriately addressed at lower modeling levels (i.e. analysis, design or implementation). Checklists for the different levels can be defined where a), b) and c) above are explicitly included in the respective checklist.

Derived Requirements:

The purpose and functionality of Item shall be defined by means of an Item's Feature(s) and its requirements

7.1.2 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 4.4.3:

"*It shall be ensured that the boundary of the item and the item's interfaces, as well as assumptions concerning other items and elements are determined by considering the following:*

a) Elements of the item;

b) Assumptions concerning the effects of the item's behavior on other items or elements, i.e. the environment of the item, including interactions;

c) Requirements from other items, elements and environment on the item;

d) Requirements of the item on other items, elements and environment; and

e) Allocation and distribution of functions among the items and elements involved.

f) Operating scenarios of the item shall be mentioned if those impact the functionality of the item"

Recommendation: This ISO clause is concerned with an early development phase in which the starting point for the functional safety work is defined in the form of an item definition. The topics addressed in the ISO 26262 requirement above should be included in the modeling at the vehicle level. To support this modeling, a checklist can be defined where a) - f) above are explicitly included in the checklist. However, it does not seem to be appropriate to consider item-internal elements (as indicated in a) and partly in e) above) at this stage.

Derived Requirements:

The elements of the item shall be defined in terms of functional elements on Analysis Level which realize the Item’s Features

The elements of the item shall be defined in terms of functional and resource elements on Design Level which realize the Item’s Features

The elements of the item shall be defined in terms of software and hardware elements on Implementation Level which realize the Item’s Features

The effects of the item's behavior on other items or elements shall be defined through the interface definitions of the elements of the item on Analysis, Design and Implementation level

Requirements of the item on other items, elements and environments shall be defined through the output interface definition of the Item’s elements on Analysis, Design and Implementation level

Requirements from other items, elements and environment on the item shall be defined through the input interface definition of the Item’s elements on Analysis, Design and Implementation level

The item’s functionality shall be realized by elements on Analysis, Design and Implementation level.

Elements on Analysis, Design and Implementation level which realize an item shall be linked to the Item’s features with a Realize relation.

Operating scenarios of the item shall be defined in terms of traffic and environment (operating situations) and operational situation (use cases)

7.1.3 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 6.4.1:

"*The hazard analysis and risk assessment shall be based on the item definition*."

Recommendation: The requirement itself is not particularly applicable to model-based development (although it could be included verbatim in a checklist for how to perform the hazard analysis). More importantly however, the requirement implies that traceability should exist between the item definition and the hazard analysis. It is therefore highly desirable that the modeling incorporates such traceability, preferably in both directions. The applicable checklists could support this by explicitly requiring traceability.

Derived Requirements:

Hazard analysis shall be performed for each Item and represented through Hazards, Hazardous Events and related elements.

7.1.4 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 6.4.2-6.4.6:

In these clauses, ISO 26262 gives several requirements on how to perform the hazard analysis (and what is somewhat inadequately called "risk assessment").

Recommendation: We assume that the hazard analysis itself is performed outside the tool environment for model-based development. Thus, the requirements in ISO 26262 on how to perform this analysis is out-of-scope for these guidelines. However, the results of the hazard analysis should be represented in the models by being linked to the corresponding systems. These results include:

- the identified hazards, preferably expressed as inabilities of the considered system to operate as intended
- the operational situations and operating modes for which the hazards could lead to harm
- the ASIL associated with each identified hazard

It is important that hazards are defined in an appropriate way. They should be defined so that they are fully within the scope of the considered system. A hazard should not be defined so that it can only occur when certain environmental conditions are fulfilled. For example, "*the airbag will not be activated if an airbag-relevant collision occurs*" is a good example of a hazard. The hazard itself can exist independently of the driving situation even though it is only in an airbag-relevant collision that the hazard would really have an effect. In other words, the hazard can exist even if there is no collision. A less appropriate hazard formulation would be "*the vehicle is involved in an airbag relevant collision but the airbag is not activated*". This situation can only occur when there is a collision so it is not independent of the driving situation. In fact, this second example is a 'hazardous event' rather than a hazard in the ISO 26262 terminology.

Derived Requirements:

All identified Hazards shall be represented as Hazards and linked to the Item

Hazardous Events shall be defined and its corresponding operational situation.

7.1.5 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 6.4.8:

"*A safety goal shall be formulated for each hazardous event evaluated in the hazard analysis.*

e) An ASIL shall be assigned to each safety goal.

f) If similar safety goals are determined, these can be combined into one safety goal.

g) If different ASILs are assigned to similar safety goals combined in a single one according to b), the highest ASIL shall be assigned to the combined safety goal.

h) For each safety goal, there shall be a requirement that specifies a safe state that achieves the safety goal, if this safety goal can be achieved by transitioning to a particular state.

i) The safety goals together with their attributes (ASIL, safe state, if applicable) shall be specified according to ISO 26262-8, Clause 5."

Recommendation: Although the ISO requirement states that a safety goal shall be formulated for each hazardous event, in most (and possibly all) cases it makes more sense to formulate one safety goal for each hazard. In fact, a typical safety goal is simply a statement that a given hazard shall not occur. Together with the ASIL determined for the corresponding hazard, a safety goal constitutes a top-level requirement in the functional safety hierarchy. Thus, each safety goal and its associated ASIL should be represented in the requirements model if such a model is indeed created. This could be further supported by a requirements modeling checklist that explicitly states that safety goals and associated ASILs shall be represented in the requirements model and that these shall be identifiable as safety goals in this model.

Regarding the details of the ISO requirement, the following can be noted:

- Subclause f and g will rarely be applicable, assuming that safety goals are defined per hazard and hazards are expressed as specific inabilities of the considered system to operate as intended.
- Subclause h is quite unnecessary here from a strictly logical viewpoint. It should be considered in the functional safety concept and not in the safety goal formulation. However, if compliance with ISO 26262 is an absolute requirement associating a safe state with each safety goal (when applicable) is not a difficult task.
- Subclause i deals with requirements management and is addressed elsewhere in these guidelines.

Derived Requirements:

Each Hazardous Event shall have one associated Safety Goal

There shall be one safe state defined using the safe state attribute of the Safety Goal

7.2 Analysis level modeling

7.2.1 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.3.2:

"At least one functional safety requirement shall be specified for each safety goal."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation along the lines of "is every safety goal linked to at least one functional safety requirement?"

Furthermore, the functional safety requirements (as defined in ISO 26262) should be identifiable as functional safety requirements in the requirements model.

Derived Requirements:

One or several requirements shall be defined for each Safety Goal and be associated to a FunctionalSafetyConcept requirements container with role functional safety requirement.

7.2.2 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.3.3:

"*Each functional safety requirement shall be specified considering the following information, if applicable:*

a) Operating modes;

b) Fault tolerant time spans;

c) Safe states, if this requirement can be met by transitioning to a particular state;

d) Emergency operation times, and

e) Functional redundancies (e.g. fault tolerance)."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation along the lines of "has the following issues a-e been considered in the specification of each functional safety requirement?"

Derived Requirements:

For each functional safety requirement, the following information shall be defined, where applicable:

a) Operating modes- defined as associated modes indicating when the functional safety requirement is valid;

b) Fault tolerant time spans - defined in the requirement text or as a derived requirement

c) Safe states, if this requirement can be met by transitioning to a particular state - defined in the requirement text or as a derived requirement

d) Emergency operation times- defined in the requirement text or as a derived requirement

e) Functional redundancies (e.g. fault tolerance) - defined in the requirement text or as a derived requirement

7.2.3 N587_Rework_BL11_Part_3_2009-01-22.doc, clauses 7.4.3.4-7.4.3.7:

These ISO 26262 requirements specify some aspects that should be covered in the technical safety requirements: warning and degradation concept, emergency operation, assumptions on the actions of the driver or other involved people.

Recommendation: The ISO requirements can easily be translated into specific questions in a requirements management checklist: "Has the warning and degradation concept been specified?", etc. This checklist can be applied to the requirements model in a project to check whether the ISO requirements are met or not.

Derived Requirements:

A warning and back-up concept shall be specified using architectural elements on Analysis Level.

An emergency operation shall be specified using architectural elements on Analysis Level, unless a safe state can be reached by immediate switching off

Assumptions made on the necessary actions of the driver or other endangered persons in order to comply with the safety goals shall be represented as requirements on the Environment model, and possibly also behavioral models.

7.2.4 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.4.1:

"A safety architecture concept shall be developed."

Recommendation: The safety architecture concept represents the conceptual architecture of the system in terms of architectural provisions to ensure functional safety. Thus, the safety architecture concept includes redundancy principles such as replication of components (for example more than one sensor to measure a physical quantity), monitoring of a system element by another system element, activation of an actuator only when two system elements agree that such an activation shall be made, etc. The safety architecture concept can be represented by one or more block diagrams that show the redundancy principles. For the modeling, a checklist can be defined that includes the simple question "is the safety architecture concept represented in a model?"

Derived Requirements:

A safety architecture concept shall be specified using architectural elements on Analysis Level.

7.2.5 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.4.2:

"The functional safety requirements shall be allocated:

a) The allocation of functional safety requirements shall be based on the elements of the preliminary architectural assumptions of the item.

b) In the course of allocation, the ASIL and the information given in 7.4.3.3 shall be inherited from the previous level of detail.

c) If several functional safety requirements are allocated to the same architectural element, then the architectural element shall be developed according to the highest ASIL among these requirements.

d) If the item comprises more than one system, the functional safety requirements for the individual systems and their interfaces shall be derived from the functional safety requirements considering the preliminary system architecture assumptions, and these functional safety require-

ments shall be allocated to the systems.

e) The allocation of the functional safety requirements may be performed by applying the ASIL decomposition for the purpose of tailoring the ASIL. If ASIL decomposition is applied, it shall be applied according to ISO 262652-9, Clause 4."

Recommendation: The allocation of functional safety requirements should be visible in the modeling at the analysis level. This could be highlighted in a checklist for the analysis modeling. ASIL issues should be handled as indicated in the ISO requirement and this could also be highlighted in a modeling checklist.

(A detailed guideline for how to address ASIL issues as described above could be defined, but this is not done in this report since such a guideline would depend on)

Derived Requirements:

Functional Safety Requirements, i.e. Requirements in the Functional Safety Concept shall be associated to elements on Analysis level through the Satisfy association.

The ASIL of a Functional Safety Requirement shall be defined using the ASIL attribute of a SafetyConstraint associated to the requirement with a Refine relationship.

Each SafetyConstraint shall be associated to a FaultFailure. The FaultFailure defines the failure mode which is to be avoided at the integrity level according to the SafetyConstraint's ASIL attribute.

7.2.6 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.6:

"The functional safety requirements shall be verified according to ISO 26262-8, Clause 8 for consistency and compliance with the safety goals."

Recommendation: A checklist for the requirements modeling should include the need for verification of the compliance between functional safety requirements and safety goals, with explicit mentioning of applicable verification techniques like inspection, walkthrough and formal methods.

Derived Requirements:

Checklist.

7.2.7 N587_Rework_BL11_Part_3_2009-01-22.doc, clause 7.4.8:

"Criteria for safety validation of the item shall be specified in the functional safety concept."

A later draft of ISO 26262 is clearer, stating that: "The acceptance criteria for safety validation of the item shall be specified based on the functional safety requirements"

Recommendation: Assuming that validation is somehow represented by models, the acceptance criteria should be represented in such models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each Functional Safety Requirement shall be linked to a VVProcedure with a Verify association.

Each Functional Safety Requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

7.3 Design level modeling

7.3.1 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.9:

"The technical safety concept shall specify safety-related functional and safety-related non-functional dependencies between systems or elements of the item and between the item and other systems."

Recommendation: No recommendation can be given since the meaning and purpose of this requirement is not clear. However, the requirement seems to be associated with design level modeling and may possibly be relevant for modeling.

Derived Requirements:

Dependencies between different parts of the functional design architecture shall be represented by the interface definitions.

7.3.2 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.10

"The technical safety requirements shall specify requirements on safety mechanisms (see also ISO 26262-8, 5.4) including:

a) Measures related to the detection, indication and control of faults in the system itself (self-monitoring of the system);

b) Measures related to the detection, indication and control of faults in external devices interacting with the system;

c) Measures that enable the system to achieve and/or maintain a safe state;

d) Measures to detail and implement the warning and back-up concept; and

e) Measures related to tests of the above mentioned measures during power up (pre-drive checks), operation, power down (post-drive checks) and in maintenance."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like "Have technical safety requirements concerning measures related to... been specified?" (See a-e in the ISO requirement.)

Furthermore, the technical safety requirements (as defined in ISO 26262) should be identifiable as technical safety requirements in the requirements model.

Derived Requirements:

Technical Safety Requirements shall be defined as requirements that are associated to a TechnicalSafetyConcept requirements container with role technical safety requirement.

7.3.3 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.11

"Validation criteria concerning functional safety of the item shall be specified"

A later draft of ISO 26262 is clearer, stating that "The criteria for safety validation of the item shall be refined based on the technical safety requirements."

Recommendation: Assuming that validation is somehow represented by models, the acceptance criteria should be represented in such models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each Technical Safety Requirement shall be linked to a VVProcedure with a Verify association.

Each Technical Safety Requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

7.3.4 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.12

"For each safety mechanism that enables an item to achieve and/or maintain a safe state the following shall be specified:

a) The transition to the safe state including any assumptions regarding how actuators need to be controlled in order to achieve a safe state;

b) The fault-tolerant time interval;

c) The emergency operation time interval if the safe state cannot be reached by immediate switching off;

d) The maintenance of the safe state"

Recommendation: This requirement should be represented in a checklist to be used in the design level modeling ("For each safety mechanism represented in a design model, have the following been specified?...").

Derived Requirements:

Checklist.

7.3.5 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.13.1:

"Safety mechanisms dedicated to prevent faults from being latent shall be specified, if applicable."

Recommendation: This requirement should be represented in a checklist to be used in the design level modeling ("Are safety mechanisms for prevention of latent faults part of the design? ")

Derived Requirements:

Checklist.

7.3.6 N599_ISO_CD_26262-4_BL12_V1.doc, clause 5.4.16:

"The technical safety concept shall be verified to show consistency with the functional safety concept and the preliminary architectural design (see also ISO 26262-8, 8.4)."

Recommendation: A checklist for the requirements modeling should include the need for verification of the technical safety requirements with respect to consistency with the functional safety concept and the preliminary architectural design, with explicit mentioning of applicable verification techniques like inspection, walkthrough and formal methods.

Derived Requirements:

Requirements in a TechnicalSafetyConcept shall be derived from Requirements in a Functional-

SafetyConcept and linked with a Derived association.

A TechnicalSafetyConcept shall be defined in the FunctionalDesignArchitecture to realize the FunctionalSafetyConcept on Analysis Level.

Architectural elements in a TechnicalSafetyConcept shall be linked to elements in the corresponding FunctionalSafetyConcept with a realize relation.

7.3.7 N599_ISO_CD_26262-4_BL12_V1.doc, clause 6.4.3.2:

"If requirements with different ASILs are allocated to one architectural element this element shall be developed according to the highest ASIL."

A later draft of ISO 26262 is clearer, stating that "If an element is comprised of sub-elements with different ASILs assigned, or of non-safety-related sub-elements and safety-related sub-elements, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence, in accordance with ISO 26262-9:-, Clause 6 (Criteria for coexistence of elements), are met."

Recommendation: The ASIL assigned to a certain requirement shall propagate to the architectural elements to which this requirement applies in such a way that each element is assigned the highest ASIL of all the requirements that apply to the element. In the modeling, it shall be possible to associate ASILs with system elements and the modeler should check that the ASILs are inherited in the way defined in the standard. The ASIL inheritance rules of ISO 26262 can be represented in a checklist for the modeling. (Note that in some cases a lower ASIL can be assigned to a sub-element in accordance with the "criteria for coexistence of elements" section in Part 9 of ISO 26262).

Derived Requirements:

Each Technical Safety Requirement shall be associated to a SafetyConstraint using the Refine relation. Each SafetyConstraint shall define the ASIL level and define the exact failure mode to avoid using the FaultFailure element.

A Technical Safety Requirement derived from a Functional Safety Requirement shall have the same or higher ASIL as the Functional Safety Requirement. Alternatively, ASIL decomposition can be applied such that the Technical Safety Concept meets the Functional Safety Requirement at the required ASIL using redundancy.

7.3.8 N599_ISO_CD_26262-4_BL12_V1.doc, clause 6.4.3.x:

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

Recommendation: All interfaces of all design shall be precisely defined in the design models. This can be explicitly addressed in a design model checklist.

Derived Requirements:

Interfaces of safety-related elements shall be defined using ports and datatypes.

7.3.9 N599_ISO_CD_26262-4_BL12_V1.doc, clause 6.4.5.1:

"Measures for detection and control or control of random hardware failures shall be specified for the system design."

Recommendation: Mechanisms for error detection and error handling should be represented in the models. This can be explicitly addressed in a design model checklist.

Derived Requirements:

Functions in the FunctionalDesignArchitecture shall be allocated to Nodes in the HardwareArchitecture using the Allocation association.

Hardware-dependent error detection and control functions shall be defined as BasicSoftware-FunctionType or DesignFunctionType allocated to the concerned Node.

Checklist.

7.3.10 N599_ISO_CD_26262-4_BL12_V1.doc, clause 6.4.5.3:

"Applies to ASIL (B,) C and D: One of the alternative procedures of ISO 26262-5, Clause 8 "Assessment criteria for probability of violation of safety goals", shall be chosen and the target values for final validation at item level (see Clause 8.4.5.2) shall be specified."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include a formulation like "Have target values for the probability of safety goal violations been defined in the requirements model?"

Derived Requirements:

Each SafetyGoal shall be associated using Verify relation to a VVProcedure establishing the probability of violation of the safety goal. The VvIntendedOutcome of the VVProcedure shall define the assessment criteria for the probability of violation of the safety goal

7.3.11 N599_ISO_CD_26262-4_BL12_V1.doc, clause 10.4.2.x:

"The hardware-software interface requirements shall identify and detail each part of the HSI that is involved in a technical safety concept. It shall include hardware devices of the component that are controlled by software and hardware resources that support execution of software."

Recommendation: For hardware that is controlled by software and hardware that supports the execution of software, the hardware-software interface shall be represented in the models at design level and/or possibly at the implementation level. This can be explicitly addressed in a design model checklist.

Derived Requirements:

(Detailed HSI aspects are the concern of Implementation level)

The functionality of hardware components in a technical safety concept shall be defined using HardwareFunctionType.

(duplex) Functions in the FunctionalDesignArchitecture shall be allocated to Nodes in the HardwareArchitecture using the Allocation association.

The functional hardware-software interface shall be defined using the ports of HardwareFunctionTypes.

The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

7.3.12 N599_ISO_CD_26262-4_BL12_V1.doc, clause 10.4.4.x:

"The following characteristics shall at least be considered in the hardware/software interface specification:
a) Relevant operating modes of hardware devices (e.g. default, init, test, advanced modes) and relevant configuration parameters (e.g. gain control, band pass frequency, clock prescaler);
b) Hardware features that ensure independence between elements and support software partitioning;
c) Shared and exclusive use of hardware resources; and
d) Timing constraints defined for each service involved in the technical safety concept."

Recommendation: This ISO requirement can be represented in a checklist for the design level modeling ("Have the following characteristics been considered in the hardware/software interface specification?...").

Derived Requirements:

(Duplex) The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

7.3.13 N599_ISO_CD_26262-4_BL12_V1.doc, clause 10.4.6.x:

"The low level diagnostic capabilities of the hardware that are relevant to the technical safety concept and their use by the software shall be specified."

Recommendation: This ISO requirement can be represented in a checklist for the design level modeling ("Have any inbuilt diagnostic features within the hardware components been addressed in the design level modeling?").

Derived Requirements:

(Duplex) The non-functional hardware-software interface aspects shall be defined using requirements on the Hardware Architecture elements.

(Duplex) The functional hardware-software interface shall be defined using the ports of HardwareFunctionTypes.

7.3.14 N599_ISO_CD_26262-4_BL12_V1.doc, clause 6.4.8.2:

"System design shall be verified for compliance and completeness with regard to the technical safety concept. In this aim, the methods and measures in Table 4 shall be considered."

Recommendation: A checklist for the design level modeling should include the need to verify that design is compliant with the technical safety concept. Appropriate methods should be given in the checklist, such as inspection, walkthrough, simulation, prototyping and analysis.

Derived Requirements:

Checklist

Each technical safety requirement shall have a VVProcedure which shall be used to verify the requirement

7.4 Implementation level modeling

7.4.1 N580_ISO_26262-5_BL12.doc, clause 5.4.3:

"The hardware safety requirements specification shall include:

a) The hardware safety requirements of safety mechanisms dedicated to control internal failures of the hardware of the element, with their relevant attributes;

b) The hardware safety requirements of safety mechanisms dedicated to making the element under consideration tolerant to failures external to the element with their relevant attributes

c) The hardware safety requirements of safety mechanisms dedicated to fulfilling the safety requirements of other elements

d) The hardware safety requirements of safety mechanisms dedicated to detect and signal internal or external failures

e) The hardware safety requirements that describe the characteristics needed to ensure the effectiveness of the above safety mechanism."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like "Have hardware safety requirements related to... been specified?" (See a-e in the ISO requirement.)

Furthermore, the hardware safety requirements (as defined in ISO 26262) should be identifiable as hardware safety requirements in the requirements model and each hardware safety requirement should be assigned an ASIL in the requirement model.

Derived Requirements:

Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation

7.4.2 N580_ISO_26262-5_BL12.doc, 5.4.6:

"The criteria for qualification and testing of the hardware of the item or element shall be specified according to Clause 9, and ISO 26262-8, Clause 12. This shall include environmental conditions (temperature, vibration, EMC, etc)."

A later draft of ISO 26262 is clearer, stating the following:

"The criteria for design verification of the hardware of the item or element shall be specified, including environmental conditions (temperature, vibration, EMI, etc), specific operational environment (supply voltage, mission profile, etc) and component specific requirements:

a) for verification by qualification for hardware elements of intermediate complexity, the criteria shall meet the needs of ISO 26262-8:—, Clause 13 (Qualification of hardware components); and

b) for verification by testing, the criteria shall meet the needs of clause 10."

Recommendation: Assuming that verification of the hardware design is somehow represented by models, the acceptance criteria for such verification should be represented in these models. A checklist for such modeling can include this issue to aid the modeler.

Derived Requirements:

Each hardware safety requirement shall be linked to a VVProcedure with a Verify association.

Each hardware safety requirement shall have an acceptance criteria specified as a VvIntendedOutcome of the Requirement's VVProcedure.

7.4.3 N580_ISO_26262-5_BL12.doc, clause 5.4.13.2:

"The hardware-software interface requirements shall identify and detail each part of the HSI that is involved in a technical safety concept. It shall include hardware devices of the component that are controlled by software and hardware resources that support execution of software."

This requirement is identical to the one in N599_ISO_CD_26262-4_BL12_V1.doc , clause 10.4.2.x (also above in this guideline). A later draft of ISO 26262 is much more clear:

"The HSI specification initiated in ISO 26262-4:—, Clause 7 (System design), shall be detailed sufficiently to allow for the correct control and usage of the hardware by the software, and shall describe each safety-related dependency between hardware and software."

Recommendation: For hardware that is controlled by software and hardware that supports the execution of software, the hardware-software interface should be represented in the models at the implementation level. This can be explicitly addressed in a checklist for the implementation level.

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation

Check-list

7.4.4 N580_ISO_26262-5_BL12.doc, clause 5.4.13.5:

"Timing constraints shall be defined for each functionality involved in the technical safety concept. The HSI timing constraints shall be derived from performance specification of hardware parts and verified against the technical safety requirements."

Recommendation: When applicable, timing aspects should be accounted for in the modeling. These aspects include the timing constraints related to the performance of hardware parts. The timing constraints shall be checked for compliance with respect to the technical safety requirements. This recommendation could be implemented in a checklist to be used during implementation-level modeling.

> Note: The results of the TIMMO project (http://www.timmo.org) are expected to be relevant for this issue. However, this has not been investigated in the creation of this guidelines document.

Derived Requirements:

Safety-relevant Timing Requirements shall be defined using a Requirement with both a Timing Constraint and a SafetyConstraint associated using a Refine relation.

7.4.5 N580_ISO_26262-5_BL12.doc, clause 6.4.2.1:

"The hardware architecture shall implement the hardware safety requirements defined in Clause 5 at the required ASIL."

Recommendation: The hardware architecture model shall be consistent with the hardware safety requirements. This (obvious) requirement could be highlighted in a checklist for the implementation level modeling.

Derived Requirements:

The hardware architecture on Implementation Level shall be defined using AUTOSAR hardware elements that are linked to their corresponding elements on Design Level with a Realize association.

7.4.6 N580_ISO_26262-5_BL12.doc, clause 6.4.2.3:

"If the hardware element under consideration includes sub-elements allocated with different ASILs and/or not safety-related sub-elements, its development shall be conducted according to the highest ASIL of the sub-elements unless a criticality analysis is applied according to ISO 26262-9, Clause 5 and shows absence of interference."

Recommendation: In the hardware architecture model, ASILs should be associated to elements and sub-elements in accordance with the ISO 26262 requirements. The basic rule is that en element shall be assigned the highest ASIL of all the hardware safety requirements assigned to the element. However, if the "criteria for coexistence" in Part 9 of ISO 26262 are fulfilled, some sub-elements within the element can sometimes be assigned lower ASILs.

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation

Check-list

7.4.7 N580_ISO_26262-5_BL12.doc, clause 6.4.2.4:

"Traceability between the hardware safety requirements and their implementation shall be ensured down to hardware components."

Recommendation: The hardware architecture models should contain traceability-related information so that tracing between hardware safety requirements and corresponding architectural elements and solutions is possible. A checklist to be used in the modeling could highlight this: "Are traceability links established between requirements and implementation?"

Derived Requirements:

(duplicate) Hardware safety requirements shall be defined as a Requirement with an associated SafetyConstraint and associated to AUTOSAR hardware elements with a Satisfy relation

Check-list

7.4.8 N580_ISO_26262-5_BL12.doc, clause 8.4.2:

"Applies to ASIL (B), C and D: The item shall comply with one of the following sets of requirements:

a) Requirements 8.4.3;

b) Requirements 8.4.4. "

Recommendation: If requirements are modeled, the probability of a violation of each safety goal due to random hardware faults should be addressed in the requirements model. A choice should then be made about whether these requirements shall be in the form of required quantitative probabilities at the item level or in the form of (semi-qualitative) probabilities of each potential cause of an item-level safety goal violation.

7.4.9 N580_ISO_26262-5_BL12.doc, clause 8.4.3.2:

Applies to ASIL (B,) C and D: Target values of requirement 8.4.3.1 shall be expressed in terms of average probability per hour over the operational lifetime of the item. "

Recommendation: If target values for the probability of violation of a safety goal due to random hardware faults are specified, they should be expressed as average probability per hour over the operational lifetime of the item.

7.4.10 N580_ISO_26262-5_BL12.doc, clause 8.4.4.2:

"Applies to ASIL (B,) C and D: The failure rate class ranking for a hardware part failure rate shall be determined as follows:

a) The failure rate corresponding to Failure rate class 1 shall be less than the target for ASIL D given in 8.4.3.1 c) divided by 100;

b) The failure rate corresponding to Failure rate class 2 shall be less than ten times higher than the failure rate corresponding to Failure rate class 1;

c) The failure rate corresponding to Failure rate class 3 shall be less than a hundred times higher than the failure rate corresponding to Failure rate class 1. "

Recommendation: If violations of safety goals due to random hardware faults are addressed in the form of (semi-qualitative) probabilities of each potential cause of such violations, failure rate classes as defined in the ISO 26262 clause above should be used.

7.4.11 N585_ISO_26262-6_BL12.doc, clause 5.4.3:

"The software safety requirements specification shall be derived from the system design specification (see ISO 26262-4, 6.4.7). The software safety requirements shall be complete and consistent. Each software safety requirement inherits the ASIL of the technical safety requirement from which it is derived. The following shall be considered:

b) System and hardware configuration;

c) Hardware safety requirements, hardware-software interface and hardware architecture;

d) Timing constraints ;

e) External interfaces; and

f) Each operating mode of the vehicle, the system or the hardware having impact on the software."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project. The checklist would then include formulations like

- "Have software safety requirements been derived from the system design specification?"
- "Are the software safety requirements complete and consistent"?
- "Have the following been considered in the specification of software safety requirements?..." (see b-f in the ISO requirements above)

Furthermore, the software safety requirements should be identifiable as software safety requirements in the requirements model and each software safety requirement should be assigned an ASIL in the requirement model.

7.4.12 N585_ISO_26262-6_BL12.doc, clause 5.4.9:

"The software safety requirements shall include sufficient information to enable the following:

a) The software design and subsequent development activities can be performed effectively;

b) The software verification and the software safety acceptance testing can be performed effectively; and

c) Functional safety can be assessed effectively."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project: "Do the software safety requirements include sufficient information to enable the following?..." (see a-c in the ISO requirement above).

7.4.13 N585_ISO_26262-6_BL12.doc, clause 5.4.10:

"The software safety requirements shall address each software-based function whose failure could lead to a violation of a technical safety requirement allocated to software.."

Recommendation: This ISO requirement does not directly concern modeling but it could be addressed in a checklist for the requirements model if such a model is indeed created in a given project: "Do the software safety requirements address each software-based function whose failure could lead to a violation of a technical safety requirement allocated to software?"

7.4.14 N585_ISO_26262-6_BL12.doc, clause 5.4.11:

"The software safety requirements shall be verified according to Table 2 and Table 3 to show:

a) Compliance with the technical safety requirements and the system design specification;

b) Consistency with the hardware safety requirements specification;

c) Correct allocation of the ASIL of the system safety requirements to the software safety requirements; and

d) Completeness with regard to the technical safety requirements allocated to software."

Recommendation: A checklist for the requirements modeling should include the need for verification of the software safety requirements with respect to compliance with the functional safety concept and the system design specification, consistency with hardware safety requirements, correct allocation of ASIL, and completeness with regard to the technical safety requirements allocated to software. The checklist could explicitly mention suitable verification techniques (as given in the tables referenced by the ISO requirement above): Inspection, Walkthrough, Semi-formal verification, Formal verification.

7.4.15 N585_ISO_26262-6_BL12.doc, clause 6.4.2:

"The software architectural design shall be described according to Table 4."

Recommendation: Depending on the ASIL, the software architecture should be described using an informal or semi-formal (or formal) notation. For the lower ASILs (ASIL A and ASIL B), informal notation is considered sufficient but for the higher ASILs (ASIL C and ASIL D), at least a semi-formal notation should be used. This requirement could be highlighted in a checklist for the software architecture modeling.

7.4.16 N585_ISO_26262-6_BL12.doc, clause 6.4.9:

"Every software component shall be categorised as:

a) Newly developed;

b) Reused with modifications;

c) Reused without modifications; or

d) A COTS product."

Recommendation: For each software component, the software architecture model should include information about the component's origin: newly developed, reused with modification, reused without modification, or COTS (Commercial Off-The-Shelf). Like most of the recommendations in this document, this recommendation can be represented by an entry in a checklist for the modeling: "Has every software been categorised as...?"

7.4.17 N585_ISO_26262-6_BL12.doc, clause 6.4.12:

"The software safety requirements shall be allocated to the software components. "

Recommendation: The allocation of software safety requirements to software components shall be represented in the software architecture model and every defined software safety requirement shall be allocated to at least one software component. A checklist for the software architecture modeling could include these issues.

7.4.18 N585_ISO_26262-6_BL12.doc, clause 6.4.19:

"An upper estimation of required resources shall be made, including

a) Execution time;

b) Storage space; and

c) Communication resources. "

Recommendation: When appropriate, information about required resources (execution time, storage space, communication resources, etc) shall be represented in the software architecture model. This requirement may be highlighted in a checklist for the software architecture modeling.

7.4.19 N585_ISO_26262-6_BL12.doc, clause 6.4.20:

"The software architectural design shall be verified according to ISO 26262-8, Clause 8 and to Tables 8, 9 and 10 to show:

b) Compliance with software safety requirements;

c) Compatibility with target hardware; and

d) Adherence to design guidelines. "

Recommendation: The software architecture model should be verified with respect to compliance with software safety requirements, compatibility with target hardware and adherence to any applicable design guidelines. Possible techniques for this are walkthrough, inspection, simulation, prototype generation/animation, formal verification, control flow analysis and data flow analysis. See the corresponding Tables in ISO 26262 for which technique, or combination of techniques, to use for a particular ASIL.

This recommendation could be represented in a checklist for the software architecture modeling.

7.4.20 N585_ISO_26262-6_BL12.doc, clause 7.4.2:

"The software unit design shall be described according to Table 11. "

Recommendation: Depending on the ASIL, the software unit design should be described in natural language and also in an informal or semi-formal (or formal) notation. For ASIL A, informal notation is considered sufficient but for the higher ASILs (ASIL C and ASIL D), at least a semi-formal notation should be used. See the corresponding Table in ISO 26262 for more detailed information about which technique - or combination of techniques - to use for a particular ASIL. This requirement could be highlighted in a checklist for the software architecture modeling.

7.4.21 N585_ISO_26262-6_BL12.doc, clause 7.4.8:

"The software unit design and implementation shall be verified according to ISO 26262-8, Clause 8 and to Tables 13 and 14 to show:

a) Compliance with the requirements of 7.4.1 to 7.4.7;

b) Compliance with the hardware-software interface (see ISO 26262-5, Clause 10);

c) Completeness regarding the software safety requirements and the software architecture through traceability;

d) Consistency of the source code with the software unit specification through traceability;

e) Compliance of the source code with the coding guidelines; and

f) Compatibility of the software unit implementations with target hardware."

Recommendation: The software unit design should be verified with respect to compliance with any applicable requirements concerning the software unit design process (for example as defined in ISO 26262), compliance with the hardware/software interface, and completeness regarding both software safety requirements and the software architecture.

Possible techniques for this are inspection and walkthrough of a model of a software unit, semi-formal verification, formal verification, control flow analysis and data flow analysis. See the corresponding Tables in ISO 26262 for which technique, or combination of techniques, to use for a particular ASIL.

This recommendation could be represented in a checklist for the software unit modeling.

7.4.22 N585_ISO_26262-6_BL12.doc, clause C.4.2:

"The configuration data shall be specified to ensure the correct usage of the configurable software during the safety lifecycle. This includes:

a) Valid values of the configuration data;

b) Intent and usage of the configuration data;

c) Range, scaling, units; and

d) Interdependencies between different elements of the configuration data. "

Recommendation: Representation of configuration data in implementation models should be such that it supports the deployment of the configurable software. The following aspects should be represented within the model, when applicable:

- Valid values of the configuration data
- Intent and usage of the configuration data
- Range, scaling, units
- Interdependencies between different elements of the configuration data

This recommendation could be included in a checklist to be used in the implementation-level modeling.

7.4.23 N585_ISO_26262-6_BL12.doc, clause C.4.3:

"Verification of the configuration data shall be performed to ensure

a) Use of values within range; and

b) Compatibility with values of the other configuration data. "

Recommendation: The specific values of the configuration data for an intended use should be verified with respect to being in the valid range and being compatible with other configuration data. This recommendation could be included in a checklist to be used in the implementation-level modeling.

7.4.24 N585_ISO_26262-6_BL12.doc, clause C.4.4:

"The ASIL of the configuration data shall equal the maximum ASIL of the configurable software by which it is used. The ASIL of the configuration data may be reduced according to the results of the criticality analysis (see ISO 26262-9, Clause 5). "

This clause has been rewritten in later drafts of the ISO 26262 standard, but it is unfortunately still not particularly satisfactory. The following recommendation is based on the *assumed* intention of the clause.

Recommendation: There shall be provisions for associating an ASIL with the configuration data of any configurable piece of software. This ASIL shall equal the maximum ASIL of those safety requirements that might be violated by the configuration data if this data is incorrect. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

7.4.25 N585_ISO_26262-6_BL12.doc, clause C.4.7:

"The calibration data associated with software components shall be specified to ensure the correct operation and expected performance of the configured software. This includes:

a) Valid values of the calibration data;

b) Intent and usage of the calibration data;

c) Range, scaling and units, if applicable, with their dependence from the operating state; and

d) Known interdependencies between different calibration data of one calibration set; and

e) Known interdependencies between configuration data and calibration data."

Recommendation: Representation of calibration data in implementation models should be such that it supports the achievement of correct operation of the software. The following aspects should be represented within the model, when applicable:

- Valid values of the calibration data
- Intent and usage of the calibration data
- Range, scaling, units
- Interdependencies between different calibration data within one calibration set
- Known interdependencies between configuration data and calibration data

This recommendation could be included in a checklist to be used in the implementation-level modeling.

7.4.26 N585_ISO_26262-6_BL12.doc, clause C.4.8:

"The verification of the calibration data tests shall be planned in accordance with ISO 26262-8, Clause 8. The verification of calibration data shall examine whether the calibration data is within its specified boundaries."

This clause has been rewritten in later parts of the standard as "*The verification of the calibration data shall be planned, specified and executed in accordance with ISO 26262 8:—, Clause 9 (Verification). The verification of calibration data shall examine whether the calibration data is within its specified boundaries."*

Recommendation: The specific values of the calibration data for an intended use should be verified with respect to being in the valid range. This recommendation could be included in a checklist to be used in the implementation-level modeling.

7.4.27 N585_ISO_26262-6_BL12.doc, clause C.4.9:

"The ASIL of the calibration data shall comply with the maximum ASIL of the configurable software by which it is used. The ASIL of the calibration data can be reduced according to the results of the criticality analysis (see ISO 26262-9, Clause 5)."

This clause has been rewritten in later drafts of the standard as "*the ASIL of the calibration data shall equal the highest ASIL of the software safety requirements it can violate*" which has been taken as the basis for the following recommendation:

Recommendation: There shall be provisions for associating an ASIL with any set of calibration data. This ASIL shall equal the maximum ASIL of those safety requirements that might be violated by the calibration data if this data is incorrect. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

7.4.28 N585_ISO_26262-6_BL12.doc, clause C.4.10:

"Unintended changes of calibration data shall be detected by methods according to Table C.1. "

Recommendation: Calibration data should be checked at run-time (continuously or only during power-up) by an appropriate mechanism or set of mechanisms. Examples of mechanisms are plausibility checks, redundant storage and error detection codes. Of these three, plausibility checks should be the primary mechanism (according to ISO 26262 at least, but this could be debated.) This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

7.4.29 N585_ISO_26262-6_BL12.doc, clause D.3.5:

"That part of the software that implements software partitioning shall have the same or higher ASIL than the highest ASIL associated with the software partitions. "

Recommendation: An ASIL shall be associated with that part of the software that implements software partitioning. This ASIL shall equal the highest ASIL among the software partitions that are protected by this partitioning software. This recommendation can be highlighted by including it in a checklist for the implementation level modeling.

7.5 Orthogonal issues, applicable to all modeling levels

7.5.1 N578_BL12_CD_26262-2_BL12_2009_Jan_15.doc, clause 5.4.5.4:

"*The results of the confirmation measures shall be added to the safety case*"

Recommendation: A safety case is an argumentation of why a system is adequately safe. If this safety case is represented in a model, for example a GSN (Goal Structuring Notation) model, it should be ensured that the confirmation measures are included in the model. This can be addressed in a checklist for safety case modeling, with the ISO 26262 requirement as stated above included in the checklist. (The confirmation measures are the audits, reviews and functional safety assessments described in Part 2 of ISO 26262.)

7.5.2 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.2:

"The safety requirements shall be specified according to Table 1 "

Recommendation: Safety requirements should be specified using natural language and an appropriate combination of informal and semi-formal (or even fully formal) notations. Informal notation is considered sufficient, together with natural language, for ASILs A-B. For higher ASILs, a combination of natural language and semi-formal notation is considered sufficient. This recommendation can be highlighted by including it in a checklist for the requirements modeling.

7.5.3 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.3.1:

"Safety requirements shall be unambiguously identifiable as safety requirements."

Recommendation: Safety requirements should be clearly identifiable as being safety requirements. Thus, in a requirements model, the safety requirements should have some specific "tag" or other special characteristic that differentiates them from other requirements. To aid the modeler, a checklist for the requirements modeling could include the following entry: "Have all safety requirements been labeled as safety-critical in the model?"

7.5.4 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.3.2:

"Safety requirements shall have the following characteristics:

a) Unambiguous and comprehensible;

b) Atomic;

c) Internally Consistent;

d) Feasible; and

e) Verifiable."

Recommendation: Every safety requirement should be unambiguous, comprehensible, atomic, internally consistent (i.e. the requirement should not contradict itself), feasible and verifiable. These characteristics of the safety requirements can be listed in a checklist.

7.5.5 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.3.3:

"Safety requirements shall have the following attributes:

a) Unique identification remaining constant through the existence of the requirement;

b) Status; and

c) ASIL."

Recommendation: Every safety requirement should have a unique and constant identification, a status and an ASIL. If requirements models are used, these characteristics of the safety requirements should be possible to represent in the models. Furthermore, the characteristics can be listed in a checklist.

7.5.6 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.4.1:

"The following shall be ensured for the whole of the safety requirements:

a) Hierarchical structure;

b) Organisation of safety requirements;

c) Completeness;

d) External consistency;

e) No duplication of information within any level of the hierarchical structure; and

f) Maintainability."

Recommendation: The complete set of safety requirements should be hierarchical, organized, complete and consistent (i.e. requirements should not contradict each other). These characteristics of the safety requirements can be listed in a checklist for the safety requirements.

7.5.7 N468_ISO_CD_26262-8-5_Overall_Management_of_Safety_Requirements.doc, clause 5.4.4.2:

"Safety requirements shall be traceable where references shall be made to

a) All sources of a safety requirement at the upper hierarchical level;

b) All derived safety requirements at a lower hierarchical level, or direct implementation in the system; as well as;

c) The verification procedures."

Recommendation: Every safety requirement should be associated with traceability information concerning the sources at the next higher hierarchical level from which the requirement has been derived. For example, a technical safety requirement shall be linked to the functional safety requirements from which it has been derived. Similarly, links to the next lower hierarchical level (derived safety requirements or implementation) shall be established. Furthermore, traceability links shall be established to the verification procedures where the fulfillment of the requirement is checked.

This need for traceability from any requirement to related higher and lower requirements and to the verification procedures can be addressed in a checklist for the requirements modeling.

7.5.8 N589_ISO_CD_26262-9_for_BL12__2009-02-04_.doc, clause 4.4.4:

"If ASIL decomposition between the intended functionality and its associated safety mechanism is applied the following shall be complied with:

a) The intended functionality shall become a safety requirement; and

b) The intended functionality shall be implemented at the derived ASIL."

The formulation is improved in later drafts of the ISO 26262 standard and the following recommendation is based on this improved formulation.

Recommendation: ASIL decompositioning can made by decomposing a safety requirement into two equivalent safety requirements, one of which is allocated to an intended functionality (i.e. a nominal function of the considered system) and the other is allocated to an associated safety mechanism. The idea is then that the nominal function, which is typically quite complex, can be assigned a relatively low ASIL while the safety mechanism, which is typically relatively simple, can be assigned a relatively high ASIL. Thus, the need for extremely stringent development of the (complex) nominal functionality is alleviated.

ASIL decompositioning rules in general, and this recommendation in particular, can be addressed in a checklist for the system development.

7.5.9 N589_ISO_CD_26262-9_for_BL12__2009-02-04_.doc, clause 4.4.5:

"The following rules shall be applied to each safety requirement:

a) One of the decomposition schemes given in 4.4.6 shall be selected in accordance with the initial ASIL;

b) Each step from one level of the selected decomposition scheme to the lower next level defines one decomposition of the ASIL

c) Decompositions resulting in lower ASILs than those given in 4.4.6 shall not be applied; while decompositions resulting in higher ASILs may be applied;

d) ASIL decomposition may be applied more than once as long as the decomposition schemes given in 4.4.6 or higher decompositions are used;

e) The decomposed ASILs shall be marked by giving the initial ASIL before any decomposition in parenthesis."

Recommendation: ASIL decompositioning may be applied more than once, for example an ASIL D requirement may be decomposed into one ASIL C(D) requirement and one ASIL A(D) requirement. The ASIL C(D) requirement may be further decomposed into an ASIL B(D) requirement and an ASIL A(D) requirement.

7.5.10 N589_ISO_CD_26262-9_for_BL12__2009-02-04_.doc, clause 4.4.6:

"According to the initial ASIL one of the following decomposition schemes (see Figure 3) shall be chosen:

a) ASIL D shall be decomposed either as

1) One ASIL C(D), one ASIL A(D) and methods and processes according to 4.4.7; or as

2) One ASIL B(D), one ASIL B(D) and methods and processes according to 4.4.8; or as

3) One ASIL D(D), one QM(D) and methods and processes according to 4.4.7.

b) ASIL C shall be decomposed either as

1) One ASIL B(C), one ASIL A(C) and methods and processes according to 4.4.7; or as

2) One ASIL C(C), one QM(C) and methods and processes according to 4.4.7.

c) ASIL B shall be decomposed either as

1) One ASIL A(B), one ASIL A(B) and methods and processes according to 4.4.7; or as

2) One ASIL B(B), one QM(B) and methods and processes according to 4.4.7.

d) ASIL A shall not be further decomposed, except, if needed, as one ASIL A(A), one QM(A) and methods and processes according to 4.4.7."

Recommendation: If ASIL decompositioning is performed, it shall be performed in accordance with the prescribed decomposition schemes in part 9 of ISO 26262.