DRAFT INTERNATIONAL STANDARD ISO/DIS 26262-10

ISO/TC **22**/SC **3** Secretariat: **DIN**

Voting begins on: **2009-07-08** Voting terminates on: **2009-12-08**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Road vehicles — Functional safety —

Part 10: Guideline

Véhicules routiers — Sécurité fonctionnelle —

Partie 10: Lignes directrices

ICS 43.040.10

In accordance with the provisions of Council Resolution 15/1993 this document is circulated in the English language only.

Conformément aux dispositions de la Résolution du Conseil 15/1993, ce document est distribué en version anglaise seulement.

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 26262-10 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electric and electronic equipment*.

ISO 26262 consists of the following parts, under the general title *Road vehicles — Functional safety*:

— *Part 1: Vocabulary*

— *Part 2: Management of functional safety*

— *Part 3: Concept phase*

— *Part 4: Product development: system level*

— *Part 5: Product development: hardware level*

— *Part 6: Product development: software level*

— *Part 7: Production and operation*

— *Part 8: Supporting processes*

— *Part 9: ASIL-oriented and safety-oriented analyses*

— *Part 10: Guideline on ISO 26262*

Introduction

ISO 26262 is the adaptation of IEC 61508 to comply with needs specific to the application sector of E/E systems within road vehicles.

This adaptation applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic, and software elements that provide safety-related functions.

Safety is one of the key issues of future automobile development. New functionality not only in the area of driver assistance but also in vehicle dynamics control and active and passive safety systems increasingly touches the domain of safety engineering. Future development and integration of these functionalities will even strengthen the need of safe system development processes and the possibility to provide evidence that all reasonable safety objectives are satisfied.

With the trend of increasing complexity, software content and mechatronic implementation, there are increasing risks from systematic failures and random hardware failures. ISO 26262 includes guidance to avoid these risks by providing feasible requirements and processes.

System safety is achieved through a number of safety measures, which are implemented in a variety of technologies (for example: mechanical, hydraulic, pneumatic, electrical, electronic, programmable electronic etc). Although ISO 26262 is concerned with E/E systems, it provides a framework within which safety-related systems based on other technologies can be considered.

ISO 26262:

— provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases;

— provides an automotive specific risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs);

— uses ASILs for specifying the item's necessary safety requirements for achieving an acceptable residual risk; and

— provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety being achieved.

Functional safety is influenced by the development process (including such activities as requirements specification, design, implementation, integration, verification, validation and configuration), the production and service processes and by the management processes.

Safety issues are intertwined with common function-oriented and quality-oriented development activities and work products. ISO 26262 addresses the safety-related aspects of the development activities and work products.

Figure 1 shows the overall structure of ISO 26262. ISO 26262 is based upon a V-Model as a reference process model for the different phases of product development. The shaded "V"s represents the relations between ISO 26262-3, ISO 26262-4, ISO 26262-5, ISO 26262-6 and ISO 26262-7.

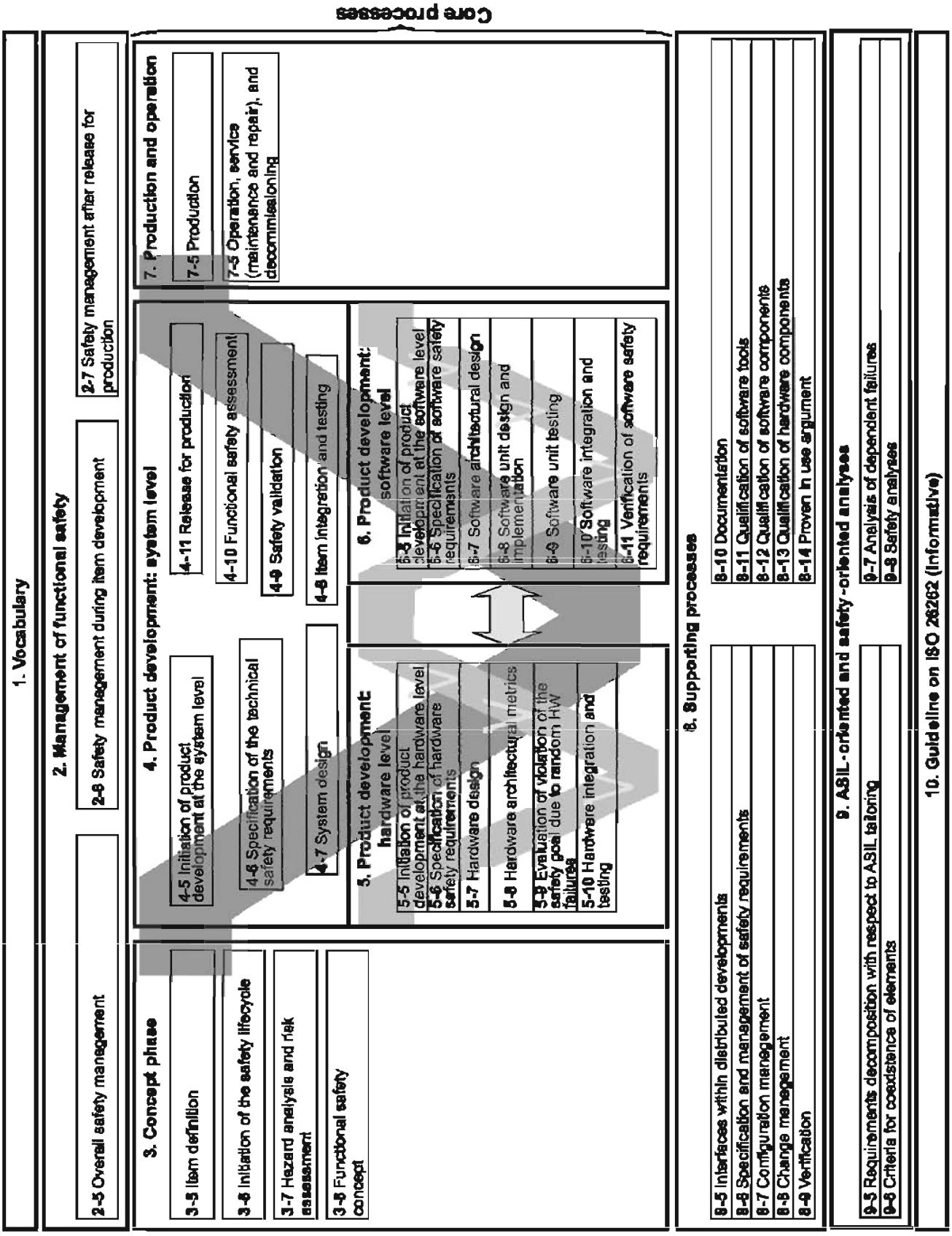

Figure 1 — Overview about ISO 26262

Road vehicles — Functional safety — Part 10: Guideline

1 Scope

ISO 26262 is intended to be applied to safety-related systems that include one or more E/E systems and that are installed in series production passenger cars with a max gross weight up to 3.5 t. ISO 26262 does not address unique E/E systems in special purpose vehicles such as vehicles designed for drivers with disabilities. Systems developed prior to the publication date of ISO 26262 are exempted from the scope.

ISO 26262 addresses possible hazards caused by malfunctioning behaviour of E/E safety-related systems including interaction of these systems. It does not address hazards as electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy, and similar hazards unless directly caused by malfunctioning behaviour of E/E safety-related systems.

ISO 26262 does not address the nominal performance of E/E systems, even if dedicated functional performance standards exist for these systems (for example active and passive safety systems, brake systems, ACC).

This Part of ISO°26262 has an informative character only.

This Part provides an informative overview of ISO°26262, as well as additional explanations, intended to enhance the understanding of the other Parts of ISO°26262. It describes the general concepts of ISO°26262 in order to make it easier to understand. The explanation expands from general concepts to specific contents.

In the case of inconsistencies between this Part, and the other dedicated Parts of ISO°26262, the requirements, recommendations and information given in the dedicated part of ISO°26262 apply.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 26262-1: —[1] *Road vehicles – Functional Safety — Part 1: Vocabulary*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 26262-1 apply.

[1] To be published

4 Key concepts of ISO°26262

4.1 Functional safety for automotive systems (relationship with IEC61508)

IEC 61508 (Functional safety of electrical, electronic and programmable electronic (E/E/PE) safety-related systems) is designated by IEC as a generic standard and a basic safety publication. This means that industry sectors will base their own standards for functional safety on the requirements of IEC 61508.

In the automotive industry, there are a number of issues with applying IEC 61508 directly. A non-exhaustive list of these issues and the corresponding differences in ISO°26262 are as follows:

IEC 61508 is based upon the model of "equipment under control", for example the industrial plant that has an associated control system. A hazard analysis identifies the hazards associated with the equipment under control, to which risk reduction measures will be applied. This can be achieved through E/E/PE systems, or other technology safety-related systems (e.g. a safety valve), or external risk reduction measures (e.g. a physical containment of the plant). Risk reduction allocated to E/E/PE systems is achieved through safety functions, which are designated as such. These safety functions are either part of a separate protection system, or can be incorporated into the plant control. In contrast, it is rarely possible to make this distinction in automotive systems. The safety of a vehicle depends on the correct operation of the control systems themselves.

Therefore, instead of the model of separate safety functions, ISO°26262 uses the concept of safety goals and the safety concept as follows:

— A hazard analysis and risk assessment identifies hazards that need risk reduction.

— A safety goal is formulated for each hazardous event.

— An Automotive Safety Integrity Level (ASIL) is associated with each safety goal.

— The functional safety concept is a statement of the functionality to achieve the safety goal. This is stated in the functional safety requirements.

— The technical safety concept is a statement of how this functionality is implemented in hardware or software. This is stated in the technical safety requirements.

— Software safety requirements and hardware safety requirements state the specific safety requirements which will be implemented as part of the software and hardware design.

Example:

— The airbag system: one of the hazards is unintended deployment

— The associated safety goal is to ensure that the airbag does not deploy, unless a crash occurs that requires the deployment.

— The functional safety concept specifies to provide a redundant function to detect whether the vehicle is in a collision.

— The technical safety concept specifies the implementation of two independent accelerometers with different axial orientations and two independent firing circuits. The squib deploys if both are closed.

IEC 61508 is aimed at one-off or low volume systems. Generally the system is built and tested, then installed on the plant, and then safety validation is performed. For mass-market systems such as road vehicles, safety validation is performed before the release for volume (series) production. Therefore the order of lifecycle activities in ISO°26262 is different. Related to this, ISO°26262-7 addresses requirements for production. These are not covered at all in IEC 61508.

IEC 61508 has an implicit assumption that the system will be designed and implemented by one organization. Automotive systems are generally produced by one or more suppliers of the customer, e.g. the vehicle manufacturer. ISO°26262 includes specific requirements for managing development across multiple organizations, including the Development Interface Agreement (DIA, see ISO°26262-8, Clause°5).

IEC 61508 does not contain normative requirements for hazard classification. ISO°26262 contains an automotive scheme for hazard classification. This scheme recognizes that a hazard in an automotive system does not necessarily lead to an accident. The outcome will depend on whether the persons at risk are actually exposed to the hazard in the driving situation in which it occurs; and whether they are able to take steps to control the outcome of the hazard. This concept is reflected in Figure 2.

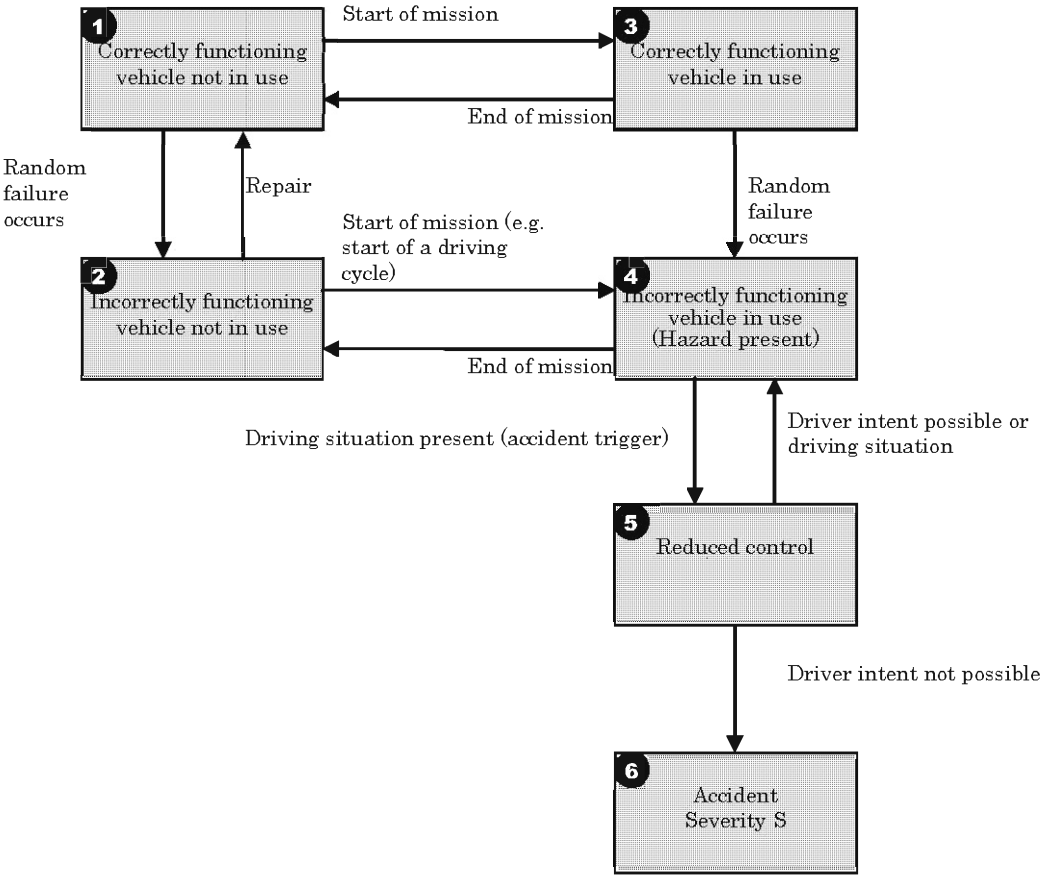

Figure 2 — State machine model of automotive risk

The requirements for hardware development (ISO°26262-5) and software development (ISO°26262-6) are adapted for the state-of-the-art in the automotive industry. Specifically, ISO°26262-6 contains requirements concerned with model-based development, which is not recognized at all in IEC 61508.

Furthermore, the requirements for techniques and measures in IEC 61508 are prescriptive and detailed rationales for the use of alternative measures are provided. The measures specified in IEC 61508 are not commonly used in the automotive industry. ISO°26262 recommends methods and measures based on automotive practices. Where possible, these methods and measures have been stated as a goal rather than a specific practice.

Risk reduction requirements in ISO°26262 are stated as an ASIL (Automotive Safety Integrity Level) rather than a SIL (Safety Integrity Level). The main motivation for this is that the SIL in IEC 61508 is stated in probabilistic terms (see IEC 61508-1, Table 3). Although IEC 61508 states that these are targets, and that they can only be quantified for random failures of hardware, in practice these headline figures are often used as the statement of risk reduction requirements. An ASIL does not contain this probabilistic requirement/target.

4.2 Item, system, element, component, hardware part, and software unit

The terms item, system, component, hardware part, software unit, and element are respectively defined in ISO°26262-1. As shown in Figure 3 and Figure 4, an item refers to the entire scope under consideration and is an array of systems, a system, or one or more functions. For the case of one or more functions, these functions are correspondingly implemented by a system or array of systems. A system is a set of elements, including at a minimum a sensor, a controller, and an actuator (see Figure 4). An element is any sub-unit of an item, and can or cannot be dissoluble. An irresolvable element is a hardware part or a software unit. A dissoluble element can, be labeled as a system, a subsystem, or a component. A dissoluable element that meets the criteria of a system can be labeled as a system or subsystem. The term subsystem would typically be used when it is important to emphasize that the element is part of a larger system. A component is a non-system level, non-elementary, logically and technically separable element. Often the term component is applied to an element that is only comprised of parts and units, but can also be applied to an element comprised of lower-level elements from a specific technology area e.g., electrical / electronic technology (see Figure 4).

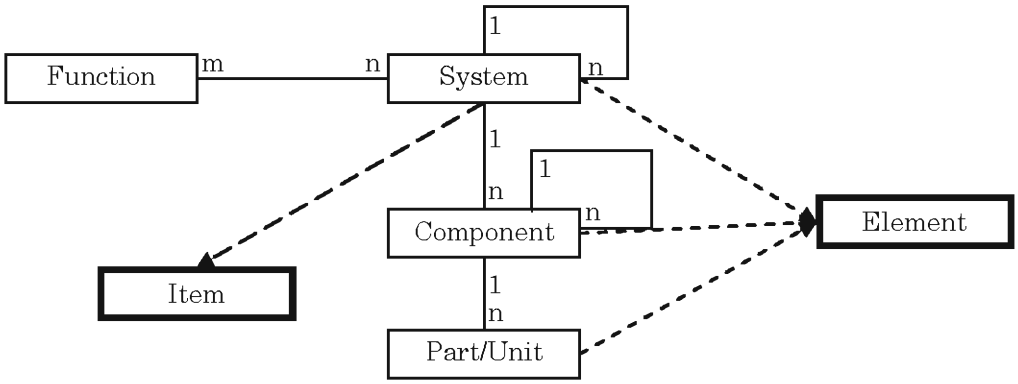

Figure 3 — Relationship of item, system component, hardware part, software unit, and element

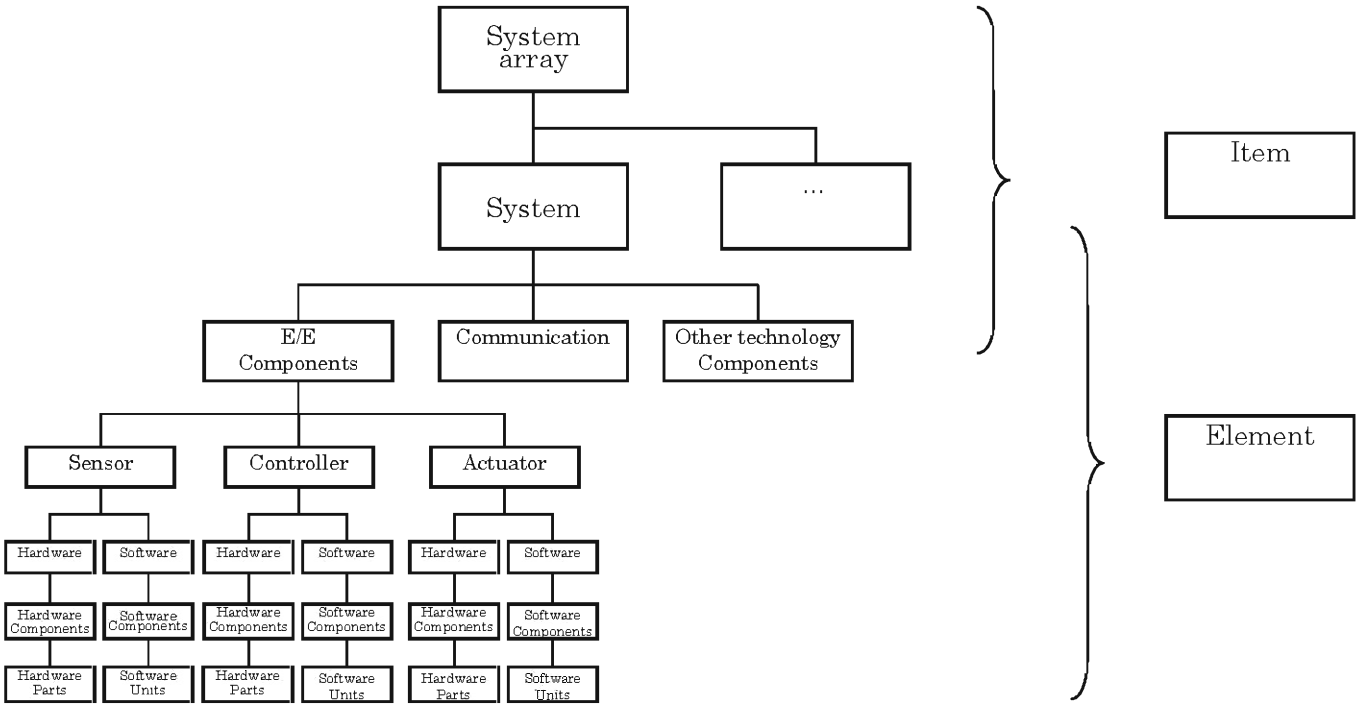

Figure 4 — Example item dissolution

5 Explanation of fault, error and failure

5.1 Relationship among fault, error and failure

The terms fault, error, and failure are respectively defined in ISO°26262-1. Figure 5 depicts the progression of faults to errors to failures from three different types of causes: systematic software issues, random hardware issues and systematic hardware issues. Systematic faults (see ISO°26262-1) are typically due to design or specifications issues; all software faults and a subset of hardware faults are systematic. Random hardware faults (see ISO°26262-1) are typically due to physical processes such as damage. At the component level, each different type of fault can lead to different failures. However, failures at the component level of abstraction are faults at the item level of abstraction, a vehicle in this case. Note that in this example, at the vehicle level, faults from different causes can lead to the same failure. A subset of all failures will be hazards (see ISO°26262-1) if additional environmental factors permit the failure to contribute to an accident scenario. For instance, if the bucking behaviour of the vehicle occurs while the vehicle is starting to cross an intersection, a crash might occur.

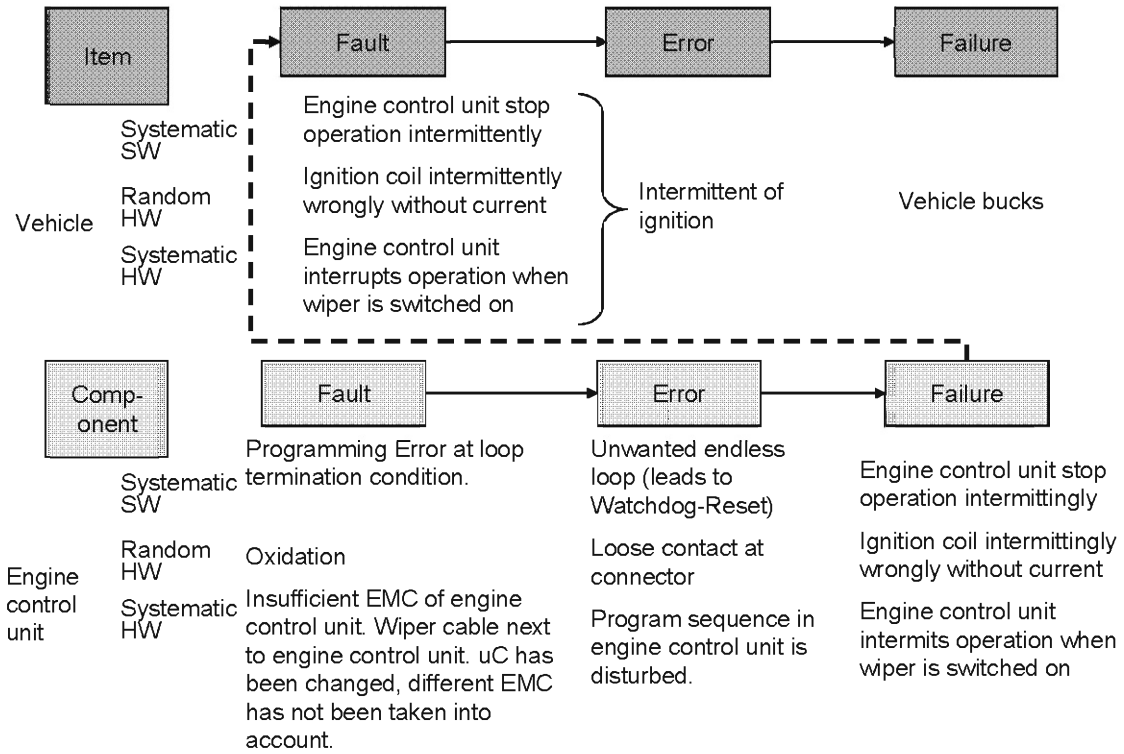

Figure 5 — Example of fault leading failures

5.2 The fault classes (applies only to random hardware faults)

In general, the combinations of faults which will be considered are limited to combinations of two faults, unless it is shown in the functional or technical safety concept that n point faults with n > 2 are relevant. Therefore in most cases a fault will be classified either a:

— Single Point Fault,

— Residual Fault,

— Detected dual point fault,

— Perceived dual point fault,

— Latent dual point fault or,

— Safe fault.

Explanations on the various fault classes, as well as examples, are given below.

— Single Point Fault:

This fault leads directly to the violation of the safety goal.

— No safety mechanism is implemented to control any fault of the HW element that has the potential to violate the safety goal.

EXAMPLE An unsupervised resistor for which one failure mode has the potential to violate the safety goal.

— Residual Fault:

— This fault leads directly to the violation of the safety goal.

— At least one safety mechanism is implemented to control some faults of this HW element that has the potential to violate the safety goal.

EXAMPLE If a Random Access Memory (RAM) module is only checked by a checkerboard RAM test, certain kinds of bridging faults will not be controlled. These faults are examples of residual faults.

— Detected dual point fault:

— This fault contributes to the violation of the safety goal but will only lead to the violation of the safety goal in combination with one other independent fault.

— This fault is detected by a safety mechanism to prevent the fault from being latent within a prescribed time.

EXAMPLE 1 In the case of a flash which is protected by parity: a single bit fault which is detected and controlled

EXAMPLE 2 In the case of a flash which is protected by an Error Correction and Detection logic (EDC): faults in the EDC logic that are detected by a test

— Perceived dual point fault:

— This fault contributes to the violation of the safety goal but will only lead to the violation of the safety goal in combination with one other independent fault.

— This fault is deduced by the driver without detection by a safety mechanism within a prescribed time.

EXAMPLE A dual point fault can be perceived by the driver if the functionality is significantly and unambiguously affected by the consequence of the fault.

— Latent dual point fault

— This fault contributes to the violation of the safety goal but will only lead to the violation of the safety goal in combination with one other independent fault.

— This fault is neither detected by a safety mechanism nor perceived by the driver. In other words, this particular fault is tolerated since system is still operable even if the driver is not informed about the fault.

EXAMPLE 1 In the case of a flash which is protected by Error Detection and Correction logic (EDC): a single bit fault that is corrected by the EDC but is not signaled. In this case the fault is controlled (since the faulty bit is corrected) but it is

neither detected (since the single bit fault is not signaled) nor perceived (since there is no impact on the functionality of the application). If an additional fault occurs in the EDC logic it can lead to a loss of control of this single bit fault, leading to a potential violation of the safety goal.

EXAMPLE 2 In the case of a flash which is protected by EDC: a fault in the EDC logic leading to an unavailability of the EDC which is not detected by a test.

— Safe fault:

  — n point fault with n > 2 can be considered a safe fault unless shown relevant in the safety concept.

  — A safe fault is also a fault that will not contribute to the violation of a safety goal.

  — In both cases the probability of violation of the safety goal is not significantly increased with a safe fault.

EXAMPLE 1 In the case of a transient fault, for which a safety mechanism restores the item to a fault free state, such a fault can be considered as a safe fault even if the driver is never informed about its existence.

EXAMPLE 2 In the case of a flash that is protected by EDC and a Cyclic Redundancy Check (CRC): a single bit fault which is corrected by EDC but is not signaled:

  — The fault is controlled but not signaled by the EDC.

  — If the EDC logic fails the fault is detected by the CRC, leading to a switch off of the system.

  — Only if a single bit fault in the flash is present, the EDC logic fails and the CRC checksum supervision fails, a violation of a safety goal can occur (n=3).

Failure modes of a hardware element can be classified as shown in figure 6 and using the flow diagram described in figure 7:

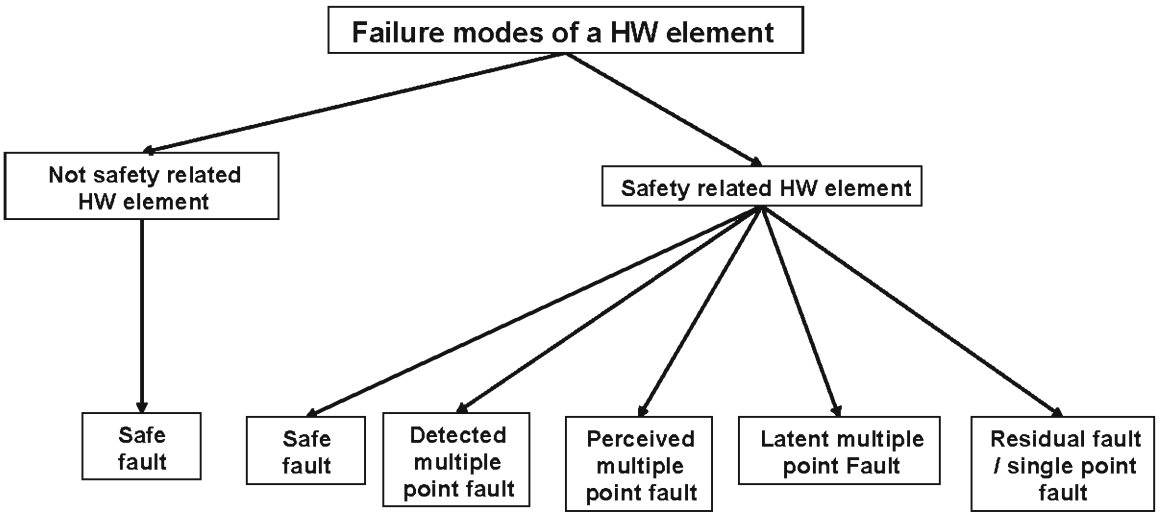

Figure 6 — Failure modes classification of a hardware element

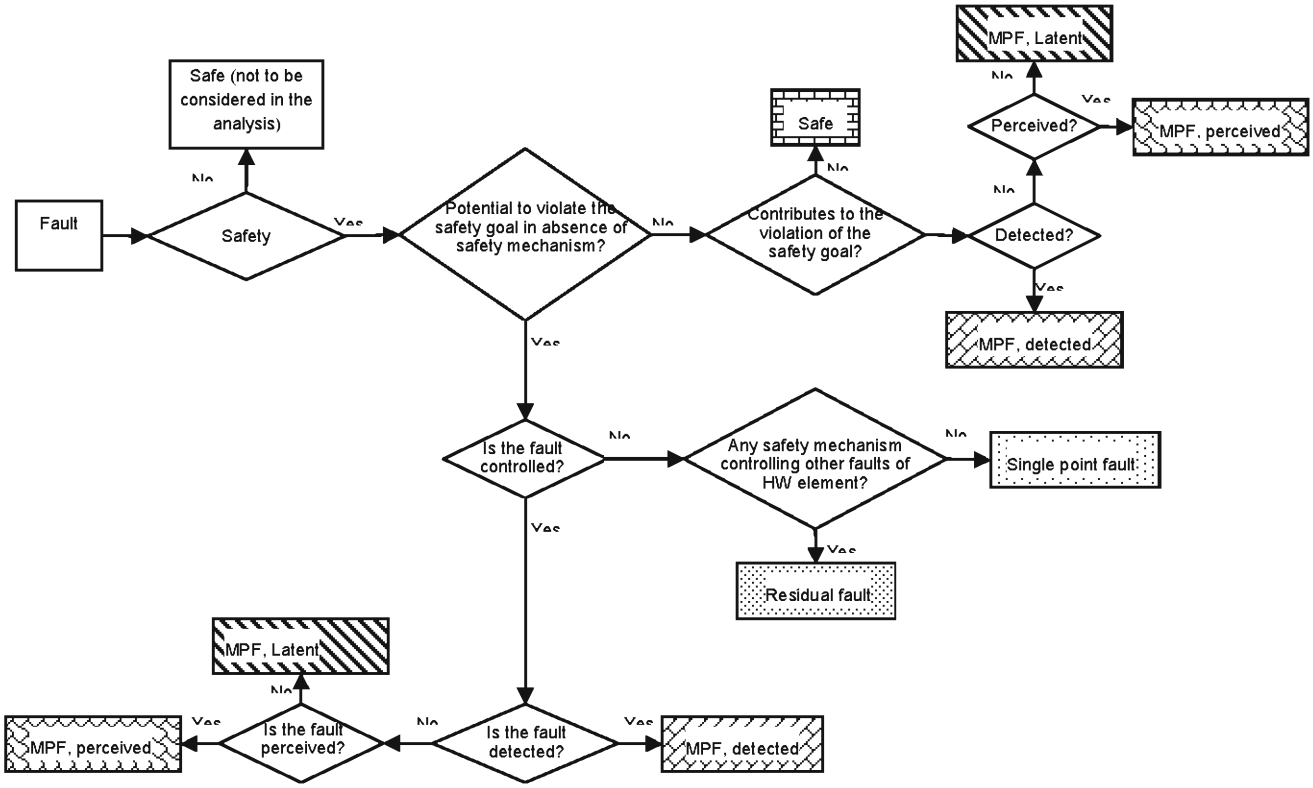

Figure 7 — Example of flow diagram for fault classification

NOTE Multiple point faults of a distance strictly higher than n=2 can be considered as safe faults unless shown relevant in the functional or technical safety concept.

6 Notions of controllability

Controllability is defined as, the avoidance of a specified harm or damage through timely reactions of the persons involved, in ISO°26262-1. In ISO°26262-3, Clause°7 it is further explained that the controllability is an estimation of the probability that the driver, or the other endangered persons, is able to gain control of the hazardous event that is arising and is able to avoid harm.

As described in ISO°26262-3, Clause°7, there are four levels of controllability, listed below with informative definitions from ISO°26262-3, Annex B in parentheses.

— C0: Controllable in general

— C1: Simply controllable (99% or more of all drivers or other traffic participants are usually able to avoid a specified harm)

— C2: Normally controllable (90% or more of all drivers or other traffic participants are usually able to avoid a specified harm)

— C3: Difficult to control or uncontrollable (Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid a specified harm)

— A hazardous event is defined in ISO°26262-1 as a combination of a hazard and an operational situation. Each relevant hazardous event is investigated in the hazard analysis and risk assessment.

— In the simplest case, only one outcome is considered for a given hazardous event and the controllability is a measure of the likelihood that this outcome is avoided. However, there might be other cases. For example, a severe outcome (e.g. severity class S2) might be possible but relatively easy to avoid (e.g. controllability C1) while a less severe outcome (e.g. S1) might be more difficult to avoid (e.g. C3). Assuming that the Exposure class is E4, the following set of values might be the result, which illustrates that it might not be the highest severity that is the most relevant:

  — E4, S2, C1 => ASIL A

  — E4, S1, C3 => ASIL B

  — In this example, ASIL B might be an appropriate classification of the hazard.

— The analysis of a given deviation from the nominal functionality of a system might consider several different operational situations, each having different properties.

7 Safety process requirement structure - Flow and sequence of the safety requirements

The flow and sequence of the safety requirement development per ISO°26262 is illustrated in figure 8 and outlined below.

A hazard analysis and risk assessment is performed to identify the risks and to define the safety goals for these possible risks. (see ISO°26262-3, Clause 7).

A functional safety concept is derived which specifies requirements to satisfy the safety goals. These requirements define the safety mechanisms and the other safety measures that will be used for the item. In addition the elements of the system architecture are identified which will support these requirements. (see ISO°26262-3, Clause 8).

A technical safety concept is derived which specifies how functional safety requirements will be implemented. These technical safety requirements will indicate the partitioning of the elements between the hardware and the software. (see ISO°26262-4, Clause 6).

The system design will be developed in accordance with the technical safety requirements. Their implementation can be specified in the system design specification (see ISO°26262-4, Clause 7).

Finally, the hardware and software safety requirements will be provided to comply with the technical safety requirements and the system design (see ISO°26262-5 Clause 6).

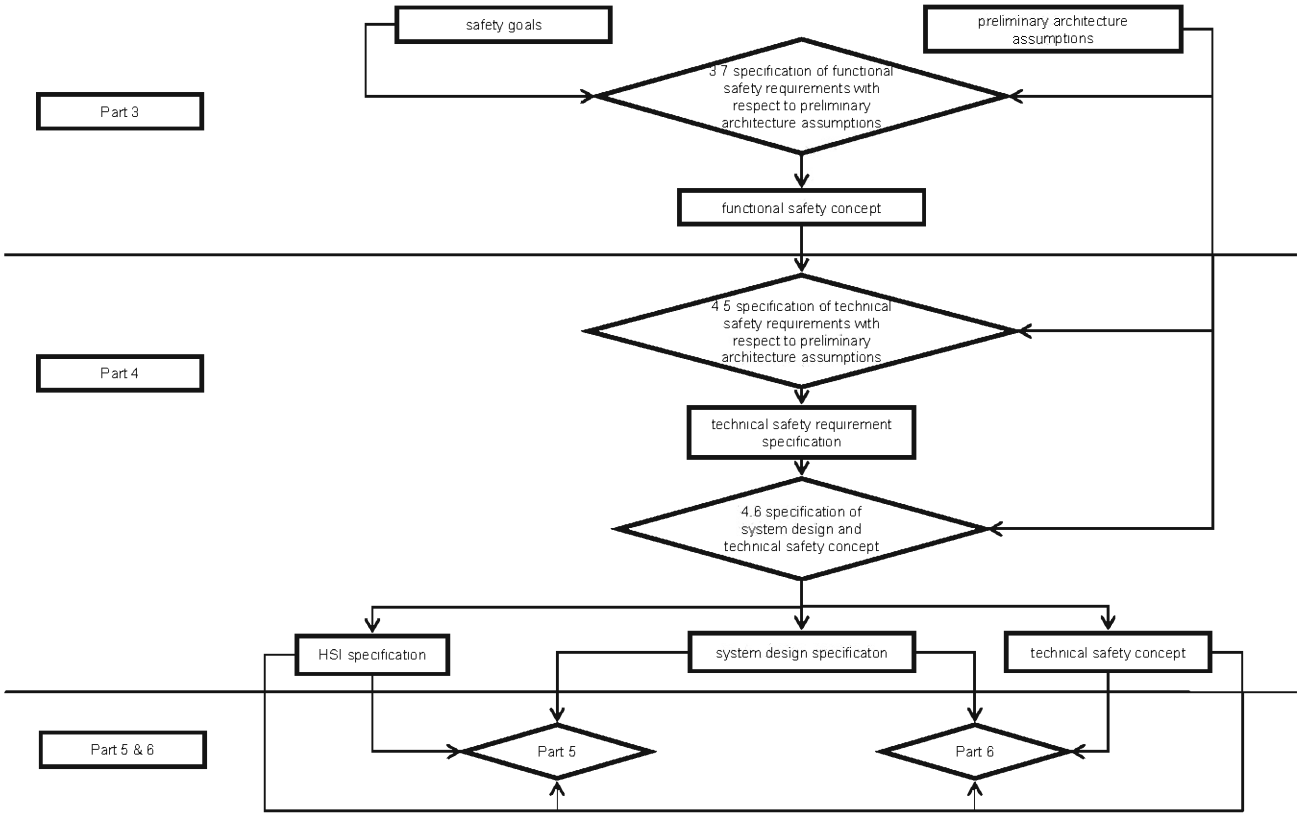

Figure 8 — Flow of safety requirements

Figure 9 illustrates the relationship between the hardware requirements and the design phases per ISO°26262.

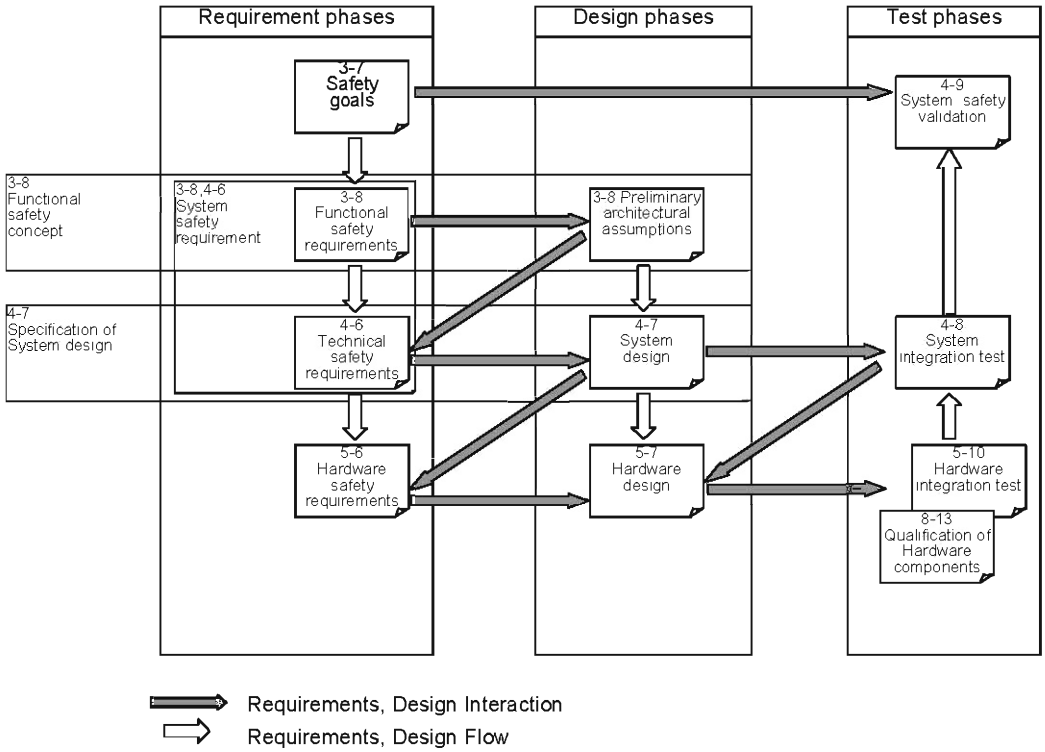

Figure 9 — Hardware safety requirements process

Figure 10 illustrates the relationship between the software requirements, the design, and the test subphases per ISO°26262.

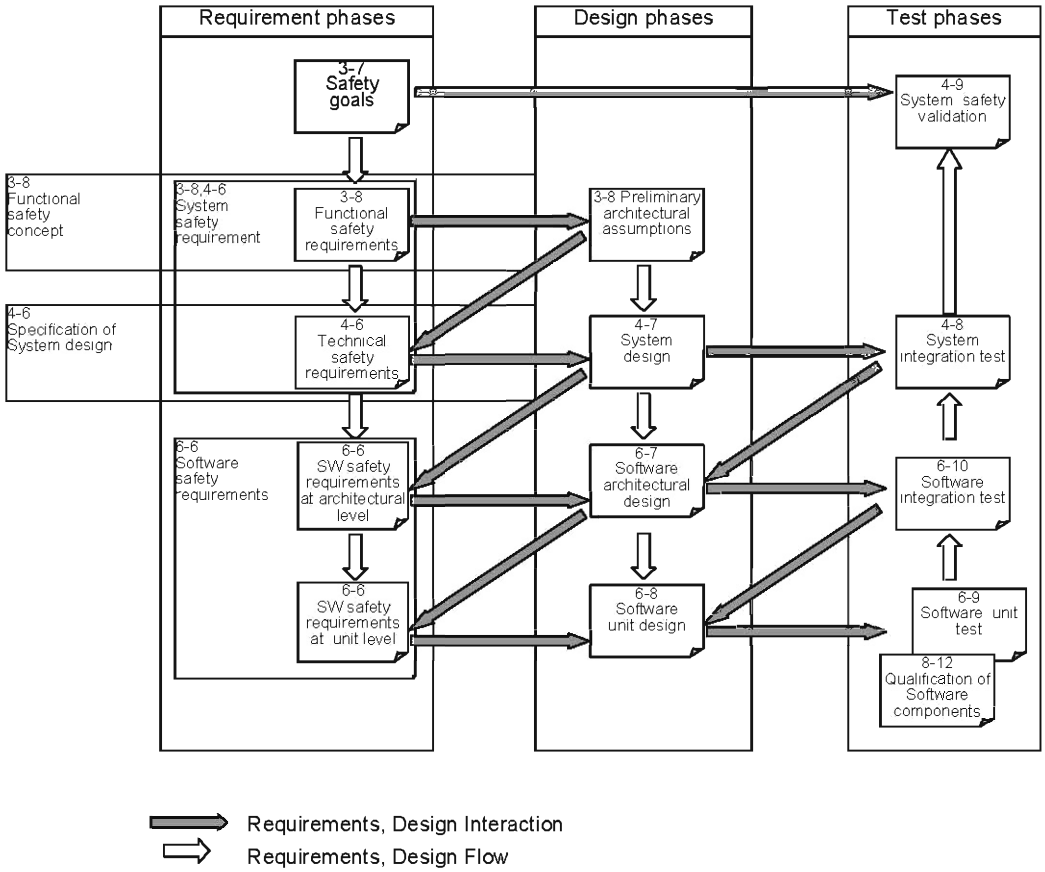

Figure 10 — Software safety requirements process

8 Work Products, confirmation measures, qualification and authority

8.1 Work products and confirmation measures

8.1.1 General

This clause describes the terms: work product and confirmation measures, including an explanation of the level of independence of the reviewers, auditors, or assessors.

8.1.2 Work products

In ISO°26262, a work product is information or data which constitutes the result of one or more system safety process activities. Work products will be in a format appropriate to the work product's content. For example, an executable model can be represented by one or more electronic files that are read using a simulation development tool, while a specification can be captured within a requirements database or a text file.

A work product in accordance with ISO°26262 might not be a separate document. The information can be included in already existing documentation, or several work products can be included in one document.

Many of the work products produced by process activities in ISO°26262 are evaluated within subsequent activities. These evaluations can be part of the confirmation measures or part of the product verification process activities.

The confirmation measures are used to ensure the proper execution of the system safety process, with sufficient completion of the safety lifecycle steps and work products. In addition, these measures provide for the evaluation of the system safety activities and work products as a whole, to enable determination of the adequacy of achievement of the functional safety goals. These confirmation measures include functional safety audits, confirmation reviews and functional safety assessments. These are detailed in ISO°26262-2, clause°6, and summarized in the following clause.

By contrast, the verification activities are intended to ensure that a given product development activity fulfils the project's technical requirements. For example, verification is performed to verify that derived requirements, as captured in work products, are technically correct, consistent and complete. Similarly, verification testing ensures the fulfilment of the specified requirements, verifying that the item or its elements comply with the requirements and achieve the intended function. The verification of the work products can therefore consist of technical reviews of the work products, execution of test plans, evaluation of test data, etc. The verification activities are given in dedicated clauses within ISO°26262-4, ISO°26262-5, ISO°26262-6 and ISO°26262-8, clause°5. In addition, ISO°26262-8, clause°9 provides information generic to all the verification activities noted within ISO°26262.

8.1.3 Confirmation Measures

8.1.3.1 General

Three types of confirmation measures are defined within ISO°26262 to corroborate the achievement of functional safety of the item. These are:

— The functional safety audit, which confirms the correct execution of the functional safety processes.

— The functional safety assessment, which confirms the steps taken to design, develop and execute a product design to ensure that the item achieves functional safety.

— Confirmation reviews, which confirm that key work products in the development cycle have achieved the goals for these work products (content, coverage, accuracy, completeness, correctness, etc.).

8.1.3.2 Level of independence for performing the confirmation measures

Each of the confirmation measures will call for the participation from experienced individuals to conduct the functional safety audit, the functional safety assessment and the confirmation reviews. In order to ensure that these evaluations are conducted in an objective manner, guidelines have been established for the level of independence of these participating reviewers. Four levels are provided for in ISO°26262. These are detailed in ISO°26262-2, clause°6.

Also, each confirmation measure can have additional criteria for the level of independence of the reviewers, auditors, or assessors. These are detailed in ISO°26262-2, Table 1.

The confirmation measures and the associated reviewer independence requirements are applied within the system safety process of an item in accordance with the highest ASIL level in the safety goals of the item under review. For example, in referencing ISO°26262-2, Table 1, a functional safety audit of an item with ASIL B and ASIL C safety goals will require participation from an individual with I2 independence from the engineering team executing the system safety process for that item (corresponding to the highest ASIL level, namely ASIL C).

8.2 Qualification and authority

Examples of the qualifications to successfully perform the activities in the first column of Table 1 are described and classified in the three columns on the right hand side of Table 1.

Table 1 — Qualification

	Qualification		
	Technical understanding of the item to be developed	Knowledge of functional safety (safety standards, safety processes and safety engineering)	Methodological know-how for the corresponding task
Functional safety management	Can be acquired during the project	Expert knowledge	Project management experience
Planning and coordination of safety activities during the lifecycle phases	Can be acquired during the project	Expert knowledge	Project management experience
Developing the safety case	In-depth technical understanding	Expert knowledge	-
Escalation of findings concerning functional safety	Basic experience	Basic experience	Management experience
Safety requirements management	Can be acquired during the project	Knowledge of the safety requirements	Knowledge of requirements management
Developing derived safety requirements	In-depth technical understanding	Knowledge of the safety requirements	Knowledge of requirements formulation
Implementation of safety requirements	In-depth technical understanding	Basic understanding	According to task
Carrying out safety analyses[a]	Can be acquired during project	Knowledge of failure models	Analysis methods
Planning, carrying out and documenting verification and validation	In-depth technical understanding	Knowledge of failure models	Experience, planning and performance of tests, simulations, vehicle testing etc.
Configuration management	In-depth technical understanding	Basic knowledge	Experience in change management, version management and variant management
Confirmation measures[b]	Can be acquired during project	Expert knowledge of the scope involved	Assessment experience if applicable
Preparation and implementation tasks for production and operation	Basic knowledge	Basic knowledge	Experience with production or service, Human Machine Interface (HMI) respectively

[a] Hazard analysis and risk assessment, FMEA, FTA, software criteria for coexistence

[b] Audit, review, assessment

Examples of authorities to successfully perform the activities in the first column of Table 2 are described and classified in the three columns on the right-hand side of Table 2.

Table 2 — Authority

	Authority		
	Access to technical information regarding the item	Access to staff resources and influence on scheduling	Influence on communication and decision-making paths
Functional safety management	X	X	Safety release
Planning and coordination of safety activities during lifecycle phases	X	X	Adoption of safety plan
Developing the safety case	X	(X)	Safety release
Escalation of safety anomalies	X	-	Achieving decisions on measures
Safety requirements management	X	-	Ensuring requirements management process
Developing derived safety requirements	X	(X)	-
Implementation of safety requirements	X	(X)	-
Carrying out safety analyses[a]	X	(X)	Communicating results
Planning, carrying out and documenting verification and validation	X	(X)	Communicating results
Configuration management	X	-	Ensuring configuration management process
Confirmation measures[b]	X	-	-
Preparation and implementation tasks for production and operation	X	-	-

[a] Hazard analysis and risk assessment, FMEA, FTA, software criteria for coexistence

[b] Audit, review, assessment

\- No specified requirement for authority

(X) Partial authority

X Authority

9 Understanding of safety cases

9.1 Interpretation of safety cases

The safety case can be interpreted in several ways. A possible interpretation is given in this clause.

The purpose of the safety case can be defined in the following terms:

A safety case communicates a clear, comprehensive and defensible argument (supported by evidence) that a system is acceptably safe to operate in a particular context.

The following are important considerations in the above definition: Above all, the safety case exists to communicate an argument. It is used to demonstrate how it is possible to reasonably conclude that a system

is acceptably safe from the available evidence. A safety case is a device for communicating ideas and information, usually to a third party. In order to do this convincingly, it will be as clear as possible. Given that absolute safety is an unobtainable goal, safety cases can convince that the system is safe enough (free of unreasonable risk). The safety case will also clearly define the context within which safety is being argued. Safety cases can be produced for entire vehicles or subsystems.

There are three principal elements of a safety case, namely: the requirements, the argument, and the evidence. The relationship between these three elements is depicted in Figure 11

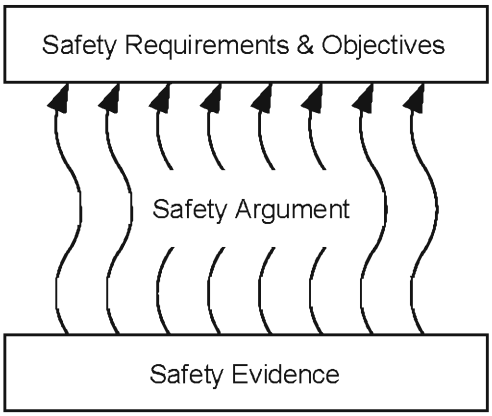

Figure 11 — Key elements of a safety case (see [1])

The safety argument communicates the relationship between the evidence and the objectives. The role of the safety argument is often neglected. It is possible to present many pages of supporting evidence without clearly explaining how this evidence relates to the safety objectives. Both the argument and the evidence are crucial elements of the safety case and go hand-in-hand. An argument without supporting evidence is unfounded, and therefore unconvincing. Evidence without an argument is unexplained, it can be unclear that, or how, safety objectives have been satisfied. Safety cases are typically communicated to third parties through the development and presentation of safety case reports. The role of a safety case report is to summarise the safety argument and then reference the external reports capturing the supporting safety evidence (e.g. test reports).

Safety arguments have been most typically communicated in safety case reports through narrative text. Narrative text can describe how a safety objective has been interpreted, allocated and decomposed, ultimately leading to references to evidence that demonstrate fulfilment of lower-level safety claims. Alternatively, it is becoming increasingly popular to use graphical argument notations (such as the Goal Structuring Notation [1]) to visually and explicitly represent the individual elements of any safety argument (requirements, claims, evidence and context) and the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument).

9.2 Common types of safety arguments

A safety argument that argues safety through direct appeal to features of the implemented product (e.g. the behaviour of a timing watchdog) is often termed a product argument. A safety argument that argues safety through appeal to features of the development and assessment process (e.g. the design notation adopted) is often termed a process argument.

9.3 Safety case development lifecycle

It is now widely recognized that the safety case development cannot be left as an activity to be performed towards the end of the safety lifecycle. A number of anomalies can result if this is done, including large amounts of re-design resulting from a belated realisation that a satisfactory safety argument cannot be constructed; less robust safety arguments being presented; and lost safety rationale. Instead, the safety case

development is treated as an incremental activity that is integrated with the rest of the design and safety lifecycle. Such an approach typically results in the production and presentation of safety case reports at a number of stages during the development of a project. For example, a preliminary safety case report can be produced after the definition and the review of the system requirements specification; an interim safety case report can be produced after the initial system design and preliminary validation activities; and a pre-operational safety case report can be produced just prior to in-service use, including implementation-based evidence of the satisfaction of the system requirements.

9.4 Safety case maintenance

Typically, safety case arguments will initially be constructed and presented prior to the system in question being in widespread operational use. The case is often therefore based on estimated and predicted system and operator behaviour rather than observed evidence. Throughout the operational life of any system, the corresponding safety case might be challenged by additional safety evidence arising from operation, changes and updates to a design, and a shifting regulatory context. In order to maintain an accurate account of the safety of the system, such challenges are assessed for their impact on the original safety argument.

9.5 Safety case review and acceptance

A safety case based regime requires a strong review element. Typically, one department is responsible for preparing the safety case. At least one other department, or organization, will be responsible for reviewing and accepting the safety case. Safety cases are, by their nature, often subjective. The objective of the safety case development, therefore, is to obtain mutual acceptance of this subjective position.

Assessing the sufficiency of a safety case will consider the relevance, the coverage and the integrity of the arguments and evidence presented. A thorough review will also consider if counter-evidence exists that can potentially undermine, or refute, the arguments being presented. Arguments based upon comprehensive analytical evidence are often considered more compelling than those based upon inductive evidence (where extrapolation is used). Safety cases based solely upon process arguments are also typically regarded as weaker than those presenting direct product arguments.

10 Safety element out of context

10.1 Safety Element out of Context Development

A Safety Element out of Context (SEooC) is a safety element for which an item does not exist at the time of the development. A SEooC can either be a subsystem, a software component, or a hardware component.

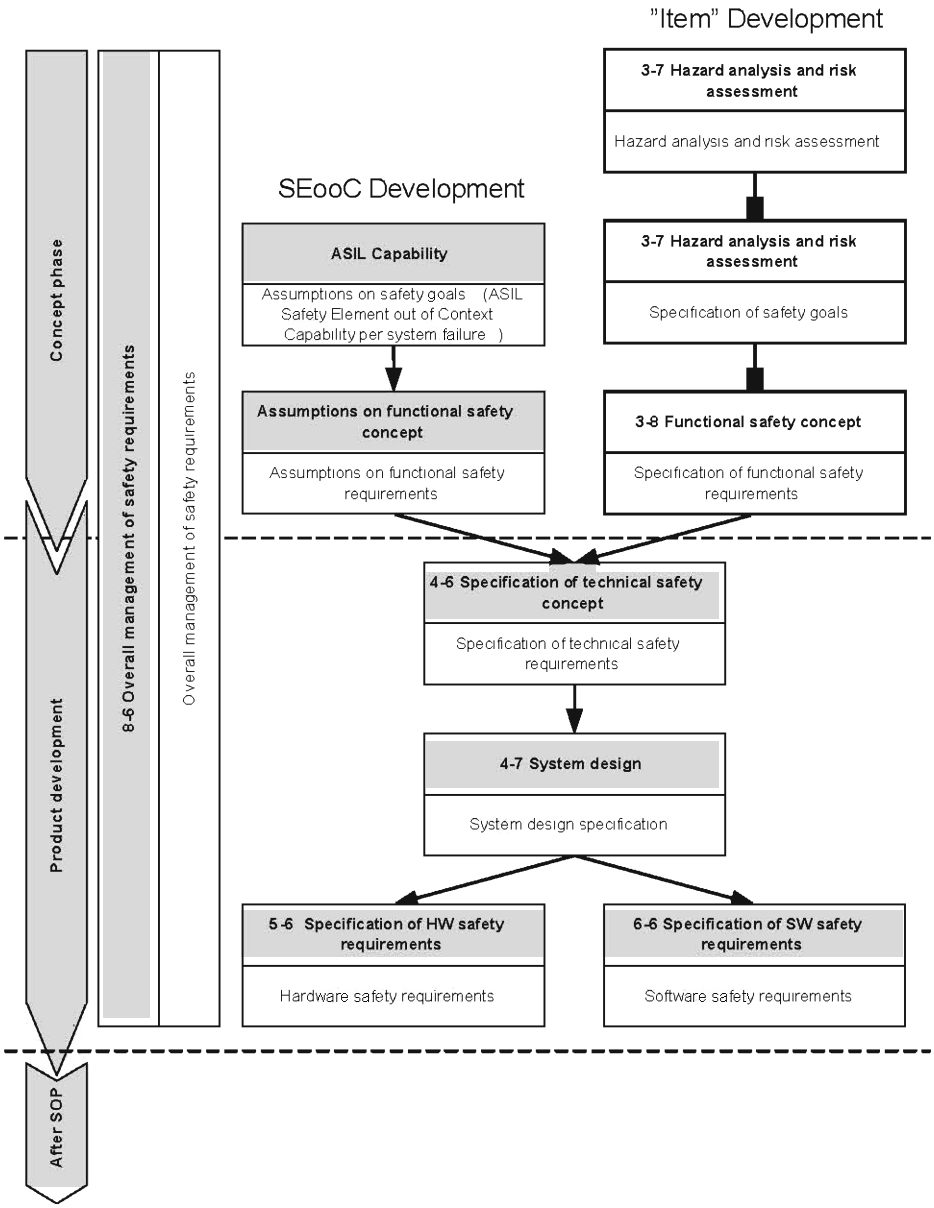

Figure 12 — Safety Element out of Context Development lifecycle for a sub-system

The subphases depicted in ISO°26262-3, Figure 2 can be replaced with the ASIL capability assignment and ASIL Safety Element out of Context assumptions as depicted in Figure 12 for a subsystem development.

10.2 Definition of a SEooC

— A SEooC is never an item.

— A SEooC can either be a subsystem, a hardware component, or a software component.

— Typically, requirements at higher levels remain in the status "assumed" (see ISO°26262-8, Clause°5) and will be confirmed when the SEooC is used in an item development.

— The correct implementation of the assumed requirements will be verified during the SEooC development, but the validation takes place during the item development. The development of a SEooC starts at a

certain level of requirements and design, and all information on requirements or design prerequisites, are pre-determined in the status "assumed".

— Non-functional requirements might be in the status "assumed" at the same level of requirements where functional ones are in the status "accepted".

10.3 Use cases

10.3.1 General

Below, some typical examples of a SEooC, namely a subsystem, a hardware component, and a software component are given.

10.3.2 Development of a Sub-system out of Context

The product development can start at the hardware level as a SEooC. This implies that the prerequisite work products are replaced by assumptions on ASIL capabilities. In this case, the system design specification (ISO°26262-4, Clause°7) and the technical safety concept (ISO°26262-4, Clause°6) that provide the ASIL attributes, are replaced by assumptions.

10.3.3 Development of a Hardware component out of Context

The product development can start at the hardware level as a SEooC. This implies that the prerequisite work products are replaced by assumptions on ASIL capabilities. In this case, the system design specification (ISO°26262-4, Clause°7) and the technical safety concept (ISO°26262-4, Clause°6) that provide the ASIL attributes, are replaced by assumptions.

10.3.4 Development of a Software component out of Context

Similarly the product development can start at the software level as a SEooC. In this case, the software is identified as a component in the system design. The system design specification (ISO°26262-4, Clause°7) and the technical safety concept (ISO°26262-4, Clause°6) that provide the ASIL attributes, are replaced by assumptions. However, the software development out of context can also start with either the software architectural design (ISO°26262-6, Clause°7) or with the software unit design and implementation (ISO°26262-6, Clause°8). Then, the software safety requirements (ISO°26262-6, Clause°6), and the architectural design specification (ISO°26262-6, Clause 7), respectively, can be replaced by assumptions.

10.4 Using a SEooC in an item development

Requirements of the SEooC in the status "assumed" are confirmed during item development. If a subsystem is the SEooC, with a lifecycle as depicted in Figure 12, the verification report at the system level, can show that the technical safety concept formulated out of context is consistent with the functional safety concept and with the preliminary architectural design, respectively. Once this is done, the argument of using the SEooC in a specific item is fulfilled.

Similarly, the applicable verification report can verify that a SEooC developed, at any level, is consistent with the requirements in the context where it is used. For example, when a software unit developed out of context is used, a verification of the software unit specification can report that the requirements in the software architectural design specification are met. This verification report can be produced when the development SEooC is finished and the item development reaches the phase where requirements on the safety element are formulated.

11 ASIL decomposition

11.1 Objective of the ASIL decomposition

The objective of the ASIL decomposition, which is defined in ISO°26262-1, is to provide rules to comply with a safety goal or a safety requirement through a combination of derived safety requirements, applying redundancy. ASIL decomposition can result in redundant requirements and their tailored ASILs implemented by sufficiently independent elements.

11.2 Description of the ASIL decomposition

ASIL decomposition refers to the allocation of a safety requirement amongst redundant architectural elements of the item. Redundant in this context does not necessarily imply classical modular redundancy. For example, the "E-throttle" safety concept can be viewed as containing redundant architectural elements, i.e. the main processor and the monitoring processor, both of which are independently capable of initiating a defined safe state, which is to go to the idle mode.

Since ASILs do not have an associated failure rate, ASIL decomposition is understood in terms of the methods and measures applied to reduce the likelihood of systematic failures.

In such a decomposed architecture, the relevant safety goal is only violated if both elements fail simultaneously.

The supported decompositions in ISO°26262 are:

— ASIL D → ASIL C(D) + ASIL A(D)

— ASIL D → ASIL B(D) + ASIL B(D)

— ASIL D → ASIL D(D) + QM(D)

— ASIL C → ASIL B(C) + ASIL A(C)

— ASIL C → ASIL C(C) + QM(C)

— ASIL B → ASIL A(B) + ASIL A(B)

— ASIL B → ASIL B(B) + QM(B)

— ASIL A → ASIL A + QM(A)

NOTE1 Additional activities are performed at the initial ASIL.

NOTE2 ASIL decomposition can be applied across independent ECUs, or across independent elements within the same ECU.

11.3 Rationale for the ASIL decomposition

Standards such as IEC 61508 generally allow a system with an allocated SIL n requirement to be composed of two SIL (n-1) elements, provided adequate independence of the elements is demonstrated.

Based on the approximate mapping in ISO°26262-9, Clause°5 it can be seen immediately that the following decompositions satisfy this principle:

— ASIL D → ASIL B(D) + ASIL B(D)

— ASIL B → ASIL A(B) + ASIL A(B)

The remaining decompositions are:

— ASIL D → ASIL C(D) + ASIL A(D)

— ASIL C → ASIL B(C) + ASIL A(C)

Splitting this best practice requirement over 2 independent elements is based on the following assumptions:

— As the elements are diverse, i.e. are not identical copies of each other, the errors that can violate the safety goal will be different.

— For the safety goal to be violated there will be an error in each item, i.e. multiple errors will exist.

— The reduced application of techniques used on each item means that the likelihood of errors remaining in each item will increase.

— However, due to the diversity of both the elements and the ASIL requirements, the likelihood that multiple errors remain is similar to that of the original item when compliant with the original ASIL requirements.

11.4 An example of ASIL Decomposition

11.4.1 General

The item and the requirements described in this clause are examples. The safety goal, its ASIL, and the following requirements are only designed to illustrate the ASIL decomposition process. This example does not reflect what the application of ISO°26262 on a similar real-life example would be.

11.4.2 Item definition

Consider the example of a powered sliding door. The user can operate a switch inside the vehicle to activate the door motion. The user can also use a switch on the remote control to open the door. These two possibilities are included in the opening request in Figure.13.

11.4.3 Hazard and risk analysis

The failure considered in the analysis is a requested or unrequested opening of the door, while driving at high speed.

This malfunction might be hazardous above a speed of 15 km/h, which might lead to an E4 classification.

A passenger might fall out of the vehicle and be seriously injured, which might lead to an S3 classification.

Harm can be avoided by staying in the car with the safety belt on; a controllability of 90% might be assumed, which can lead to a C2 classification.

Other failures might occur, but these are not considered in this example.

11.4.4 Associated safety goal

Not to open the door while the vehicle speed is higher than 15 km/h : ASILC.

11.4.5 Preliminary architecture and safety concept

11.4.5.1 General

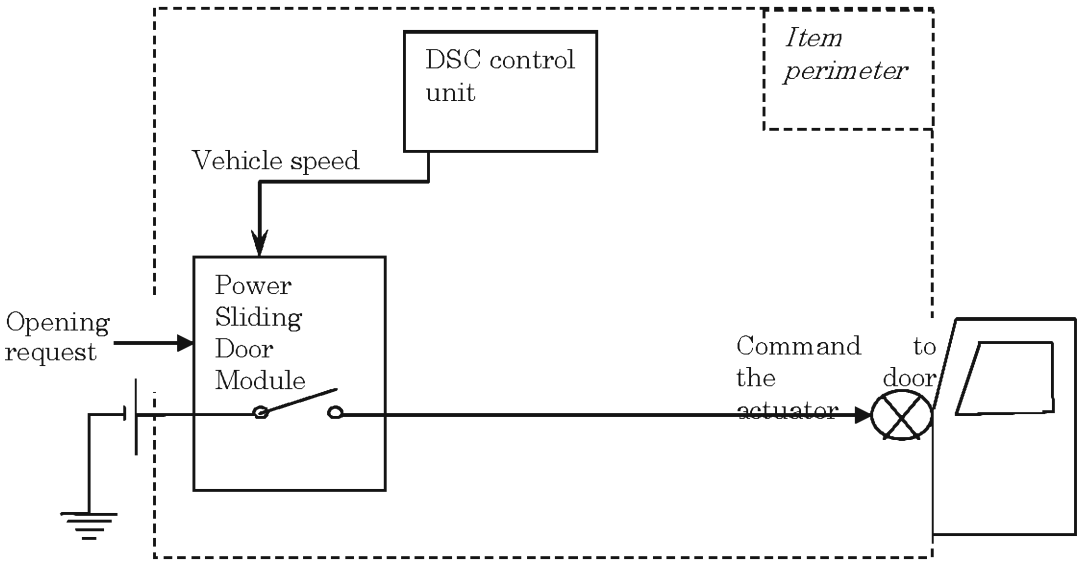

Figure 13 — Item perimeter of the powered sliding door

11.4.5.2 Purpose of the elements (preliminary architecture):

— The Dynamic Stability Control (DSC) unit provides the Power Sliding Door Module (PSDM) with the vehicle speed.

— The PSDM monitors the opening requests, tests if the vehicle speed is below 15 km/h, and drives the power of the door actuator.

— The door actuator moves the door when it is powered.

11.4.6 Functional safety Concept

11.4.6.1 General:

— Requirement A 1 : The DSC will send the accurate vehicle speed information to the PSDM. =>ASIL C.

— Alternatively : Unwanted transition of the vehicle speed information below 15 km/h will be prevented => ASIL C.

— Requirement A 2 : The PSDM will allow the powering of the actuator only if the vehicle speed is below 15 km/h => ASIL C.

— Requirement A 3 : The door actuator will only open the door when powered by the PSDM. => ASIL C.

11.4.6.2 Evolved Safety Concept of the item

The developers can choose to introduce a redundant element, as illustrated in Figure 14. By introducing this redundant element, the PDSM might be developed with an ASIL that is lower than ASIL C, in accordance with the results of an ASIL decomposition.

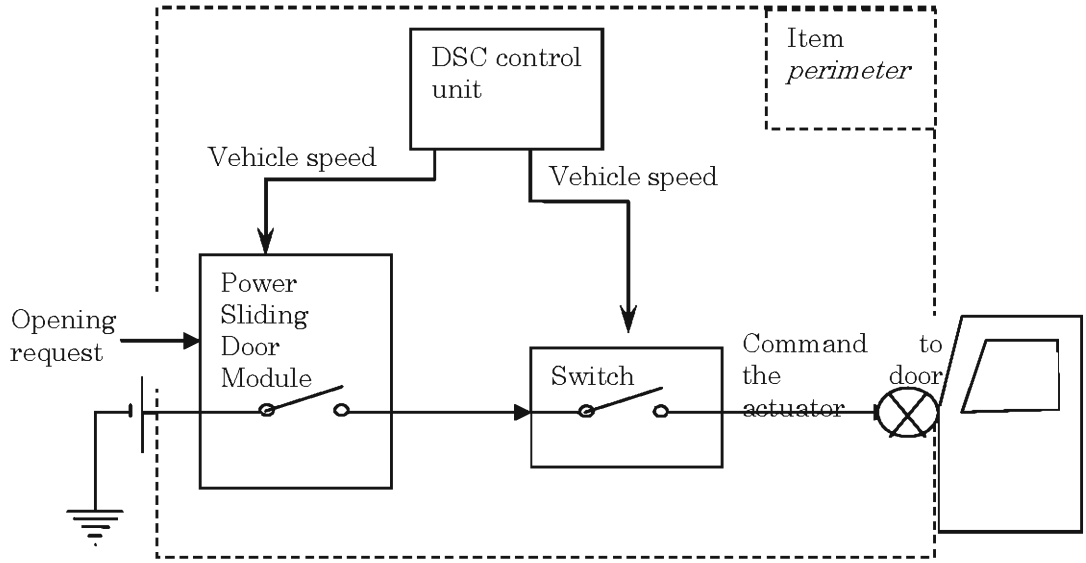

Figure 14 —second iteration on the item design

Purpose of these elements (evolved architecture):

— The DSC control unit provides the ECUs with the vehicle speed.

— The Power Sliding Door Module monitors the opening requests, tests if the vehicle speed is below 15 km/h, and drives the power of the door actuator.

— The switch is on the power line between the PSDM and the door actuator, and switches on, if the speed is below 15 km/h and off whenever the speed is above 15 km/h, regardless of the state of the power line (its power supply is independent).

— The door actuator moves the door when it is powered.

Functional safety requirements:

— Requirement B1: the DSC will send an accurate vehicle speed information to the PSDM => ASIL C.

— Alternatively : the unwanted transition of the vehicle speed information below 15 km/h will be prevented => ASIL C.

— Requirement B 2 : the PSDM will allow the powering of the actuator only if the vehicle speed is below 15 km/h => ASIL x (see table 3).

— Requirement B 3 : the DSC will send the accurate vehicle speed information to the switch => ASIL C.

— Requirement B 4 : The switch will be in an open state if the vehicle speed is above 15 km/h => ASIL y (see table 3).

— Requirement B 5: The door actuator will only open the door when powered by the PSDM and the switch is closed => ASIL C.

To enable an ASIL decomposition, the developers can add an independency requirement:

— Requirement B 6 : The PSDM and switch will be independently implemented: ASIL C.

Therefore, requirements B2 and B4 implement redundantly the fulfilment of the safety goal, and an ASIL decomposition can be applied..

Table 3 — Possible decompositions

Requirement B2 : ASIL x	Requirement B4 : ASIL y
ASIL C (C)	QM(C)
ASIL B(C)	ASIL A(C)
ASIL A(C)	ASIL B(C)
QM(C)	ASIL C(C)

12 Criteria for the coexistence of sub-elements - Description

See ISO°26262-9, Clause°6.

Bibliography

[1] IEC 61508 (all parts), Functional safety of electrical/electronic/programmable electronic safety-related systems

[2] T. P. Kelly, Arguing Safety – A Systematic Approach to Safety Case Management, DPhil Thesis, Department of Computer Science, University of York, UK, 1998