IEEE SA
STANDARDS
ASSOCIATION

IEEE SA
WHITE PAPER

A LANDSCAPE FOR THE DEVELOPMENT OF DEPENDABLE MACHINES

Authored by

IEEE P2851 Working Group

TRADEMARKS AND DISCLAIMERS

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

The ideas and proposals in this specification are the respective author's views and do not represent the views of the affiliated organization.

ACKNOWLEDGEMENTS

Special thanks are given to all the members of the IEEE P2851 WG.

The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA

PDF: STDVA24820 978-1-5044-7770-3

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA DOCUMENTS

This IEEE Standards Association ("IEEE SA") publication ("Work") is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the activity that produced this Work. IEEE and the IEEE P2851 WG expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE P2851 WG disclaim any and all conditions relating to results and workmanlike effort. This document is supplied "AS IS" and "WITH ALL FAULTS."

Although the IEEE P2851 WG members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE-SA OR ICAP MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder.

This Work is published with the understanding that IEEE and the IEEE P2851 WG members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

TABLE OF CONTENTS

A LANDSCAPE FOR THE DEVELOPMENT OF DEPENDABLE MACHINES

ABSTRACT

This white paper, developed in the framework of the IEEE P2851 standardization project, describes a landscape of activities to be performed to address the dependability of intelligent machines during their development and usage in the field.

In this context, the landscape is intended as the definition of the entirety of the activities that are executed within the autonomous machine dependability lifecycle. Dependability here is intended as the property of an autonomous machine to perform reliably, safely, securely, in a time-deterministic manner, etc.

It should be noted that despite autonomous machines introducing specific dependability challenges, the scope of this document is not limited to full autonomy; rather, it covers the whole spectrum of degrees of automation, from no automation to full automation.

Also worth noting is that nowadays dependable machines are extremely connected. Using automotive as an example, Connected Automated Vehicles (CAV) involves interactions with other vehicles and infrastructure (V2X) and also link with the cloud for functions such as fleet management, teleoperation, maps, Over the Air (OTA) updates, etc. Therefore, the proposed lifecycle is considered end to end, covering the dependability aspects of connectivity as well.

The landscape includes the definition of *needs* for each activity, in terms of methodologies, description languages, data models, and databases that have been identified as necessary or critical to perform those activities.

The goal of this work is to provide structures and directions to allow a seamless exchange of information and interoperability between activities at the same or different level of abstraction.

1. ABOUT IEEE P2851

The IEEE P2851 initiative [1] has been created upon a request of the IEEE Computer Society Special Technical Community (STC) on Reliable, Safe, Secure and Time Deterministic Intelligent Systems (RSSTDIS) [2].

The STC noted that the development of safety and cybersecurity critical systems is rapidly growing due to the expansion of new applications such as automated driving or autonomous mobile robotics.

Standards such as ISO 26262 [4], IEC 61508 [5], and many others define the complete set of activities that need to be performed, requiring companies at different levels of the supply chain to tailor the analysis and verification activities that apply to them and deliver results to other levels of the chain for which other requirements are applicable. EDA vendors have already started to provide tools to automate those activities. However, currently, there are not common methodologies, languages, or formats to provide those results. Even worse, today there is not a cohesive view of all those activities that are executed, often in a disjointed way. As a result of this gap, companies are struggling with many different types of methodologies and description languages and are investing valuable time and effort to reconsolidate, compare, integrate, and combine the data. For this reason, the safety critical community is urgently asking for a solution to accelerate the safety engineering process while reducing risks and costs.

In that context, the IEEE P2851 standardization initiative has been started to define and deliver a comprehensive view of all those activities. One objective is to ultimately define a format for exchange/interoperability for analysis and verification activities (including requirements, safety cases, etc.) and facilitate companies in delivering results in a consistent way. Additionally, the work products of this standardization initiative will enable interoperability between tools.

The initial scope of IEEE P2851 was identified by the RSSTDIS STC as related to Functional Safety at IP and SoC level. However, during the IEEE P2851 kickoff, it was already identified that the scope had to be expanded to systems and items, including software, as well as expanded to cybersecurity and the other properties of dependable systems as in the scope of the RSSTDIS STC: reliability, safety, security, time determinism, and others. Therefore, the end goal of IEEE P2851 is to cover the full spectrum of dependability topics.

2. THE LIFECYCLE APPROACH OF P2851

Several international and industry standards (e.g., ISO 26262 [4], IEC 61508 [5]) propose various details of development lifecycles. These details vary in scope, format, terminology, and consistency of language. This lack of uniformity already presents industry challenges to developments that need to address multiple standards. Given that IEEEE P2851 spans several industry domains, it is essential to define the lifecycle in a manner that provides a common language and understanding to enable translation with lifecycles defined by other standards.

2.1. LIFECYCLE BACKGROUND

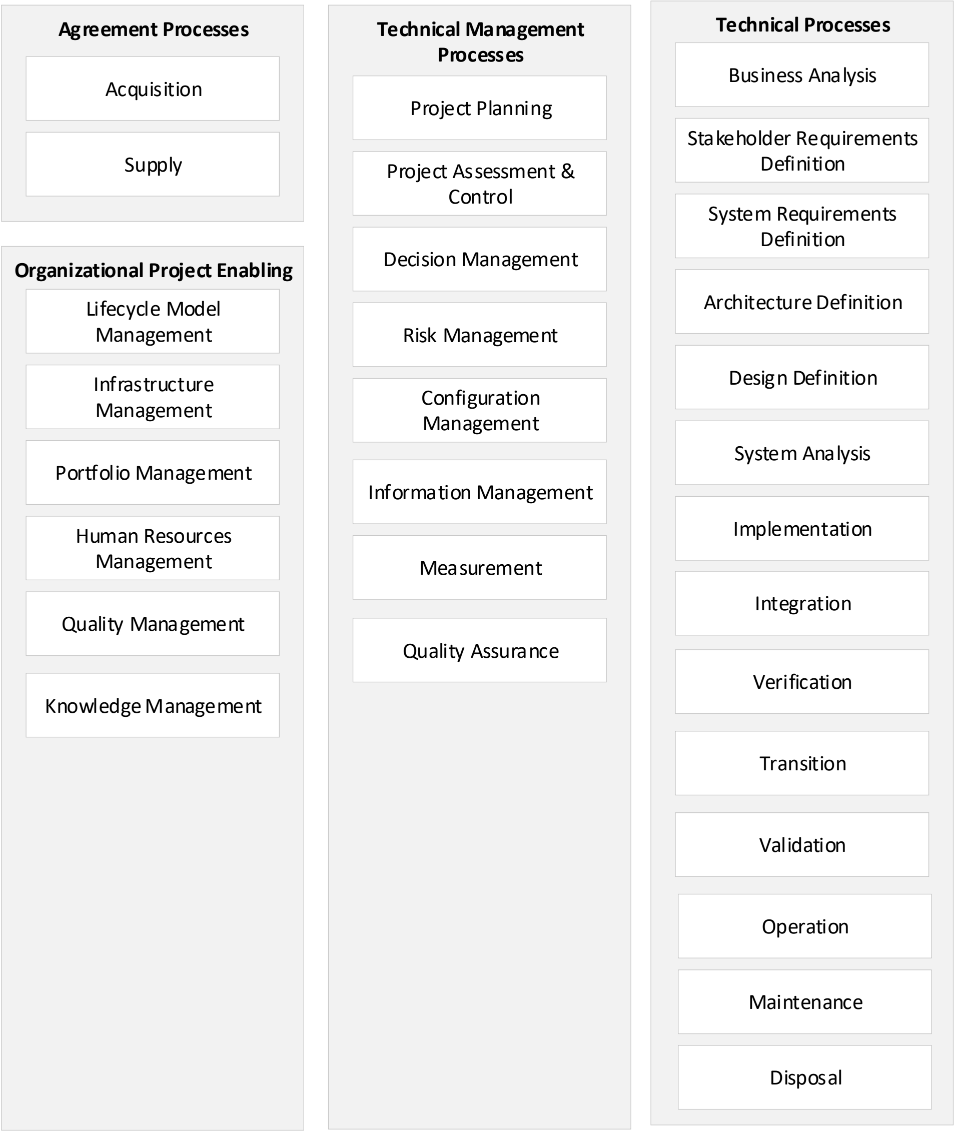
ISO/IEC/IEEE 15288 [6] and ISO/IEC/IEEE 24748 [7] were developed to promote uniformity in lifecycle models and hence are leveraged by IEEE P2851. The following paragraphs provide a brief overview of these standards.

ISO/IEC/IEEE 15288 defines a process as a set of interrelated activities that transform inputs into outputs. A process is described in terms of five characteristics: title, purpose, outcome, activities, and tasks necessary to execute the transformation. This level of description reduces the potential for misinterpretation and improves the ability to translate with other standards. ISO/IEC/IEEE 15288 goes on to define a complete set of processes necessary to enable the development of a product from conception to decommissioning. It groups these processes into the following four categories:

- **Agreement processes:** Concerned with supply and acquisition between organizations and within organizations.
- **Technical processes:** Concerned with technical actions that transform the product from one form to another form.
- **Technical management processes:** Concerned with managing resources and assets to enable the execution of the technical processes in a manner that meets what has been agreed via the agreement processes.
- **Organizational project enabling processes:** Concerned with establishing the environment in which development projects are conducted.

FIGURE 1 summarizes these processes. The intent is not to say that this is the only set of processes by which a product may be developed. Indeed, ISO/IEC/IEEE 15288 allows for the tailoring of processes and acknowledges that an organization may identify additional useful processes.

FIGURE 1 — **ISO/IEC/IEEE 15288 System Lifecycle processes**



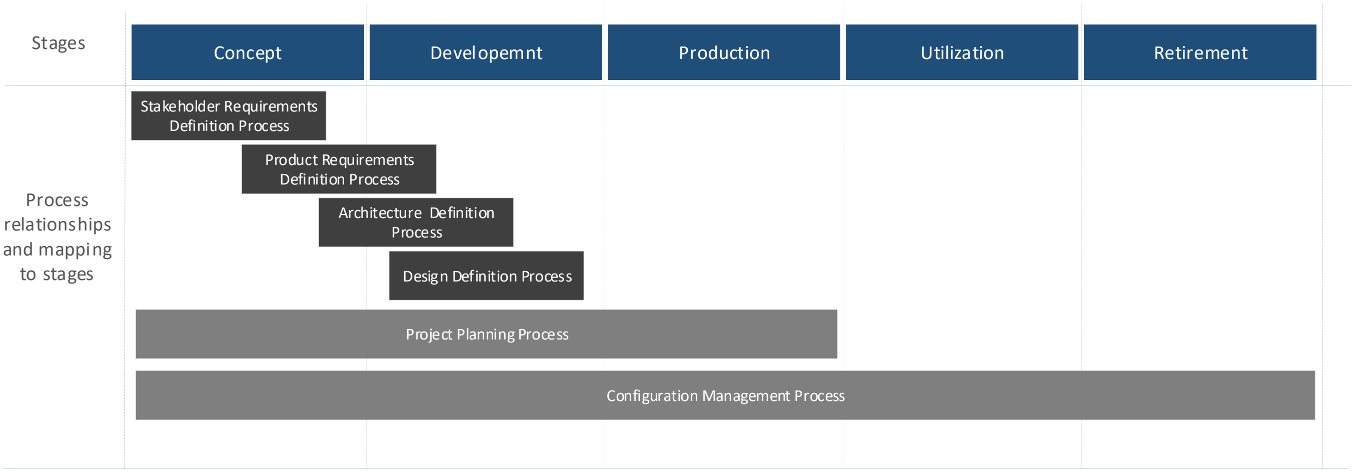
ISO/IEC/IEEE 24748 provides guidelines on how ISO/IEC/IEEE 15288 processes may be arranged to enable product development. A lifecycle model is expressed in terms of the specific processes employed and the relationships and interdependencies between them. It defines a phase to represent a major lifecycle period that relates to the state of the products description or realization. Entry and exit criteria may be defined for each stage, enabling support for strong project control and monitoring. It presents a representative lifecycle model consisting of the following six stages:

- **Concept:** Initial definition of needs and requirements. Exploration of alternative options and proposal of viable solutions.
- **Development:** Refine requirements, create solution description, build system, verify, and validate the system.

- **Production:** Produce system, inspect, and test.
- **Utilization:** Operate system.
- **Support:** Provide sustained system capabilities.
- **Retirement:** Store, archive, or dispose of a system.

FIGURE 2 presents a highly simplified view of a potential mapping for a subset of processes into stages.

FIGURE 2 — Potential mapping of processes to stages

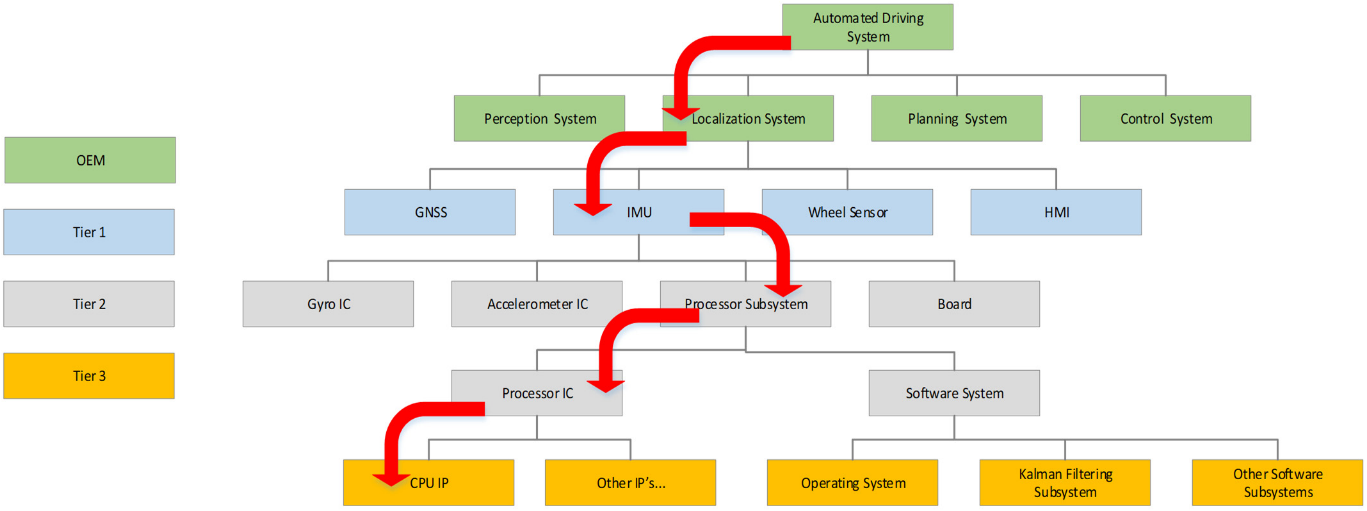

Stages
Concept
Developemnt
Production
Utilization
Retirement
Process relationships and mapping to stages
Stakeholder Requirements Definition Process
Product Requirements Definition Process
Architecture Definition Process
Design Definition Process
Project Planning Process
Configuration Management Process

The intent is not to say that this is the only process to stage mapping that could be assigned to development.

Both ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 24748 highlight the importance of the iterative and recursive nature of processes in product development.

Recursive application of process, i.e., the same process applied at different levels of granularity, is expected, as requirements at one level create requirements on lower levels. FIGURE 3 provides an automotive centric example of this recursive nature, taking an automated driving system as an example. The red arrows describe the top-down break down of requirements from top level (item) to the bottom (semiconductor IP level). The automated driving subsystem will place requirements on the localization system. The localization system in turn comprises various subsystems including a Global Navigation Satellite System (GNSS) and an Inertial Measurement Unit (IMU), each of which inherits requirements. The IMU in turn is comprised of multiple integrated circuit and software components all of which inherit their own requirements.

FIGURE 3 — **Example of recursive application of processes**

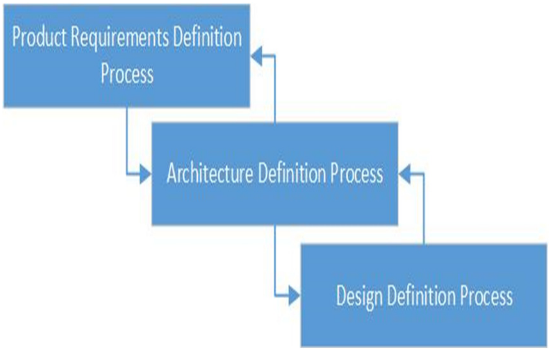


It should be recognized that each of these developments will go through their own development lifecycle and must interface at the appropriate points with the needed levels of granularity.

Iterative application of process, i.e., a process repeated on the same level of product granularity, is not only appropriate but also expected, to account for evolving knowledge and trade-offs. FIGURE 4 presents a simplified view of the iterative relationship between the requirements, architecture, and design processes. The intent is to highlight that some requirements cannot be derived until some portion of the architecture or design evolves. This iterative nature applies to all processes, not only the requirements process.

ISO/IEC/IEEE 24748 further introduces the notion of a development strategy that prescribes certain processes and activities that may be performed sequentially, repeated and/or combined. The lifecycle model and processes are mapped to the desired development strategy. Examples of development strategies include Waterfall, V-model, and Agile developments.

FIGURE 4 — **Iterative relationship between requirements, architecture, and design processes**


Product Requirements Definition Process
Architecture Definition Process
Design Definition Process

2.2. THE DEPENDABILITY LIFECYCLE

The IEEE P2851 landscape finds it useful to introduce the notion of a Product Dependability Lifecycle (PDL). The term *dependability* has been selected to cover a broad spectrum of properties, including Functional Safety, which is the safety of the intended functionality, cybersecurity, reliability, availability, maintainability, and timing guaranteed.

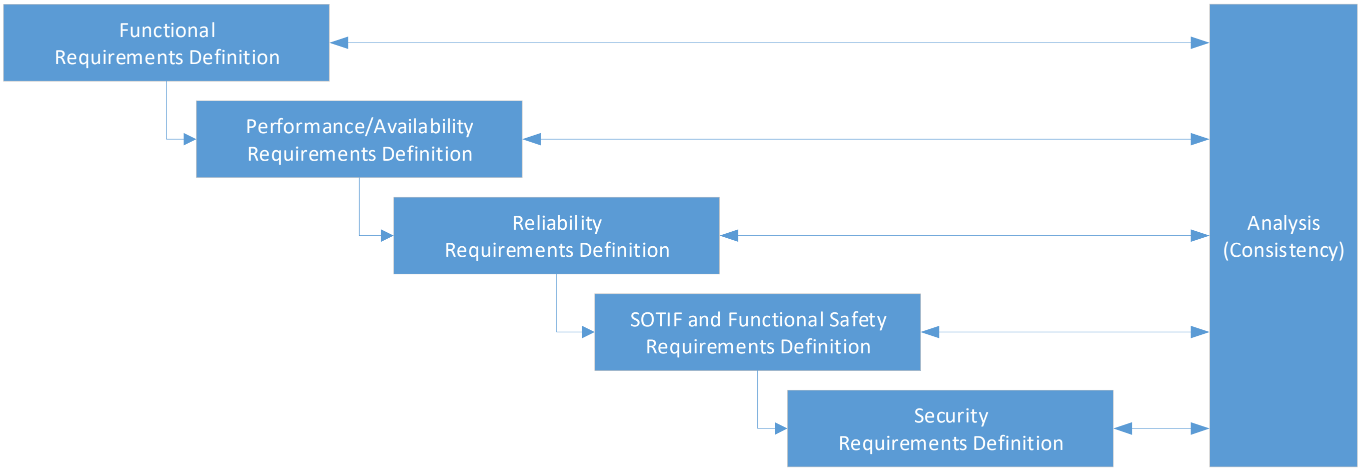
It is important to recognize that these properties are nothing more than requirements for product development. These requirements are uncovered during the requirements definitions phases of the product development. In an IEEE P2851 PDL, requirements can serve, for example, the purposes of:

- **Functional:** The functions supported by the product. To evaluate the performance of the prescribed functions, the function definition shall be atomic in nature.
- **Performance:** The performance targets necessary to support functional requirements. This would include any timing behavior and **availability** requirements. In an integrated circuit context such requirements could include performance characteristics as diverse as CPU clock frequency, channel phase mismatch, or signal-to-noise targets.
- **Reliability:** Including mission profile and product lifetime. Many other reliability requirements are often inherited from auto reliability standards, e.g., AECQ.
- **Safety of the Intended Functionality (SOTIF):** Typically, understood as the nominal performance necessary to avoid hazards. This may be equal to or a subset of the Performance requirements.
- **Functional Safety:** Requirements related to the avoidance of hazards due to malfunctions. These may introduce their own real-time requirements, for example, the need to execute a safety mechanism within certain timing guarantees.
- **Cybersecurity:** Requirements related to the avoidance of hazards due to deliberate attacks.

It is important to recognize that functions share resources (such as memory space, use of sub-functions). Hence, it is critical that a holistic solution must be arrived at, which always fulfills all requirements. This can be achieved by a combination of smart refinement of requirements, appropriate architecture choice and efficient implementation of the functions. This is a classical system engineering challenge. For example, consider a RADAR transceiver integrated circuit. An on-chip safety mechanism to provide coverage for random faults on the RX and TX paths must share IC resources with the main radar functionality. An architecture and implementation must be found that fulfills the combined performance requirements.

FIGURE 5 presents a simplified view of the need to ensure requirements consistency within an iteration. The iteration could require jumping back to any higher level of the flow as needed. The staggering of Functional, Performance, Reliability, Safety, and Cybersecurity requirements is to convey that first the function should be understood to an adequate level, which then allows performance to be revealed, which in turn allows the safety requirements to be revealed via the appropriate analysis (FMEDA, FTA, etc.).

FIGURE 5 — **Need to find an appropriate trade-off between requirements**


Functional Requirements Definition
Performance/Availability Requirements Definition
Reliability Requirements Definition
SOTIF and Functional Safety Requirements Definition
Security Requirements Definition
Analysis (Consistency)

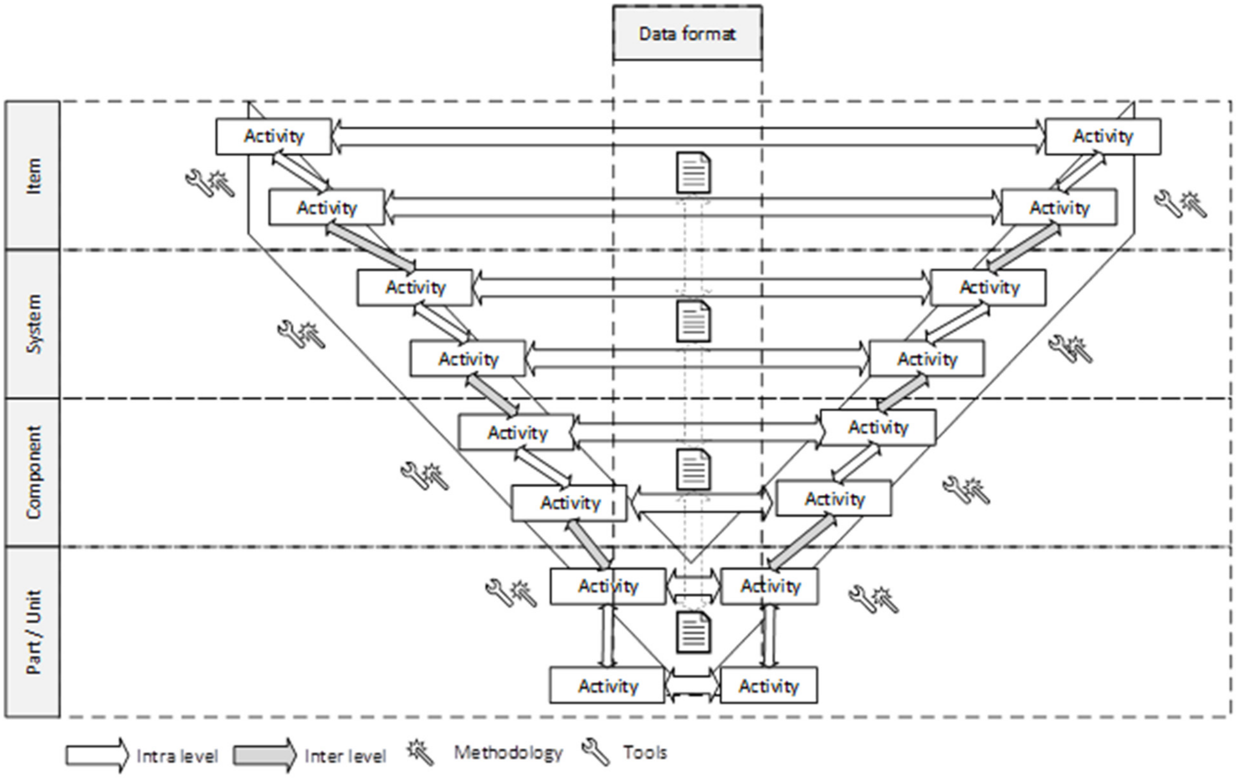
However, this intent is not to suggest that this is the only possible flow, rather it is an example. Note, in this diagram Safety includes both Functional Safety and SOTIF for simplicity.

How the requirements are elicited on a real program will depend on the specificity of that program. For example, some Functional Safety and SOTIF requirements may be elicited during the stakeholder requirements definition process during customer discussion.

IEEE P2851 will leverage the V-model (FIGURE 6) when discussing the development strategy as it best represents the connection between design activities (left side of the V) and verification activities (right side of the V).

Each level (item, system, component, and part/unit) includes one or more activities, belonging to one or more phases of the PDL. Activities relate to intra or inter level interfaces. Each activity could be linked to methodologies and tools to be executed. It should be recognized that this figure is simplified; each of the item, system, component, and part/unit levels shown will implement their own V-model as already described in relation to FIGURE 1.

FIGURE 6 — **IEEE P2851 V model**



3. AN INTRODUCTION TO LANDSCAPE NEEDS

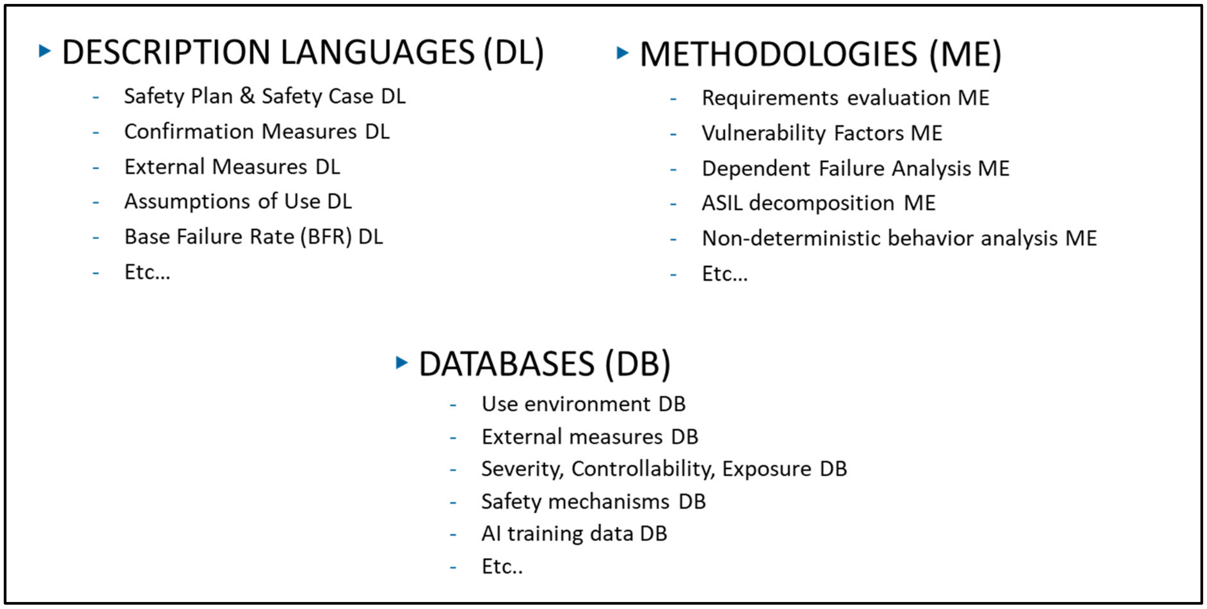
To help guarantee interoperability and ease of exchange of information across the landscape, *needs* are identified for each activity so that they can be transformed into actionable steps of an engineering process.

In the IEEE P2851 context, needs are identified in terms of the following:

- **Description Language (DL):** Language used to describe the framework, syntax, and behavior of an activity. It can comprehend a **data model (DM)**.
- **Methodologies (ME):** Best known methods, guidelines, and principles describing the detailed steps needed to implement an activity, including examples.
- **Databases (DB):** A structured collection or listing of parameters and/or datasets needed as input to or an output of an activity.

FIGURE 7 provides an example of the needs identified by the IEEE P2851 WG.

FIGURE 7 — Example of needs



- DESCRIPTION LANGUAGES (DL)
  - Safety Plan & Safety Case DL
  - Confirmation Measures DL
  - External Measures DL
  - Assumptions of Use DL
  - Base Failure Rate (BFR) DL
  - Etc…
- METHODOLOGIES (ME)
  - Requirements evaluation ME
  - Vulnerability Factors ME
  - Dependent Failure Analysis ME
  - ASIL decomposition ME
  - Non-deterministic behavior analysis ME
  - Etc…
- DATABASES (DB)
  - Use environment DB
  - External measures DB
  - Severity, Controllability, Exposure DB
  - Safety mechanisms DB
  - AI training data DB
  - Etc..

The IEEE P2851 WG described each need with an *identification card*, composed of: Description, Problem Statement, Proposal, Limitations, Levels applicable, and References.

For example, the following tables provide the identification cards for a DL (the Safety Case Description Language), a ME (the Safety Requirements Evaluation Methodology), and a DB (the Dependent Failure Initiators Database).

TABLE 1 Safety Case DL identification card

Element	Text
Description	The safety case is a key work product created during the development of an item or an element and represents the final argument that Functional Safety can be achieved, and that the safety concept can be met. The safety case is compiled during the development of the project and uses the safety plan as input to determine the work products that will be created and will be used as part of the final argument.
Problem statement	IEEE P2851 members believe that there are often missing or even inadequate safety arguments in the safety case and that this is a common issue in the industry due to a lack of examples and dedicated know-how.
Proposal	IEEE P2851 plans to provide one or more examples that will emphasize the benefit of applying an existing description language together with some existing guidelines to a safety case for an integrated circuit or IP. A template/outline (DL) could also be proposed in addition to examples not yet covered by MISRA guidelines. Another path could be to consider requesting the ISO 26262 WG to place

Table continues

Element	Text
	requirements on the formalization of a safety argument in ISO 26262-10, clause 5.3.1.
Limitations	The proposal should allow showing some specific examples but will likely not be exhaustive and not show the specificities of all fields of expertise. It is not the intention of IEEE P2851 to create a new description language, as there are already several available today including Goal Structuring Notation (GSN), Claims Arguments Evidence (CAE), and Structured Assurance Case Metamodel (SACM). Also, at least one auto industry guideline on how to construct a safety case argument already exists, see Guidelines for Automotive Safety Arguments, MISRA.
Levels applicable	The safety case is applicable to all fields of expertise (hardware, software) and all levels in the V-Model
References	Goal Structuring Notation: https://www.goalstructuringnotation.info/ Claims Arguments Evidence: https://claimsargumentsevidence.org/notations/claims-arguments-evidence-cae/ Structured Assurance Case Metamodel: https://www.omg.org/spec/SACM/About-SACM/ Guidelines for Automotive Safety Arguments: https://www.misra.org.uk/Publications/tabid/57/Default.aspx#label-comp2

TABLE 2 Safety Requirements Evaluation ME identification card

Element	Text
Description	Safety Requirements specify the safety functionality of an item, system or element and need to be evaluated at different levels on whether they can be met as defined by the safety concept despite the presence of systematic faults or random hardware faults.
Problem statement	IEEE P2851 WG believes that there is a high risk that Safety Requirements are not evaluated in a complete and systematic way, which could result in a highly increased risk for the item.
Proposal	IEEE P2851 WG proposes to create a methodology that will describe how the evaluation of safety requirements at different levels of the V-Model could be performed in a complete and systematic way. The proposal could also include an example showing, in a specific case, how to demonstrate that the vertical traceability is complete. IEEE P2851 WG also proposes to define the tools needed to evaluate the safety requirements assigned to the item, system, or element under development.
Limitations	None.
Levels applicable	The evaluation of safety requirements is applicable to all fields of expertise (hardware, software, AI) and all levels in the V-Model.
References	SCDL—Safety Concept Description Language—V1.3: https://ssl.scn-sg.com/main/wp-content/uploads/2016/06/SCDLSpecification_v1.3_en.pdf

TABLE 3 Dependent Failure Initiators DB identification card

Element	Text
Description	This DB contains descriptions of events that may serve as initiators for dependent failures, including common cause and common-mode failures.
Problem statement	The completeness of the Dependent Failure Analysis depends on the completeness of the list of initiators. Therefore, it makes sense to have a single and shared collection of initiators. Currently, the information on initiators is dispersed between standards and reference manuals.
Proposal	Create a database containing a description of events that may serve as initiators for dependent failures or common cause/common-mode failures.
Limitations	For each DFA, the list of initiators will need to be carefully reviewed, and tailored, and new initiators added as required.
Levels applicable	Item, system, HW, SW.
References	ISO 26262-5 IEC 61508-2, IEC 61508-3

Appendix A of this document provides the list of the main needs identified by the IEEE P2851 WG. Further details of each need will be developed in the IEEE P2851 standard. It should be noted that not all needs will correspond to a development in the IEEE P2851 standard—some of them could be a reference to other standardization projects. Readers are welcome to provide feedback on those needs by writing to the IEEE P2851 WG using the email addresses listed on the IEEE P2851 home page [1].

4. CONCLUSIONS

This white paper represents the initial milestone of the IEEE P2851 standardization path.

The white paper is being used by the IEEE P2851 WG to communicate the intended direction, collect feedback on the identified needs, and consequently modify the roadmap.

In essence, the key takeaways of this document are as follows:

- There is a need for a holistic view that connects all the different types of requirements defined during the lifecycle of a dependable machine.
- While existing standards ([4], [5], [9], [10], [13] and others) partially cover certain aspects or the high-level requirements of the PDL, that holistic view—to become practical and actionable—must be supported by a set of well-defined methodologies, description languages, and databases that link the stages and related activities of the PDL.
- The role of the IEEE P2851 standard is to define those methodologies, description languages and databases in a way to guarantee interoperability and ease of exchange of information across the value chain.
- To provide a context within which the defined needs are used and interact with one another, IEEE P2851 will also define reference PDLs, specific to automotive, industrial, robotics, medical, and avionics use cases.

The IEEE P2851 activity is aligned with Accellera Functional Safety Working Group (FSWG) [8]. In fact, some of the needs recognized by IEEE P2851 have been already identified by the Accellera FSWG and once complete, the Accellera FS standard can be contributed to IEEE P2851 to be part of, or referenced by, the standard. This approach is like the successful collaboration used to contribute several standards published under the IEEE DASC. These standards include SystemC, SystemVerilog, UPF, UVM, IP-XACT, and others.

5. REFERENCES

The following list of sources either have been referenced within this paper or may be useful for additional reading:

[1] https://sagroups.ieee.org/2851/

[2] https://www.computer.org/communities/special-technical-communities/rsstdis

[3] CAST-32A, Certification Authorities Software Team (CAST), Position Paper, Multi-core Processors.

[4] ISO 26262 (2018), Road Vehicles—Functional Safety.

[5] IEC 61508 (2010), Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems.

[6] ISO/IEC/IEEE 15288 (2015), Systems and software engineering—System life cycle processes.

[7] ISO/IEC/IEEE 24748-1 (2018), Systems and software engineering—Life cycle management—Guidelines for life cycle management.

[8] https://www.accellera.org/activities/working-groups/functional-safety

[9] ISO/DIS 21448, Road vehicles—Safety of the intended functionality.

[10] ISO/SAE FDIS 21434, Road vehicles—Cybersecurity engineering.

[11] SAE ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.

[12] SAE J2980, Considerations for ISO 26262 ASIL Hazard Classification.

[13] UL 4600: Standard for Safety for the Evaluation of Autonomous Products.

APPENDIX A

IEEE P2851 NEEDS

The following is the list of the main needs identified by the IEEE P2851 working group, with a description including the problem statement and the related IEEE P2851 WG proposal.

1. Safety Case DL
    - The safety case is a key work product created during the development of an item or an element and represents the final argument that functional safety can be achieved, and that the safety concept can be met. The safety case is compiled during the development of the project and uses the safety plan as input to determine the work products that will be created and will be used as part of the final argument. The IEEE P2851 WG believes that there are often missing or even inadequate safety arguments in the safety case and that this is a common issue in the industry due to a lack of examples and dedicated know-how. The IEEE P2851 WG plans to provide one or more examples that will emphasize the benefit of applying an existing description language together with some existing guidelines to a safety case for an integrated circuit or IP. A template/outline (DL) could also be proposed in addition to examples. It is not the intention of IEEE P2851 to create a new description language, as there are already several available today including Goal Structuring Notation (GSN), Claims Arguments Evidence (CAE), and Structured Assurance Case Metamodel (SACM). Also at least one auto industry guideline on how to construct a safety case argument already exists, as documented in Guidelines for Automotive Safety Arguments, MISRA.

2. Safety Requirements DL and ME
    - Safety requirements are an essential part of the description of the safety characteristics of an item or any relevant safety element. It is very important to define those safety requirements in a clear and complete way, which can be unambiguously understood by all levels and all fields of expertise. The IEEE P2851 WG believes that today the safety requirements are often not well described, and the incorrect application of requirements definition practices is an industry-wide issue. The IEEE P2851 WG plans to provide one or more examples that will emphasize the benefit of good requirements practice. The examples will align with the ISO 26262 as well as other relevant standards related to the use of natural, semi-formal, and formal languages. The IEEE P2851 WG also plans to provide a description language to allow for the description of specification insufficiencies as well as performance limitations related to SOTIF. The proposal will consider for the need for traceability of safety requirements between the different vertical levels in a bidirectional way. In addition, the IEEE P2851 WG will try to describe a possible methodology to demonstrate that safety requirements are complete and fully cover the needs of the previous level.

3. Safety Requirements Evaluation ME
    - Safety Requirements specify the safety critical functionality of an item, system, or element and need to be evaluated at different levels on whether they can be met as defined by the safety concept without fault (systematic fault) and in case of fault (random hardware fault). IEEE P2851 WG believes that there is a high risk that Safety Requirements are not evaluated in a complete and systematic way, which could result in a highly increased risk for the item. IEEE P2851 WG proposes to create a methodology that will describe how the evaluation of safety requirements at different levels of the V-Model could be performed in a complete and systematic way. The proposal could also include an example showing, on a specific case, how to demonstrate that the vertical traceability is complete. IEEE P2851 WG also proposes to define the tools needed to evaluate the safety requirements assigned to the item, system, or element under development.

4. Assumptions of Use (AoU) DL
    - Assumptions of Use are critical to show the context in which the safety critical item or element has been developed and they need to be properly communicated to the different levels in the V-Model as well as the different fields of expertise (hardware, software, …). AoUs are critical during the integration of the item in the vehicle or the integration of the elements in the system. IEEE P2851 WG members experienced that the Assumptions of Use are not always well written. It is not always clear which Assumptions of Use are key in the concept and why they are important. The impact of not fulfilling the Assumptions of Use as indicated in the documents needs to be better defined. There are also poor or missing links of the AoUs to, among others, the contexts, safety concept, use cases, boundaries, interfaces, and operating conditions. The proposal from the IEEE P2851 WG is to create a description language that would help to clarify the purpose of the AoUs, to identify the owner as well as to describe their importance for the safety of the element and the item so that communication between suppliers and their customers could be done more smoothly. The proposal may also include a description on how to implement and validate Assumptions of Use

requirements during the integration of an element. As last, some dedicated examples could also be added to emphasize the importance of Assumptions of Use in an IP and Integrated Circuit context.

5. Failure Mode DL

- During a safety analysis, the description of how an element will fail is very important in the industry. The need here is related to the creation of a description language that would allow to support the description of failure modes of elements and make those descriptions common over the industry and the different fields of experience. IEEE P2851 WG members believe that the way to write failure modes is not standardized in the industry and is, in many cases, poorly done, which creates issues to properly integrate elements in a system, item, or vehicle. A description language of failure modes containing the most relevant characteristics associated with the failure mode will be provided (example: expected Fault Handling Time interval, associated Assumptions of Use or the expected target safe state to be reached to mitigate the failure mode). Assigning those relevant characteristics should help creating an understanding of what needs to be verified related to the coverage of the failure mode. On top of the description language, the proposal could include dedicated examples. The examples could include some standardized list of failure mode terms or examples on how a failure mode can be translated across the levels of the hierarchy. The main goal would be to define a set of critical characteristics that would represent the failure mode and allow to take the failure mode into account into the higher-level safety analysis.

6. Failure Mode Effect DL

- The effect of a given failure mode on a design is critical but at the same time error prone as it may be difficult to understand by people not involved writing it. It is important that the impact on other elements and on the item itself can be correctly assessed and described in a systematic way. The effect of the failure mode is often too briefly written without a clear description of the context and how to link this effect with the other modules in the whole design. The proposal is here to define a description language that could help to improve the efficiency of this description. IEEE P2851 WG believes that creating a description language to define standard attributes that need to be considered when defining the effect of a failure mode (functionality impacted, ports impacted, etc.) would be beneficial. Possibly the description language could support the creation of a common justification method to link local effects from one sub-system or element to another one, which could, for example, help to improve reuse of the IP by connecting the effects of the failure modes inside of the IP with the external modules.

7. Functional Interface Analysis (FIA) ME

- This is the analysis of the interface between sub-systems (e.g., CAN, Ethernet, LIN), sensors or actuators to the controller. The impact of their failure modes is an important point of the safety analysis of a system. IEEE P2851 WG members experienced that the FTA/FMEA methodology used by the different development teams' part of a distributed system development focus too much on the sub-system they develop. There is often not enough focus made on the interface between all the sub-systems inside of the system resulting in an inaccurate analysis. It is also worth noting that the definition of signal failure modes between sub-systems is different than the failure modes from the original sub-system and that this analysis would make it possible to take the interaction into account. IEEE P2851 WG proposes to create a methodology for a Functional Interface Analysis (FIA). The methodology could define a way to specify the type of safety critical ports (observation ports, diagnostic ports, etc.).

8. Functional Safety Design DL

- The proposal of the IEEE P2851 WG is to propose a description language to describe the HW design under development from a safety perspective. The IEEE P2851 WG believes that safety-related information for hardware designs is not always described in a standardized way from a safety perspective, which makes interaction between different levels of the V-Model difficult and could result in systematic issues due to miscommunication. In other cases, the critical information related to functional safety management is sometimes not available at all for the design, which makes the integration at a high level very difficult. The IEEE P2851 WG proposes the creation of a description language that should standardize the description of the design with all the critical parameters and characteristics related to functional safety management. This could cover characteristics like the expected type of monitoring, expected safe state, expected fault handing time interval, and safety indicators.

9. HW Random Failures Evaluation ME

- A common way to evaluate random hardware failures is to use fault injection as proposed by the different functional safety management related standards. At some levels, like for example board design, the preferred method may be based on physical fault injection tests while at some other levels, like for example IC design, the main method will likely be fault injection simulation for permanent faults and transient faults. In some cases, a combination and comparison of physical random hardware tests and fault injection simulations may be the recommended solution. This need will describe the method to perform this kind of fault injection and how to combine and compare the results. The IEEE P2851 WG believes that fault injection can quickly become a complex topic and that there is a real risk to incorrectly perform fault injections, which will lead to an overly optimistic diagnostic coverage and wrongly considering the safe fraction when calculating the final metrics. Today, there is no standardized method on how to perform fault injection and how to apply the raw FIT to all classified faults. The IEEE P2851 WG believes that performing fault injection on digital modules, analog modules, or a PCB design need to be performed in their own way and that the specificities and correct defect models are often not fully considered leading to incorrect results. The IEEE P2851 WG proposes the creation of a methodology that will cover fault injection on the different hardware modules ranging from IC development to system development. The methodology will focus on fault injection in general by considering physical fault injection tests as well as fault injection simulations. The proposal will also consider tool confidence level and tool qualification of for those critical tools related to both physical HW fault injection and fault injection simulation.

10. Vulnerability Factors modeling ME

- In devices processed in technologies that are susceptible to soft errors (i.e., high-energy and thermal neutrons or alpha particles), it is important to understand how vulnerable the devices is to estimate this susceptibility, it may be critical to perform a vulnerability analysis that takes into different vulnerability factors which are function of, among others, the architecture, the workload, and also the clocking strategy. The IEEE P2851 WG believes that the industry would be benefit from a standardized methodology to estimate the vulnerability factors of hardware to soft errors. The IEEE P2851 WG proposes to define a methodology to model the vulnerability of a component considering vulnerability factors and the relationship to safety parameters such as F_{safe}. The vulnerability factors that will be considered are: AVF = Architectural Vulnerability Factor, TVF = Time Vulnerability Factor and PVF = Program Vulnerability Factor. The IEEE P2851 WG also believes that it would be beneficial to define how the vulnerability factors can be applied at each level.

11. Base Failure Rate (BFR) DL

- When calculating the base failure rate of a component, several aspects need to be assumed or considered. Those aspects can have an impact on the final value and need to be carefully communicated to a higher-level integrator. The IEEE P2851 WG believes that data for the calculation of the base failure rate is often not correctly transmitted between the system integrator and the component supplier, which may result in incorrect metrics at the system level. The IEEE P2851 WG proposes the creation of a description language that will cover the board level as well as the silicon components and will consider all the relevant aspects with among others: Mission profile/ power dissipation/air flow/self-heating, Confidence Level, Complexity of the mission profile vs. different use cases, Dependency/interoperability between use case domains, Device technology type (SRAM, Digital, etc.), Package type etc.

12. HW elements evaluation ME

- Hardware elements need to be evaluated for compliance with the safety concept and relevant standards. The methodology that companies use to evaluate hardware elements is not standardized and varies in the industry with the possibly of several weaknesses. This can result in many discussions and frustration from the hardware suppliers as well as the system integrators as they need to understand, assess, and adapt to different methodologies depending on the companies with whom they work. The IEEE P2851 WG proposes to standardize and homogenize the evaluation methodology for hardware elements.

13. Safety Manual DL

- The Safety Manual is an important document that helps to describe how a component or sub-system needs to be integrated in a higher level in line with the expectations and assumptions of the component or sub-system manufacturer. The IEEE P2851 WG members note that it is not always clear whether we provide enough or too much information and whether the way requirements are written is easily understandable by other vertical levels. For example, it is not always easy to understand from existing safety manuals whether the provided data/requirements are mandatory (including safety metrics) or are simply good to have and can be avoided. The proposal is to create a standardized description language through a template with a fixed format containing the essential information that needs to be considered for the creation of safety manuals in the industry. The IEEE P2851 WG proposes to also define the minimal information that is most critical for the vertical levels and how to best represent that information in a consistent format.

14. Safety diagnostic information DL

- In advanced self-driving systems, there will be a need to perform in-field monitoring and diagnostic measurements to understand the health level of the system and report potential future issues so that maintenance can be performed before a potential safety critical failure could occur. The IEEE P2851 WG believes that this will be an important need in the future and that waiting for a system to fail will not be acceptable. The IEEE P2851 WG believes that there is a need to develop a consistent description language or template to define the kind of data, for example, the parameters that need to be measured and reported to control the health state of the system and detect potential safety weaknesses. The scope here includes in-field monitoring and diagnostics techniques such as cache error reporting and other telemetry for safety data reporting.

15. Define criteria and project-specific tailoring activities ME/DL

- Safety activities are commonly tailored in the safety plan at the beginning of a project. The tailoring is dependent on the safety concept, the specificities that could arise during the development of the project or the selected methodology. The IEEE P2851 WG believes that, although allowed, tailoring is an important point that needs to be well communicated between the different vertical levels to avoid miscommunication. The IEEE P2851 WG believes that it would be beneficial to tailor the safety plan in a common and consistent way by defining a common methodology and description language.

16. Stochastic behavior analysis ME

- Stochastic generally implies that uncertainty of outcomes is quantified in terms of probabilities. Stochastic behavior needs to be analyzed for usability in safety critical systems and assessed for meeting the expectations from a safety point of view. The IEEE P2851 WG believes that there is no real methodology available today that could support the analysis of stochastic behaviors and demonstrate that an AI development is fit to be used in safety critical systems. The IEEE P2851 WG believes that stochastic behavior is an AI unique behavior that needs to be analyzed and understood and a common methodology needs to be defined.

17. AI System Safety Performance Indicator ME

- The IEEE P2851 WG believes that safety performance indicators can be used to evaluate the AI performance of an implementation and indicate whether it meets the best practices. The IEEE P2851 WG concluded that training models may not always have the expected performance and that it is very difficult to understand from outside whether a trained model is reaching the expected performance and meets best practices in this field without any performance indicator. There is no standardized methodology today to define such safety performance indicators. The IEEE P2851 WG believes that there is a need to define a methodology to build a diverse verification data set to help ensure safety performance indicators are met on a system with a trained model.

18. Cybersecurity SW testing ME

- Cybersecurity SW testing is important to understand weaknesses of SW code. The SW testing applied to SW code related to Cybersecurity could be used to understand whether the weakness also has an impact on the functional safety of the car. The teams responsible for the functional safety analysis of the car or elements from the car do not always know the benefits of cybersecurity-related SW testing techniques while those techniques may be useful for them as well. The proposed approach is to examine current techniques for cybersecurity SW testing and define the methodology to consider these in the FuSa SW test flow in order to assess if the threat has impact on safety (e.g., FUZZ testing).

19. Safety assessment for SW updates ME

- Security and safety, among others, will require continuous updates of firmware and embedded software in the future, in order to cope with safety weaknesses and potential security threats. Software and firmware updates are often very difficult to assess for impact on the safety goals. Those updates, when not fully validated and approved, present a risk to have an unacceptable impact on the safety goals of the item. The IEEE P2851 WG believes that there is a need for a methodology to determine the effect and impact quickly and efficiently on the safety goals and proposes to create such a methodology.

20. SAD (SW Architectural Design) ME/DL

- The software architectural design represents the architecture of the software, its elements, and their interactions. The SAD also allocates the input requirements to the different elements composing the software. The SAD is a very important document as it has a direct impact on the rest of the development activities. Incomplete, ambiguous, or incorrect descriptions could lead to potential systematic issues. Current standards provide some hints on the content of the SAD; however, detailed guidelines and methods are missing. Having common methods to describe the SAD would ease the interface between tools and stakeholders. It would be helpful to provide a guideline for the SAD. Examples of tools or diagrams (UML) that could be used to describe the static and dynamic aspects of the SW are: what should be described, which tools or methods should be used, and whether there can be a common language between SW and System Architectural Language, e.g., SYSML.

21. Systematic Analysis ME

- The goal is to provide a methodology for systematic analysis of components per ISO 26262 requirements. The IEEE P2851 WG proposes BKM/Methods to analyze HW and SW components in order to achieve compliance with systematic requirements per ISO 26262.

22. Concept DL

- Concept is the first phase of the IEC 61508 safety lifecycle. The goal is to provide sufficient description of the requirements applicable to the product and its environment (physical, applicable standard and regulation, etc.) to enable the other safety lifecycle activities to be properly executed. The issue is the lack of standardization for this activity leading to an inconsistent description of the requirements and environment applicable to the product. The IEEE P2851 WG members propose providing a high-level description of the safety element being developed including environmental description, assumptions of use, high level functional description, list of potential sources of hazard and effects description (including interaction with other elements). This should include the list of applicable statutory or regulatory requirements. It would be good to define a common list of parameters to be considered or eventually a common template.

23. Risk Level/ASIL alignment Matrix ME

- The Risk Level/ASIL alignment Matrix is a work product to ensure alignment between functional safety and cybersecurity during development. The IEEE P2851 WG members believes an alignment Matrix is needed to set up actions for security requirements with safety implications. The IEEE P2851 WG plans to provide one or more examples of showing how such an alignment matrix maybe created, and the necessary steps for ensuring safety consideration are given to the security implementation.

24. AI training ME

- AI-based elements need to be trained with massive data to achieve their intended functionality such as audio recognition, object detection, object classification, etc. Considering AI training is a special type of software configuration and calibration, such a methodology should be added. It must include requirements/considerations of how to build a sufficient training data set. In alignment with the direction given by ISO/DIS 21448, AI training could be considered as a special type of software configuration and calibration. IEEE P2851 can help by providing a methodology and examples about how to apply ISO 26262-6 to such a specific context.

25. Safety Goal DL

- Safety goal is top level safety requirement because of the hazard analysis and risk assessment at the vehicle level. The biggest issue seen in scenarios where SGs need to be exchanged is that they may not be available, or they may not meet ISO 26262 requirements. There are already DLs for writing requirements; however, they impose semi-formal to formal notation constraints. One direction is to agree that safety goals should be common to functional safety and SOTIF. Therefore, it may be possible to derive a high-level common safety goal, with sub goals incorporating functional safety specific information, such as failure rate.

26. HARA ME and DL

- This refers to the methodology and description language to identify and to classify the hazardous events and formulate safety goals. There are considerations and methods described in ISO 26262 and SAE J2980 for ASIL hazard classification. The IEEE P2851 WG should also review and consider both standards and other relevant methods including the application of STPA (Systems Theoretic Process Analysis) to help ensure no duplication and should provide examples.

27. Safety Verification DL/ME and Attack Tree Analysis (ATA) for SG violation

- Safety verification determines whether an examined object meets its specified requirements. There can be significant concerns with design verification at the HW, SW, and System level. Teams can get too focused on specifics of tests and not have a good understanding of what they truly need to verify and whether the verification approaches selected are adequate including their verification plan and specification. Any DL would need to be careful not to compound this trend. Any work in this area needs to allow for the reality that there is not one correct set of verification evidence, i.e., there are multiple wants to have a compliant verification argument. ME is helpful to define high level considerations (and scope) for fault injection testing across levels based on the requirements. High level fault injection methodology/steps for simulation and measurements are desired, including augmenting the training dataset with various technology, training set modification techniques in AI/ML applications. The goal is to define a methodology to build a diverse verification data set to help ensure that the key performance indicator and safety performance indicators are met on the trained model. Targets could be achieved via standard verification testbenches, or via point tool solutions that do fault injection etc. The intention is to create a DL/ME for safety verification to ensure adequate considerations are given. The main focus areas are: 1). The evaluation of safety mechanism diagnostic coverage with sufficient test coverage; 2). Assessment of whether the safety mechanism can detect related faults, transform the system into a safe state, and recover from the safe state within given timing requirements; 3) AI/ML related verification.

28. Safety Mechanism DL and Safety Envelope DL

- The scope is both HW and SW based safety mechanisms, including requirements for implementation and HSI. Safety mechanisms benefit from being described in a formal way with different characteristics: detection time, reaction time, the way an issue is flagged, the entered safe state, diagnostic coverage, whether it can be used to detect SPF or LF, etc. This makes the safety analysis and confirmation reviews easier and hence helps to decrease the number of issues during the safety analysis. Intent is a template and syntax to capture SM criteria. It needs to be consistent with ISO 26262.

29. FMEA/FMEDA/DFMEA ME/DL

- Each of these mentioned analyses are well described. The focus here is on creating consistency among different analysis. Also, there are always specific cases that are not as clear and that deserve some attention: SR/NSR module identification, properly indicating the DFA and FMEDA linkage, making the FMEDA adaptable to different safety concepts (different external conditions, safety mechanisms, different safe states, different accuracy figures). While "how to do" and FMEDA is generally understood, the "when to do," "why to do," and the relationship with other analyses is not that well understood. Some issues are as follows: lack of understanding of the relationship between the APQP DFMEA and ISO FMEA and how they can be efficiently integrated into a single lifecycle; jumping straight to the FMEDA without thinking about whether this is a good strategy; executing the analysis post design vs. having the analysis drive the design. The proposal is intended to also include guidance on how to decide on safe faults, transients, etc. and should align with DFMEA handbooks. IEEE P2851's value would be to provide examples for IC and IP and to consider SW FMEA consistency. The intention is to also create examples to reflect AI-related applications and to also add security threats and security measures effects on design safety.

30. FTA ME

- FTA (Fault Tree Analysis) is a top-down, deductive failure analysis in which an undesired state of a system is analyzed. There is a need for a consistent methodology/Best Known Method across domains. There is no formally documented methodology for FTA, but it would make sense to define one that would complement the FMEDA and allow for the information to be reused in the effect analysis and effect justification. The FTA methodology is generally well understood by OEMs and T1s; it is just not used widely by IP and IC vendors. A bigger issue is the lack of understanding in the IP/IC industry of the potential benefits of an FTA and when it is beneficial to apply it. It is true that the method does not have the equivalent to the FMEA blue book. Equivalent examples to that which exist for an FMEDA in ISO would help. A contribution would be to also provide education via examples of how to apply analysis efficiently to IP and IC and where they bring real benefit/intent.

31. AI Safety Validation DL/ME

- Safety validation is an assurance, based on examinations and tests, that safety goals are adequate and have been achieved with a sufficient level of integrity. In the presence of AI workloads, it is not clear as to what should be examined or tested. The IEEE P2851 WG plans to provide a reference list of relevant safety performance indicators, key performance indicators, and safety requirements that should be met, in the presence of AI workloads, in addition to providing examples.

32. DFA ME/DL

- DFA (Dependent Failure Analysis) aims to identify failures (either common cause or cascading failures) that invalid the required independence or freedom from interference between given elements/components. Consistency is needed in how the outcome of the DFA, assumptions etc. are captured, while considering unwanted interferences on function/item/system level with regard to SOTIF aspects. This includes unintended function/item/system interaction DL, ME, and DB for translating a security threat/security remediation effect to DFA. It also Includes a template/structure/framework for work products, including the DFA assessment. The goal is to provide examples, including SW in scope, and to consider DFA in the context of FTA.

33. Configuration and Calibration DL and AI data management ME

- AI-based elements are widely used in automotive, e.g., for object detection using Convolutional Neural Networks (CNNs), object classification etc. AI systems are trained, not programmed in a "classical" way. Training of AI elements require complex systems for data management. Many companies experience problems with proper implementation of data harvest, management, and data quality control. The proposal is to define a ME for AI data management and a DL for calibration and configuration data (which may include NN weights).

34. SW components qualification ME

- Third-party SW components, including AI-based SW (NNs) are widely used in automotive applications on different levels. Currently only the methodology for qualification of embedded SW without AI-related elements is standardized. The proposal is to define a comprehensive ME for qualification of SW components.

35. ASIL decomposition ME

- ASIL decomposition is a powerful tool to get credit for redundant architectures while remaining flexible (i.e., not prescribing concrete architectures). It is not always clear how to properly document the decomposition and how far one needs to go to be independent enough from an implementation point of view. When performing decomposition, the flow for systematic issues can be reduced; however, there are aspects that are considered as a minimum requirement (even for QM parts) in order to not have availability issues. There is some confusion on the confirmation measures for the decomposed elements, so this methodology could help to clarify this aspect by defining a ME for ASIL decomposition considering minimum requirements for the decomposed elements.

36. Identification, impact, and resolution of safety anomalies ME

- Safety anomalies are potentially hazardous incidents as well as accidents observed after the development safety lifecycle has ended (i.e., when the system has been validated and considered reasonably safe). Safety anomalies' resolution process is a requirement by quality and safety standards. However, it differs between industries. Many important standards (e.g., ISO 26262) require such a process but do not specify it in a precise way. The proposal is to define a ME for safety anomalies identification, impact, and resolution, which can be applied in various industries on many levels.

37. Item/Item Description DL

- Item is a functional system on the level of the vehicle/equipment under control. Item description includes a description of functions, interdependencies with other items, and external mechanisms (incl. mechanical ones). There is no standardized way to described functions, interdependencies, and external safety mechanisms. The proposal is to define a DL for Item Description that can be used across domains.

38. Operational/driving situations and operating modes DL/ME

- Safety requirements need to be adhered to for every operational/driving situation (scenario) in the operational design domain (ODD). Complex functionality like automated driving requires definitions of hundreds of driving situations (scenarios). It is impossible to solve this problem without the automation of a scenario definition. The proposal is to define a DL and ME for definition of operational situations allowing automation of the following use cases: simulation of the scenario, description of the driving situation in DL, concretization of the scenario, and randomized testing. Chip vendor and OEM perspectives to be considered.

39. System architectural design DL

    - Description of system architecture includes system composition and system behavior. Many descriptions have been proposed for system architectural design. One of the most widely known are UML and SysML by Object Modelling Group (OMG). Their intention are system-level descriptions. The proposal here is to define a DL for system architecture design that can be used on multiple levels.

40. SW design/algorithm impact analysis ME

    - Impact analysis allows tailoring of the engineering lifecycle based on the idea of reuse of the work products that retain their actuality. Currently, only goals and the scope of the impact analysis are described in ISO 26262. However, using known interdependencies between lifecycle artifacts, it is possible to standardize and automate it. The proposal is to define a ME for impact analysis allowing automation (at least partially).

41. Development Interface Agreement (DIA) DL

    - DIA is an interface between business and technical processes in a company. DIA is different for every customer and can be very short or very long with reference to the ISO 26262 standard requirements, but it is not always easy to find important information and identify special requests. Reviewing DIA files is sometimes tedious and requires detailed discussion to homogenize the needs compared to the safety plan. The proposal is to define a description language to link the DIA with the safety plan in a more direct way.

42. Confidence in use of SW tools evaluation ME, DL

    - The proposal is to define a methodology and a description language for expressing the confidence in use of software tools used in development of a system or its software or hardware elements. Developing a unified flow with the CAD department is not easy but is key to implementing a robust flow, which allows using the TCL documents from the EDA vendors or from internal tools, to describe reference robust flows with sufficient tool coverage and add regression tests when appropriate. For example, fault injection tools should provide reliable and repeatable results, and the tools and flow should be clear and easy to implement. The proposed DL should define all required parameters in a TCL description data element for each SW tool in use and define parameters required for describing the TCL of a tool chain. The proposed ME should define a methodological approach to handle all TCL documents from EDA vendors or internal tools and testing.

43. Criteria for coexistence of elements ME

    - The proposal is to define a methodology for defining the criteria for coexistence of safety-related sub elements with non-safety-related sub elements within the same element, or safety-related sub-elements with different ASILs assigned within the same element. This need is expected to define a methodology to show that an element implementing requirements with lower ASIL cannot impact an element with requirements with a higher ASIL. It would also show that an element implementing non-safety-related requirements cannot impact an element with safety-related requirements. Many complex systems or SoCs are developed with multiple applications in mind, some are safety critical and some are not (SR vs. NSR) or may have varying degrees of required ASIL (e.g., infotainment vs. AD tasks). ISO 26262:2018-9, section 6 provides details on the methodological approach to this need; however, the standard seems to lack details on how to approach the architecture, what kind of capabilities are needed for isolation, partitioning, etc. and how to consider the stack/layers of the application. A DFA analysis may serve as a starting point, and we need to determine what kind of methodology would need to be developed in addition to that analysis. ME – Define rules for coexistence of sub-elements of different Safety Integrity Level in one element by presenting the mitigation for each potential effect; start from DFA.

44. Production offline verification of safety mechanisms ME

    - The IEEE P2851 WG proposes a methodology for the verification of the correct behavior of the safety mechanisms during production. At production, as part of the element acceptance tests, it is desired to add testing of the Safety Mechanisms. A special consideration for this test needs to be applied during the element design, since Safety Mechanisms do not assert an error or notification during normal operation of the element. From an element integrator point of view, sub elements with SMs need to come with a clear verification description for those SMs.

45. After sales verification of safety mechanisms ME

    - This need is very similar to item 43; however, the testing focus is on the next level integrator of the element.

46. Impact assessment vs. requirements in case of reuse of element(s) ME, DL

    - The IEEE P2851 WG proposes a DL for describing the required activities vs. reuse of activities of derivative elements. When developing derivative elements, it is important to be able to limit the amount of work in case the element has a big portion of reuse. This need is very close to the project tailoring need as some steps will be tailored out as common with the previous project. Note that the focus here is on the changes needed for an already compliant element. It would become the safety plan for the enhanced product. The proposed DL would provide a description language in which the activities tailoring due to reuse can be standardized. The proposed ME would provide a methodology to keep track of this reuse activity, and provide example use cases.

47. AI models update ME

- This is a proposal for a ME for describing malfunction or incorrect behavior due to malicious attack that results in a safety goal violation. AI-based AV solutions need continuous improvement. When performing a road test, the AI model will likely meet a lot of corner cases, which may require the vendor to trim it down to mitigate such risk. The trimmed AI model should not create new limitations. During OTA upload, the model should be safely uploaded to the car system. A ME is needed for standardized testing of AI models updates before OTA upload.

48. Threat and Risk DL

- This is a proposal for a DL for describing malfunction or incorrect behavior due to a malicious attack that results in a safety goal violation. When considering security threats and attacks during TARA, one of the implications may be a safety goal violation. As part of the Safety-Security alignment, these threats need to be communicated to the Safety team for further safety analysis and mitigation. The proposed DL would provide a description language for describing security threats and risks that impact safety.

49. Fault model for side channel threats (HW)/Fault extraction/injection ME

- This is a proposal for a methodological approach to Fault Modeling and Fault Injection modeling side channel threats. Side channel attacks such a clock or voltage glitching, overheating, and laser beams or ion beams are well known types of physical security attacks. These attacks may lead to permanent damage to the system (permanent faults) or to temporary value change in gates and registers (SEUs, transient faults). Fault Injection during SW execution may interfere with the correct execution of instructions and derail the execution to some implanted malware. It is important to address the fault models of those security attacks as part of the design verification, and to come up with the proper fault injection methodology to effectively simulate these attacks. The proposal is for IEEE P2851 to come up with fault models and a methodology for fault injection simulation tools and flows for side channel attacks.

50. Best Practice for System Integration and Testing, ME

- A methodology for best practices for system integration and testing for the intended functionality. The SOTIF standard (ISO/DIS 21448) describes the need for a system integration and testing plan, and lists methods for verification of the intended functionality. The recommendations given in ISO/DIS 21448 are, however, quite generic. It seems that examples and list of best practices would be helpful in this context—a list of best practices for integrations and system testing of the safety of the intended functionality.

51. STPA ME

- A methodology STPA (System Theoretic Process Analysis) is a new technique for hazard analysis, considering unsafe interactions between components (including humans) as well as component failures and faults. Like other functional safety analyses, a well-defined methodological approach to correctly perform the STPA is needed. The proposed ME would describe how to approach STPA and include examples of STPA. Since a STPA handbook is available, it is suggested that the IEEE P2851 WG will review the handbook and decide whether to keep as-is or add supplemental examples.

52. Confirmation Measures DL

- This is a proposal for a description language to help in writing the results of the confirmation measures. ISO 26262 v2 (2018) tried to clarify that the critical element of compliance is to meet intent (aka objective) vs. getting hung up on specific words used in clauses. The confirmation measures should always focus on whether intent was met as the primary consideration. Any DL should not detract from this. The proposal is to build a framework for all confirmation measures focusing on whether the intent was met as the primary consideration.

53. Safety Plan/Security Plan DL

- This is a proposal to define a DL to describe a template for the plan to execute safety/security activities. Different players have different understandings of the needed safety plan content and description of the requirements from the ISO standard. Hence a unified DL is needed. In addition to the DL, it is also important to understand the structure of the Safety Plan and the sections in it. A structural template for the content of Safety/Security plan, with examples of types of plans, activity tailoring, and rationales for tailoring will be helpful. The proposal is to define a database with all the safety plan requirements extracted.

54. Risk Priority Number (RPN) ME

- This is a proposal for a methodology and examples for setting the RPN metric. RPN (Risk Priority Number) a well-known method for assessing the combination of Severity/Occurrence/Detection of a failure in HW. Moreover, it is being replaced by many users with AIAG&VDA, which define the AP (Action Priority) for FMEA. Nonetheless, RPN is still in use, and apparently there is a need for it in other domains, as well as in alignment of Safety and Security requirements. The proposed ME would provide guidelines on the metric that leads to a RPN value in different domains in a way that enables comparison of the priority of requirements from those different domains.

55. Analysis of functional insufficiency of the intended functionality ME, DL

- This is a proposal for a methodology and a description language for analyzing functional insufficiency of the intended functionality. ISO/DIS 21448 (SOTIF) addresses the functional insufficiencies of the intended functionality, either SW specifications insufficiency, or performance limitations, as the main causes of hazards that may lead to unreasonable risk. ISO 21448 proposes methods for validating the intended functionality and to uncover insufficiencies. A standardized way of defining these insufficiencies it needed. The proposal is to provide a methodology and a DL for analysis that is performed on the function spec at item/system level, uncovering any insufficiencies in the specifications. And to also provide a methodology and a DL for analysis for function validation failures that are due to performance insufficiencies.

56. SW development environment DL

- This is a proposal for a DL for the SW development environment, including development model, languages, and tools. ISO 26262 part 6 specifies the requirements for product development at the software level for automotive applications. However, a clear description of the SW development environment, in a templatized form, is called for to satisfy the ISO 26262 requirements on SW in a uniform way. The proposal is to create a DL for describing all the part of SW development environment, and to consider some templates developed by companies willing to donate their templates to the IEEE P2851 committee work.

57. Collection, inferring, and standardization of failures sources ME

- This is a proposal to provide a unified description of failure sources. There are several standards that provide reliability prediction considering failures resulting from development or manufacturing errors, overstress, underlying technology, use conditions etc. Specifically, the need is to review IEC 63142 and JESD89A to determine if there are gaps/limitations that need to be covered here. The plan is to collect all widely used reliability standards and come up with a methodological approach to define reliability data prediction and Technology Failure Rates.

58. Proven in-use evaluation ME

- This is a proposal for a reference methodology to clarify the way to assess a “safe” proven in-use methodology. A proven in-use argument is an alternate means of compliance with the ISO 26262 series of standards that may be used in case of reuse of existing items or elements and field data is available. The IEEE P2851 WG believes that proven in use is a difficult topic for IP re-use for example as the conditions of use of the component are always somehow different and correlation between “uses” of the IP are not easy to prove. Also proven in-use is often confused with qualification data that is not the purpose. Proven in-use methodology is mainly applicable to hardware sub-systems and elements and to their integration at a higher level.

59. DB of Dependent Failure Initiators (DFI)

- This DB would contain descriptions of events that may serve as initiators for dependent failures or common cause/common-mode failures. The completeness of the Dependent Failure Analysis depends on the completeness of the list of initiators. Therefore, it makes sense to have a single and shared collection of initiators. Currently the information on initiators is dispersed between standards and reference manuals. The proposal is to create a database containing a description of events that may serve as initiators for dependent failures or common cause/common-mode failures.

60. DB of HW/SW/System triggering conditions (SOTIF)

- A triggering condition is a condition external to the item that can possibly lead to a hazardous behavior (cf. ISO/DIS 21448). Triggering conditions are similar between applications within one target environment (automotive, avionics, etc.) ISO/DIS 21448 lists classes of triggering conditions to be considered in the analysis. The proposal is to create a database for triggering conditions to ensure the common level of details in safety analysis. The list of triggering conditions will need to include references to the performance limitations they cause.

61. DB of system-level permitted interface behaviors

- ADAS/AD systems interface the vehicle systems (braking, steering, etc.) As the latter are highly standardized in their functionality, the interfaces can be standardized/described in a database, too. Parameters like FTTI, precision, and delay influence the safety of the interfaces. The proposal is to create a list of parameters influencing the safety of interface behaviors and to list malfunctions that may result from faulty interface behavior, as well as their potential consequences and hazards.

62. DB of effects and SW real-time constraints

- AD/ADAS systems perform functions that can be largely standardized (AEB, LKA, LCA, etc.). As far as the functional description remains same on the vehicle level, it is possible to define potential effects of malfunctions of this function. The proposal is to provide a complete and clear description of the effects of the malfunctions related to a function, which is key to completeness and correctness of the HARA. Also, a list of the effects of potential malfunctions should be created. For each item of the list, parameters like controllability and FTTI would be considered.

63. DB of AI system safety performance indicator

- To assess AI-based applications, specific methods and procedures are required. Common test sets/stimuli databases are widely used tool to assess the capability of AI-based systems. A standardized test set/stimuli database may help ensure that all systems are measured against one benchmark and fulfill the same level of safety. The results of application of this DB may serve as a safety performance indicator. A domain-specific stimuli database for AI-based item/system/HW and SW component characterization (real-time performance, accuracy).

64. DB of failure rates

- Base failure rate estimation is needed for the safety analysis and can be computed using methods described in various industry standards. The IEEE P2851 WG members believe that there are several methods that can be used to calculate the base failure rate and the complete list of available standards for this estimation is not currently documented for reference. IEEE P2851 plans to provide all relevant base failure rates methods including former IEC/TR 62380, SN29500, etc. This could be interesting for some types of devices.

65. DB of failure modes including foreseeable misuse and known specification gaps

- Largely standardized functions call for the standardization of failure modes. Completeness and correctness of the failure modes under analysis play key role for the completeness and correctness of the safety analysis. The database includes fault models and the resulting manifested failure modes. It covers analog and mixed signal devices and digital components. It includes known cases together with statistics (probabilities).

66. DB of operational situations and modes, performance metrics, and dynamic conditions

- Operational situations are similar within one target environment (i.e., one list for automotive, one list for aviation, etc.). Analysis of operational situations is an activity residing on a higher level than item/system analysis. It requires knowledge of the intended environment, which is not necessarily available at the system manufacturer. The database should support categorizations (i.e., selection of a category of roads). The database should support a coverage calculation using the algorithm provided by the user. The database describing dynamic conditions logically should support time scaling.

67. DB of S, E, C

- This DB is a reference DB for the DBs of failure modes and effects.

68. DB of AI training data

- Standardized operational situations call for standardization of scenes and scenarios, i.e., data used for AI training. The application of best practices (e.g., coding and design guidelines) does not ensure the safety of AI-based applications to the degree it does for “classic” programmable SW. A sufficient training set for AI needs to be considered with respect to its intended applications according to industry’s best practice.

69. DB of failures sources

- The base failure rate estimation is computed using prediction models that are accelerated by sources that induce the occurrence of failure mechanisms. The IEEE P2851 WG members believe that there are several sources of failure that lead to the manifestation of failure mechanisms and resulting base failure rate and the complete list of failures sources for this estimation is not currently documented for reference. IEEE P2851 plans to provide all relevant failures sources such as radiation sources for single event effects, etc.

70. DB of assumptions for HW metrics

- The HW metrics are calculated based on certain assumptions that are included in the safety analysis. The IEEE P2851 WG members believe that the assumptions on which the safety analysis is based need to be defined and documented to prevent hazards due to wrong assumptions. As such, there is not an exhaustive list of various AoU and other assumptions that are documented for reference. IEEE P2851 plans to provide a list of all relevant AoU (safety mechanisms from external systems), as well as additional assumptions such as the failure rate distribution and definition for different failure modes, protection electric circuits (e.g., ESD and soft errors).

71. DB of safety mechanisms

- Despite huge diversity in system architectures and technical solutions, there are safety mechanisms that are used universally. Different implementation of the safety mechanisms aimed at similar functionality leads to difficulties in their validation and assessment, potentially creating unreasonable risk. Besides that, commonly known limitations of those safety mechanisms often go unchecked (e.g., non-systematic use of parity bits, CRCs, and checksums for safe communication). A list of typical safety mechanisms on different levels for different units (digital/analog/SW). The list needs to include known limitations of the safety mechanisms.

72. DB of hazards and risks

- A hazard and risk assessment is completed in order to assess the target safety goals for the system, taking into account various scenarios that could trigger a hazardous situation. The IEEE P2851 WG members believe that a list of potential hazards and risks for various scenarios should be documented for reference. Such a list/database does not currently exist. IEEE P2851 plans to provide a list of all relevant hazards and risks that should be considered in the safety assessment, across domains.

73. DB of external measures

- Similarly to internal safety mechanisms, there are universally used external safety measures. Analysis of external measures is an activity residing on a higher level than item/system analysis. It requires knowledge of the infrastructure, which is not necessarily available at the system manufacturer. A list of typical external safety measures for different systems. The list needs to include known limitations of the measure.

74. DB of use environment

- This DB is a reference DB for the DB of operational situations.

75. DB of SW tools safety evaluation benchmarks

- SW tools for engineering and design (CAE, CAD) perform safety-relevant tasks. Covering their safety is a part of system safety lifecycle. Information related to the tools is available mainly from tool vendors. It makes tool comparison harder. Potential safety violations are only found when the tool is implemented, thus creating both business and safety risks. A list of SW tools used for safety-relevant tasks. For each tool, potential safety-related failures and measures to counteract them are provided.

76. DB of criteria for tailoring

- A project specific tailoring is done to determine relevant functional safety activities for the system under consideration. The IEEE P2851 WG members believe that the complete list of criteria and considerations for tailoring of the functional safety activities is not currently documented for reference. IEEE P2851 plans to provide a list of the relevant criteria in the form of a database. However, this will be somewhat dependent on the type of components and it may be difficult to have a generalized list.

77. Metrics definition DL

- Multiple metrics can be measured during to the development of the software product to describe its characteristics including performance, quality, complexity, etc. In current standards examples of metrics are provided but not always in an exhaustive way. When metrics examples are provided often the corresponding target values are not defined. A few standards provide a list of metrics and targets, but there is no consensus on which one to use. To help ensure a consistent software quality level across different organizations, it would be helpful to define the list of metrics to be measured and for each of them a target value. List of metrics used to determine the quality of the SW code could be provided as well as the corresponding targets e.g., cyclomatic complexity target value.

78. System/Life Profile DL

- Life or Mission profile is a representation of the stress seen by the product over its useful lifetime. This is important information that could have direct impact not only on the reliability of the product but on the safety analysis as well. Life/Mission profile descriptions for the safety element being developed are not always complete and consistently defined between the different stakeholders. The proposal is to provide a description of the mission profile including the type of stress to be considered and to provide examples of life/mission profile per type of applications.

79. Models ME/DL

- Models are used to describe and verify a product's functionalities at a high abstraction level. This allows, at an early stage, the behavioral verification of the product and facilitates the design activities. Multiple languages and tools exist that can be used to write models that make data exchange between stakeholders difficult or impossible. The proposal is to provide a list of tools and languages available to write models, and to provide a list of models' parameters that could be used to ease interoperability between tools and exchange between stakeholders.

80. Coding guideline/Code review DL

- Coding guidelines provide recommendations on the use of a programming language in order to avoid systematic issues or weaknesses in the code. Currently, there are different programming languages subsets with multiples rules and recommendations that are recommended by safety standards. However, there are no requirements for the tailoring of the rules themselves, which is left to the development team. It would be useful to provide a common coding guideline for safety development, and to define the criteria for selecting the coding guidelines as well as the criteria for tailoring the rules.

81. Failure modes for SW DB

- This is a proposal for the failure modes for software elements used during the safety analysis of the software product. Unlike for hardware, the failure modes for software elements are not correctly described in current standards potentially leading to quality issues or/and inconsistencies between different safety analysis. The proposal is to provide a database including the list of SW failure modes that could be used in the software safety analysis (e.g., SW scope of DFMEA analysis).

82. Life profile examples DB

- The life profile contains detailed information on environmental and operational aspects to be encountered by the product. This can include various type of stress including thermal, electrical, mechanical, chemical, humidity, etc., their levels, duration, and sequence. The life profile is a representation of the stresses seen by the product during its lifetime and has a direct impact on the failure rate of the device. Therefore, during a safety analysis, the life profile plays an important role as any wrong or incomplete information could lead to wrong analysis. Today it is rather difficult to find a list of typical life profiles for specific applications, resulting in incorrect assumptions by stakeholders. The proposal is to create a database of life profiles for different automotive applications.

83. ME for criteria for Common-Mode Analysis (subsystem level)

- The CMA (Common-Mode Analysis) is a methodology required by SAE ARP4761 in order to verify that in case of a failure that is the consequence of two independent events (hardware fault, software fault, abnormal condition, etc.) the events at the origin of the failure are truly independent. The methodology is applied to ANDed events of Fault Tree Analysis, Dependence Diagrams, Markov Analysis. SAE ARP4761 Appendix K gives general guidance on how to perform the CMA. But it lacks criteria and methodology to determine for each identified combination of event to be tested the fail/pass criteria. IEEE P2851 plans to provide some guidance about how to perform the CMA.

84. Single Event Effects models and failure modes DL

- Single Event Effects (SEE) faults/errors/failures are erroneous states observed at unit/component/system/item level. They are caused by an energetic particle and can take several forms. There is a need to express Single Event Effects (SEE) faults/errors/failures in a consistent format and language. This includes not only low-level faults (such as Single Event Upset/Transient) but also have to cover system-level failure modeling. IEEE P2851 plans to give a detailed lists and definitions of SEE from low to top level. A review of the available State-of-the-Art and the literature will allow us to prepare a good overview of the available SEE modeling approaches and methodologies. A reasonable number of representative methods will be selected.

85. Formal methods for the validation of vulnerability factors ME

- Electronic Design Automation (EDA) tools will most likely produce most of the data and information presented in any analysis and reports. Many of this data capture in a number or metric increasingly sophisticated methodologies applied on complex circuits and systems. To trust this data, we need to indicate clear paths for the validation process and methodology for any numerical concepts (such as vulnerability and de-rating factors) that contribute to the calculation of safety metrics. First-principle methods (such as fault simulation) can be good candidates for reference and benchmarks. Community-driven benchmarking packages (applications, testbenches, scenarios reference databases, etc.) can hugely contribute to the success and ease of the validation.

86. Platform tuning ME for safety critical real-time applications to meet CAST-32A requirements

- CAST-32A is concerned with the topics that could impact the safety, performance and integrity of a software airborne system executing on Multi-Core Processors (MCP). There is a need of a ME to ensure that safety critical real-time applications to meet CAST-32A requirements. IEEE P2851 should support the approaches and mitigations to ensure that safety critical applications are able to meet WCET targets.

87. Real-time—Considerations for safety analysis ME

- Coherent, synergetic management of the interactions between real-time and safety need to be considered, as well as how the impact should be comprehended in the safety analysis. IEEE P2851 will provide guidance for this management.

88. Real-time—Use of formal methods to verify contention ME

- Establishing and using formal methods are a well-established approach to address design verification complexity. This requirement is concerned with the support of applicable methods or Best Know Methods (BKMs) for verification of real-time performance to comply with CAST-32A. IEEE P2851 will provide guidance for this methodology.

89. Modeling of the radiation working environment ME

- Single event effects are caused by energetic particles that are specific to either the working environment (such as neutrons for automotive/avionics, protons and heavy ions for space applications) or to the device (such as alpha particles emitted by packaging impurities). There are a lot of working environments and/or mission profiles that should be described in a unified way. IEEE P2851 proposes a methodology to describe the working environment (in terms of energetic particles).

90. FDAL/IDAL (Functional/Item Design Assurance Level) Decomposition Analysis ME

- In safety critical avionics, Functional/Item Development Assurance Level (FDAL/IDAL) is the equivalent of ASIL for avionics. FDAL/IDAL differences with ASIL should be clarified. IEEE P2851 plans to provide some guidance/clarification about how to perform the FDAL/IDAL.

91. ME for multi-core COTS considerations in safety analyses

- The use of commercial off-the-shelf (COTS) components and technologies is increasing; today's powerful multi-core CPUs present undisputable advantages in terms of SWaP-C (size, weight, power, and cost), features facility of use and richness of the supporting ecosystems. Designing hardware and software for multi-core CPUs needs to be aware of any requirements and goals for functional safety and reliability aspects. IEEE P2851 plans to provide guidance on how to consider COTS in safety analysis.

92. Activities for multi-core SOC development, in compliance to DO and CAST/AMC/Cert memo DL

- CPU and multi-core CPU IPs are readily available from a variety of commercial IP providers and collaborative/open-source projects. These IPs can be integrated in complex SOC as is or customized according to the application. The synergetic evaluation of safety metrics and goals requires cooperation from multiple partners and is subjected to a variety of requirements and evaluations. It will be beneficial to be able to express quantitative and qualitative attributes in a consistent manner across the design and manufacturing flow.

93. System/component/IP-level HW and SW requirements traceability DL

- Safety and reliability requirements need to be managed at any hierarchical or abstraction level, from technology to system-level or application. Requirements need to be clearly exchanged between suppliers (such as technology or IP providers) and their users (ASIC/board/system) designers. Hardware and software safety-related requirements need to be traced harmoniously and synergistically, in order to be able to address them with a coherent methodology, without having to pay significant overheads in each layer or design stage. IEEE P2851 should provide guidance on this DL.

94. Requirement's traceability in verification, validation, and testing DL

- Requirement's traceability in verification, validation, and testing is a key point to help ensure that all the safety requirements are met. Given the complexity of the supply, design and manufacturing flows, any requirement prepared at any stage of the flow must be handled without omissions, loss of fidelity, or misinterpretations. A consistent view or database populated with clearly defined requirements that can be checked for integrity is a must. Requirements and their resolution must be known to all the partners from the design and manufacturing flow.

95. Non-terrestrial radiation testing ME

- The accelerated radiation tests are a key topic to verify the safety requirements. It should be performed taking into account the different working environment encountered during the lifecycle of the part/unit, component, system, and item. The reuse of results obtained using existing standards needs to be evaluated. Ways to possibly integrate such results in efforts that will comply to the current standard will be presented.

96. Tradeoff assessment for real-time and safety metrics ME

- The real-time and safety requirements are a key points of avionic use cases. A successful and effective real-time scheme requires a careful implementation, sometimes at the detriment of other system qualities (such as safety). Synergetic co-design is expected to improve the outcome of the process. This optimization should be supported by a set of real-time and safety metrics to allow for an effective balancing and extended capabilities at the intersection of the domains. Furthermore, chip level interoperability can be considered to reduce burden on OEM.

97. Real-time ME: System level considerations for deterministic performance

- Concurrency and (massive) parallelism are well-established methodologies for improving both system performance (through multi-core processors, NoC, accelerators) and safety/reliability (n-module redundancy, hardware, and software diversity). However, critical safety applications require a careful system-level implementation to ensure a deterministic performance (CAST-32A [3]). There is a lack of a coherent system-level design methodology to address deterministic performance. A coherent system-level design methodology to address deterministic performance requires the capability to express requirements in a standard format and to produce, use, and present data and metrics related to these requirements.

98. Module design, integration, and testing report DL/Guideline

- The design process is highly hierarchical and builds upon smaller modules to build a larger system. In the classical design flow, functional, performance, and cost metrics are clearly expressed and made available to the user of the module, for an efficient reuse and integration. Safety and reliability information and metrics needs a similar treatment in order to improve the design, integration and testing activities and reporting or documentation. IEEE P2851 will provide DL and guidance to improve the design, integration, and testing report.

99. DB of System RAS (Reliability Availability Serviceability) architecture capabilities

- A database containing the available capabilities and features to manage Reliability-Availability-and-Serviceability of the system is needed. Today's complex systems include a plethora of schemes, techniques, and features to mitigate the effects of faults, errors, and failures. However, the capabilities of many of these schemes are not well documented to the system architect or used in a coherent, synergetic, system-level approach. Moreover, the capabilities, advantages, or disadvantages of each scheme are not presented in unified format or location to allow the user to perform system-level RAS budgeting and optimization. It is proposed to describe RAS capabilities in a database allowing for qualitative (name, applicability, purpose) and quantitative information (such as metrics, overheads) to be stored and used in RAS management schemes.

100. DB of applicable Single Event Effects

- Create a database containing the available Single Event Effects applicable to all the abstraction layers of the system, from technology to system-level. Faults, errors, and failures caused by Single Events can affect all the layers of the system from technology (Single Event Upsets in flip-flops and other sequential logic, Single Event Transients in combinatorial cells etc.) to system (such as Single Event Functional Interrupt – SEFI). The SEE-related information can be produced or consumed by multiple engineers from several fields or organizations and can suffer from misinterpretation and erroneous usage. This is a proposal for a database containing the applicable SEE information for each applicable system part or subpart, described in a coherent, homogenous format.

101. DB of Telemetry parameters that need to be monitored

- Create a list of parameters that need to be monitored, to have data collected by the telemetry framework. Field data is a hugely valuable tool for RAS management and future design efforts improvements. However, the list of information that can/must be collected may not be well known by the maintenance engineers. The proposal is for a list/database of the parameters that can be recorded and the associated record, storage, and analysis procedures.

102. DB of Reliability/Telemetry—Vulnerable circuits/components

- Create a database containing the components and circuits that have the lowest expectations of reliability. Not all circuits and components are created equal; the reliability of the today's technology, while exemplary, can still suffer from defects in specific components references or series. It is common industry practice to include a list of possible causes and issues in the service manuals of most products. A list of circuits that are "weak links", i.e., limiting and most susceptible to early failures in the field will be helpful.

103. Systematic faults DB

- This is a proposal for a list of systematic faults that can affect the system. Systematic faults can be difficult to trace during the complex design flows involving and can be easily omitted when designing safety mechanisms and error management schemes. It could be beneficial to build and reference a list of potential undetected systematic faults that have to be handled by safety mechanisms.

104. DB of spectra for the energetic particles that can cause Single Events Effects/Soft Errors

- This is a proposal for a unified way to describe spectra of particles that can affect the functioning of electronics. Informal formats and approaches to describe a particular radiation environment exists (CSV, XLS, tool-specific formats), but they can benefit from harmonization and integration with the other IEEE P2851 initiatives. A format to store a spectra (flux vs. energy/LET) of energetic particles.

105. DB of common-mode faults

- This is a proposal for a list of common-mode faults that can affect modules and subsystems contributing to the same function. Today's complex systems include a plethora of schemes, techniques, and features to mitigate the effects of faults, errors, and failures. However, the capabilities of many of these schemes are not well documented to the system architect or used in a coherent, synergetic, system-level approach. Moreover, the capabilities, advantages, or disadvantages of each scheme are not presented in unified format or location to allow the user to perform system-level RAS budgeting and optimization. This database is expected to support designers looking to address or mitigate them (e.g., by employing dissimilarity, diversity, and other approaches).

106. DB of system level radiation testing requirements, effects, BKMs for safety analysis

- This is a proposal for a repository containing pointers to BKMs, handbooks and/or papers that describe system level radiation testing BKMs. Hardware testing of complex systems can quickly become a challenging task because of the efforts required to set-up the experiments (radiation tests), exercise the system (Device Under Test – DUT) and interpret the results. While some levels of consensus exist for elementary components (such as memories), the testing of more complex components, boards and systems is not done in an homogenous way by different test partners in the various available facilities that provide radiation beams. A database (repository) containing links, citations or copies of the applicable documents, standards or BKMs for system testing including practical examples and recommended practices will be provided.

107. DB of key parameters to consider for tradeoffs between real-time and safety metrics

- This is a proposal for a list of parameters that are needed to reduce jitter/interference to be compliant to CAST-32A and those are needed to comply with safety metrics. Consider the synergy/conflict of these parameters to the adjacent domain. A list/database of the parameters that are needed to reduce jitter/interference to be compliant to CAST-32A and those are needed to comply with safety metrics will be included.

108. DB of contention/shared resources DB of FuSa-real-time intersections

- This refers to a repository containing a list of on-chip shared resources that should be considered to demonstrate compliance to Worst Case Execution Time and to meet real-time and safety metrics.

109. DB of system level considerations for enabling deterministic performance

- Off-chip capabilities to mitigate interference and comply with CAST-32A requirements. Deterministic performance can be adversely affected by systemic causes (such as incorrect system design). There is a need for a centralized database to store information regarding the management of threats to the deterministic performance of the system. The proposed database will describe available system-level schemes, strategies, and measures to reduce interference, contention, deadlocks, and timeouts.

RAISING THE WORLD'S STANDARDS

3 Park Avenue, New York, NY 10016-5997 USA http://standards.ieee.org

Tel.+1732-981-0060 Fax+1732-562-1571