29th Annual INCOSE international symposium
Orlando, FL, USA
July 20 - 25, 2019

# OMG standard for integrating safety and reliability analysis into MBSE: Concepts and applications

Geoffrey Biggs
Tier IV, Inc.
3-22-5 Hongo, Bunkyo-ku, Tokyo, Japan
gbiggs@ieee.org

Kyle Post
Ford Motor Company
20000 Rotunda Drive, Dearborn, MI, USA
48124
kpost1@ford.com

Andrius Armonas
No Magic Europe
Savanoriu pr. 363, Kaunas, Lithuania
andrius.armonas@nomagic.com

Nataliya Yakymets
CEA LIST LECS,
Point Courrier 174, Gif-sur-Yvette, F-91191
France
nataliya.yakymets@cea.fr

Tomas Juknevicius
No Magic Europe
Savanoriu pr. 363, Kaunas, Lithuania
tomas.juknevicius@nomagic.com

Axel Berres
German Aerospace Center
Lilienthalplatz7
38108 Braunschweig, Germany
Axel.Berres@dlr.de



**Abstract**. Model-Based Systems Engineering (MBSE) is gaining popularity in organizations creating complex systems where it is crucial to collaborate in a multi-disciplinary environment. SysML, being one of the key MBSE components, has a good foundation for capturing requirements, architecture, constraints, views and viewpoints. However, SysML does not provide the necessary constructs to capture safety and reliability information in the system model. A group of industry experts at the OMG has been working since 2016 to define a new specification providing the necessary capabilities. This paper provides an update on the progress of this work. It discusses the proposed specification's use of generic concepts to allow information interchange amongst diverse analyses, its use of existing SysML constructs to provide automation of safety and reliability work in existing modelling tools, and describes several of the supported analysis methods.

## Introduction

Since 2016, there has been an ongoing effort at the Object Management Group (OMG) to define a standard profile for UML that enables modelling of the safety and reliability aspects of a system. This effort was begun due to a growing consensus firstly that model-based approaches have much to offer this important area of systems engineering, and secondly that existing modelling languages, in particular SysML, do not have the necessary capabilities.

A Request for Proposals was published by the OMG in March, 2017 (OMG 2017). The RFP calls for a UML profile that provides SysML with the capability to model safety information, such as hazards and the harms they may cause, model reliability analyses, including Fault Tree Analysis (FTA) and Failure Mode and Effects Analysis (FMEA), and use structured argument notation to organize the model and specify assurance cases. An initial version of the specification was submitted to the OMG on the 28th of August, 2017, in accordance with the OMG procedure. The proposed specification was

heavily revised throughout 2018 as the submission team grew and developed increasingly powerful and integrated model-based solutions for including safety and reliability information in a system model.

At the time of writing, the submission group consists of representatives from 88solutions, The Aerospace Corporation, France's Alternative Energies and Atomic Energy Commission (CEA), Change Vision Inc., Ford Motor Company, GfSE e.V. (the German chapter for systems engineering, Gesellschaft für Systems Engineering), MITRE, Multi Agency Collaboration Environment (MACE), NASA's Jet Propulsion Laboratory, Japan's National Institute of Advanced Industrial Science and Technology, No Magic, Inc., oose Innovative Informatik eG, and Rolls-Royce plc. An increasingly-large number of other contributors provide irregular additional feedback and comments vital to producing a specification that covers a number of fields that are related in their approach to safety and reliability but still have important differences.

The need for a standardized UML profile for addressing safety and reliability aspects emerged long ago – group members have seen a number of commercial-grade model-based safety and reliability solution implementations being developed during the recent years and successfully used in practice. One of the key goals for the new OMG group is to reconcile these different approaches so that the industry does not need to repeatedly design support for safety and reliability in their tools. The specification aims to provide the necessary modelling capabilities for tool vendors to build safety and reliability modelling tools that mimic existing user interfaces while using a modern approach.

In this paper, we present the current state and content of the specification. This paper updates a previous publication from 2017 (Biggs, *et al*. 2018), which discusses the need for this specification and initial work performed. We demonstrate the core concepts of the specification and show how the simple concepts are powerful enough to unite all safety and reliability information across a variety of analysis types. We also demonstrate the specification's approach to automating several safety and reliability analyses, which is built on leveraging existing SysML functionalities to ensure that the profile is usable with existing tooling. Some discussion of the domain-specific aspects of modelling safety is also given. Finally, we describe the roadmap for the final steps of adopting the specification and for how we envision the specification's life after adoption.

## Structure of the specification

The specification is divided up into several inter-dependent packages, as shown in Figure 1. These packages are roughly organized in a hierarchy of increasing specialization.

At the top of the hierarchy lies the core concepts package, providing the most universal concepts of the specification as described in section "Core concepts". More specific concepts are provided by the general safety and reliability package, described in section "General safety and reliability package". The reason for this division is to enable reuse of concepts that are more widely-applicable than safety and reliability (for example, in security) without requiring a separate specification to import concepts that are specific to the safety and reliability domains. This allows additional domains to achieve information interchange via the core concepts with models built using this specification. With the increasing realization that domains such as safety and security need to be handled in tandem, future specifications and system models built using them will benefit from this compatibility.

Finally, the specification defines several method-specific packages. These provide the functionality of the specification in terms of information that can be modelled and analyses that can be performed directly in the model. Several reliability analysis methodologies are covered by the specification. These are currently FMEA, FTA and HARA, and STAMP/STPA is currently being integrated. The modelling constructs that support modelling of these analyses are each contained in a separate package. Interchange of information between these packages is supported by their use of the core concepts

and the general concepts. This allows, for example, a failure from an FMEA built using the FMEA package to be used as an event in a fault tree built using the FTA package.

This re-use of the modelled information between different analyses is a key advantage provided by the specification. It helps ensure consistency between different analyses that use the same or related information, makes it possible to trace between different analyses and find all analyses related to a particular part of the system, and it enables an impact analysis to operate on all reliability analyses together.

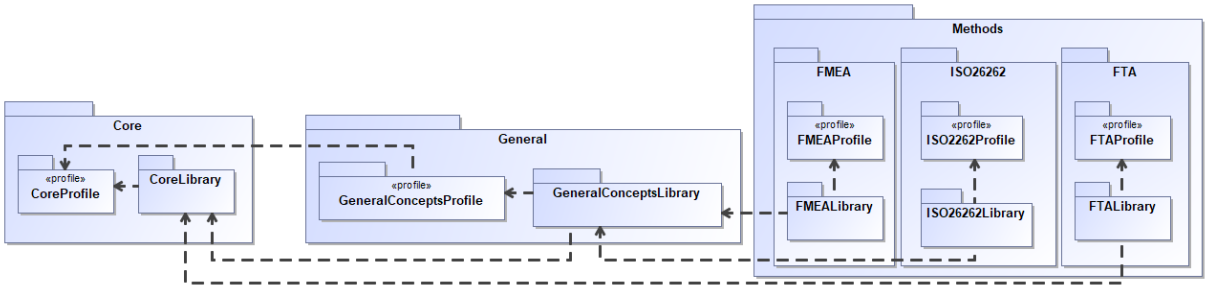


Figure 1. The structure of the specification's packages

The specification title (assigned during the proposal stage) states that it contains a profile, but in reality, this is only half of the specification. Each package in the specification provides both a profile and a model library. Using model libraries has several significant benefits compared with doing everything in a profile. Firstly, it makes use of the full UML structural modeling apparatus instead of just using metamodeling capabilities, which are further limited by the UML prescriptions for stereotyping. Composition aspects are especially important for capturing situation interdependencies and a build-up of composed situations (such as in fault trees). The tools with good support for UML/SysML class and composite structure diagrams can make use of their existing generic functionality for safety modeling. Secondly, it enables end users to extend the frameworks provided by the specification with their own customizations, which is important as safety field is rife with domain- and company-specific methodology extensions. Finally, it is typically easier to make modifications and extensions to model libraries than to profiles, as it entails modeling at lower metalevel. Note that end users are also encouraged to build libraries of reusable safety elements. For example, an organization can accumulate over time a library of modelled safety and reliability information, for example, the FMEA for a specific piece of hardware used across a variety of systems. When constructing a new system that uses an existing part, the relevant safety and reliability information can be pulled from this library directly into the system model for that new system without needing to reconstruct it. Examples of the use of model libraries are given in the FMEA and FTA sections below.

Although the profile is titled "for UML", the specification as a whole makes significant use of SysML elements (Friedenthal, Moore & Steiner 2014). The profile heavily uses existing SysML constructs to enable automation even in existing modelling tools. In particular, parametric diagrams are used in all the reliability analyses to automate the calculation of values such as probabilities and RPNs. By using parametric diagrams, rather than relying on requiring tools to provide new extensions or relying on external tools, it is possible to achieve significant automation of the supported analysis methodologies in any system modelling tool that fully supports the existing SysML specification. To support users of the specification who only wish to work with UML, a limited version of the profiles is also specified. However, it loses much of the functionality that is provided by integrating with SysML.

## Core concepts

In this section, we present the core concepts domain model (depicted in Figure 2). The submission team uses this domain model to derive the CoreLibrary and CoreProfile packages (shown in Figures 3 and 4). The other libraries and profiles of the specification are based on the CoreLibrary and CoreProfile packages, and contain elements and relationships representing concepts common across safety and reliability analysis methods.

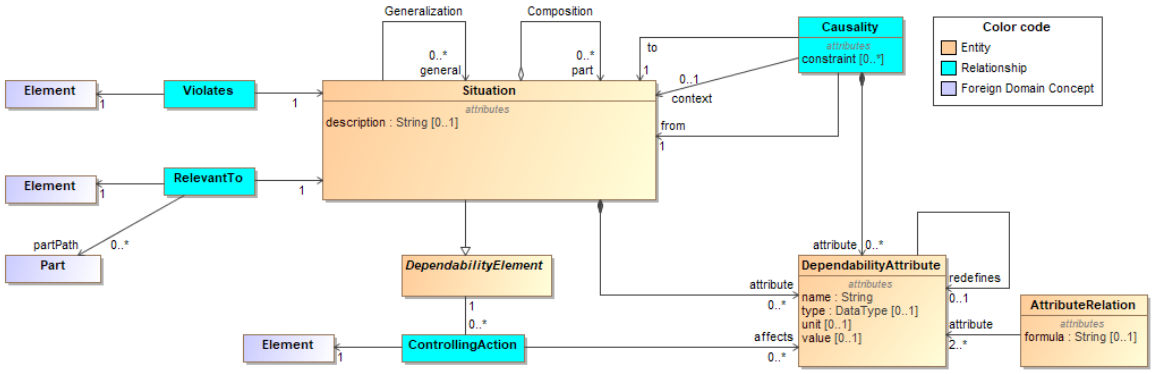


Figure 2. Core concepts domain model

The central element in the core concepts domain model is the "Situation" concept. We define a situation occurrence as a system being in a given place at given time and in a given state. For example, "Boeing 747 with S/N 12305 is being refueled at Gate 7 of Amsterdam Schiphol at 11:45 on Monday, 30th of July 2018." An elementary situation is a classifier. It describes a set of situation occurrences of some type. The system, place, time and state parameters are described by classifiers rather than individual descriptions.

When describing a situation, some of its parameters may be omitted if the situation does not need to be specific with respect to that parameter. For example:

- Fire in the engine compartment of the ship.
- Finger injury of the circular saw operator.

Different Situations can have generalization/specialization relationships between them. Generalization between two situations expresses the subset/superset relationship between the sets of occurrences that these situations represent. For example, "bone fracture" may be defined as a subtype of "Injury".

Situations can have quantitative attributes, such as probability of occurrence. These are defined using the DependabilityAttribute class. Quantitative attributes can be related to each other and to attributes of the system by formulae using the AttributeRelation class. Formulae can be expressed in any language that the modeling tool can compute, including OCL and other executable languages. For example:

$$\text{FMEAItem.RiskPriorityNumber} = \text{Cause.Occurrence} \times \text{FailureMode.Detectability} \times \text{Effect.Severity}$$

Different Situations can be associated with each other using the Causality class, expressing semantic relationships between situations such as simple causality, conditional causality, and probabilistic connections. These relations may also have quantitative attributes, such as the probability of occurrence of the "to" situation if the "from" situation occurs. For example, a car in frequent contact with salt, causing safety-critical parts to corrode, which causes leaks in the brake line, causing the brakes to fail, causing a car accident, causing a passenger injury.

A non-elementary situation (the “Composition” relationship in Figure 2) is a concept encompassing multiple elementary situations: a single system or combination of several systems in a mutable layout, flowing in time through a sequence of states. The choice of whether to use a composite situation with parts described by sub-situations, or to use a single situation, is at the discretion of the modeler. It depends on the modeler's needs, such as the depth of analysis required.

Situations can violate requirements, constraints defined/prescribed for the system, or other specifications describing how the system should operate. For example, a Situation where the system cannot detect glucose level violates the requirement that “the insulin pump must work for 1 week without the need to replace batteries”.

The RelevantTo relationship is used to link situations to system model elements to provide context and relevance for the Sitution. For example, in the aforementioned insulin pump, a Situation where the insulin pump cannot be charged would be related to the main battery element in the system model.

Situations can be mitigated, detected, and prevented via the ControllingAction. The use of this relationship introduces new safety requirements.

## Core library and profile

It was decided early on to reuse as many concepts from the SysML language as possible and only add concepts that are missing in SysML to address safety and reliability aspects of systems. This avoids duplication between two languages that will typically be used together. It also enables tool vendors to implement the new profile and library without requiring new tool capabilities, assuming SysML is supported. This lead to a very small library and profile on top of SysML/UML being sufficient to cover all core concepts. The core domain model is covered by SysML/UML concepts as shown in Table 1. The CoreProfile package is shown in Figure 3. The CoreLibrary package is shown in Figure 4. This profile and library are used by all domain-specific methods in the specification.

Table 1: Mapping of core concepts to the SysML/UML language

| Core concept | SysML/UML concept |
|---|---|
| Situation | A specialization of a Block in SysML and a new stereotype «Situation » |
| DependabilityAttribute | SysML Value Property and UML class attribute |
| AttributeRelation | SysML constraint block |
| Generalization | UML generalization relationship |
| Composition | UML composition relationship |
| Violates | A stereotyped UML dependency |
| RelevantTo | A stereotyped UML dependency |
| Causality | An association/connector combination |
| ControllingAction | A stereotyped UML dependency |

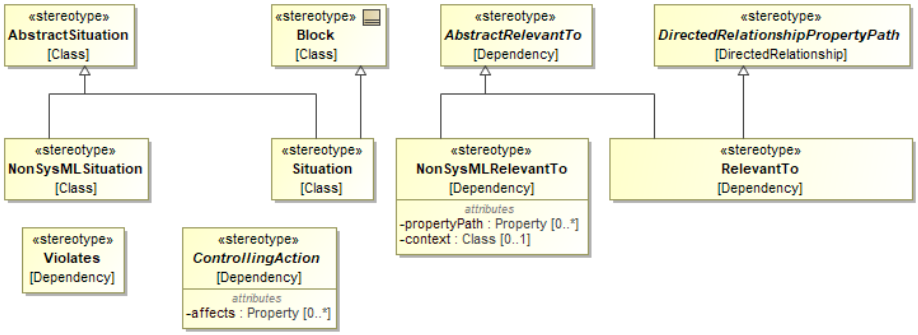


Figure 3. The CoreProfile package


Figure 4. The CoreLibrary package

## General safety and reliability package

The specification includes a general safety and reliability package that extends the core package. It defines common concepts that are used or extended in the method- and domain-specific reliability and safety packages. The package provides a profile, shown in Figure 5, and a model library, shown in Figure 6.

The general concepts contained in this package can be used as-is to model the safety and reliability related aspects of a system. However, the intended purposes of the package are as follows.

1. Provide a common base for the method- and domain-specific reliability and safety modelling packages. The same concepts are used in a number of safety and reliability techniques (such as FMEA and FTA), so the role of this package is to prevent duplication of common concepts in other packages. This also enables movement of information between domains for cross-domain issues. This is particularly important as different domains may use the same concepts with different vocabulary. A common foundation provides a way to translate between these.
2. Provide traceability links between safety and reliability artefacts across the system life cycle. For example, the failure modes defined during Hazard Analysis and Risk Assessment (HARA, defined in the ISO 26262 package) and in an FMEA could be traced and taken into account during an FTA.
3. Provide a foundation on which additional methods, techniques and domains with safety and reliability concerns not currently included in the profile can be built by users. For example, a tool vendor could build an additional package for the railway domain by building on the general safety and reliability foundation. This both reduces effort to introduce an additional domain and allows additional domain packages to be compatible with the existing specification content.

Figure 5 shows the content of the general safety and reliability profile. It extends the Situation and ControllingAction concepts taken from the CoreProfile package and defines concepts for fault propagation modeling, formal analysis and different types of controls.

The FailureMode concept is used in FMEA and ISO 26262 packages. One of the definitions of a failure mode (according to ISO 26262 (ISO 2011)) is a manner in which an element or an item fails. The Fault and Error concepts are defined in IEC 61508 (IEC 2010) and introduced to provide the ability to model fault propagation scenarios (Avizienis, *et al*. 2004).

The FailureState concept might be used in various formal analysis methods based on state machines. Furthermore, one or more failure states could be associated with a failure mode via the RelevantTo relationship to show traceability between these artefacts across several analysis methods. (Safety and reliability analysis flows typically include several methods and techniques.)

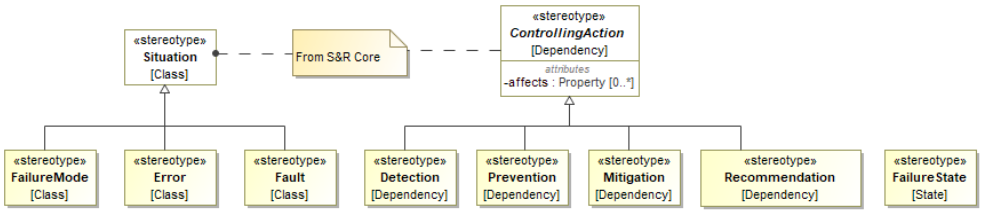


Figure 5. General safety and reliability profile

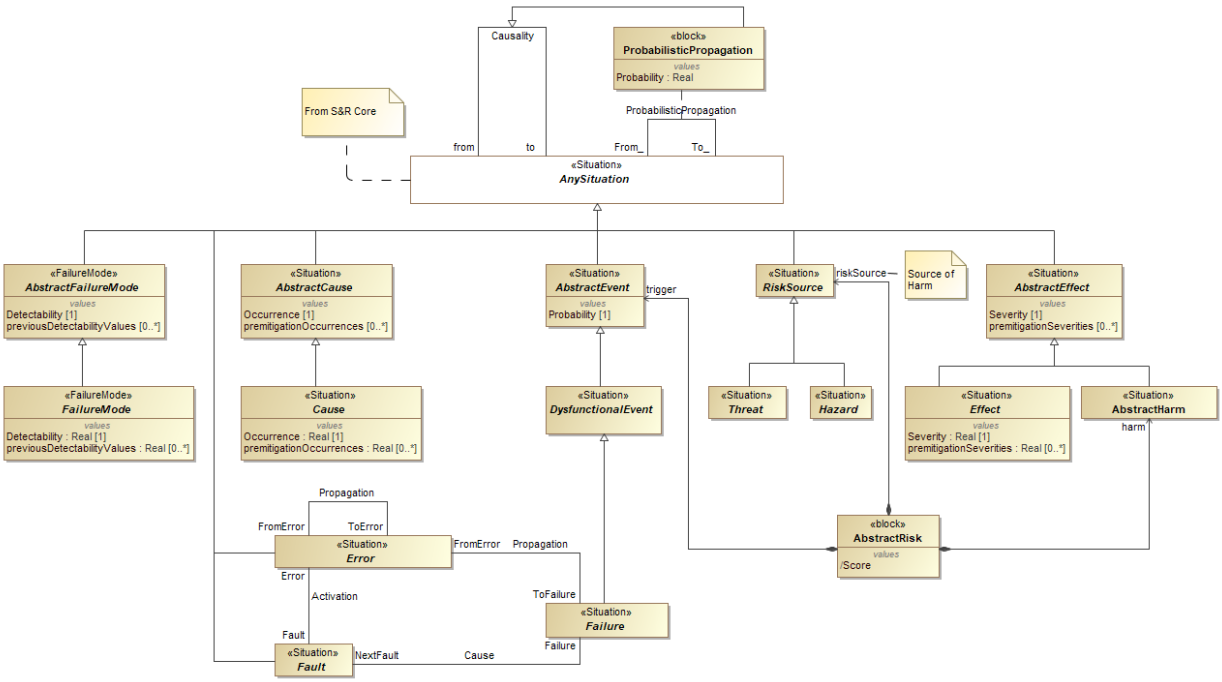


Figure 6. Library of general safety and reliability concepts

The concepts Detection, Prevention, Mitigation and Recommendation extend the generic ControllingAction concept. The Detection concept defines actions or means which exist to detect or plan the appearance of various dysfunctional, feared, or undesired Situations. The Prevention concept defines actions that reduce the probability of occurrence of Situations. The Mitigation concept defines actions that reduce the severity property of a Situation. The Recommendation concept is used to connect a Situation to an action item. The action item is normally a requirement; however, it could be also an advice (for example, rationale).

Figure 6 shows the general safety and reliability library. AnySituation is reused from the CoreLibrary. The ProbabilisticPropagation concept extends the core Causality relationship to introduce the probability of occurrence of each situation in a causality chain.

AnySituation is extended to several more specific safety and reliability concepts. Most of these concepts contain dependability attributes (such as occurrence or detectability) and generalize their corresponding non-abstract concepts. For example, the library contains the following related concepts: AbstractFailureMode and FailureMode, AbstractCause and Cause, AbstractEvent and DysfunctionalEvent, AstractEffect and Effect. This enables the specification of the dependability attributes both as a quantitative value and as a literal referring to a specific standard. The latter can be defined in

domain-specific libraries, because domain specific standards often provide lists of levels or categories for dependability attributes. For example, the ISO 26262 standard recommends four classes of severity and controllability attributes and five classes of exposure or occurrence (ISO 2011).

The AbstractEvent and DysfunctionalEvent concepts define a generic event, the occurrence of which can cause a dysfunctional behavior of the system. They have the probability property. The DysfunctionalEvent concept is a generalization of such concepts as hazardous event, failure, and feared event that are used in the domain-specific packages or might be re-defined by users when-introducing their own methods and techniques.

The AbstractEffect and Effect concepts define the generic effects of safety and reliability artefacts such as risks, failures, and failure modes. As shown in Figure 6, the AbstractEffect concept is extended to the AbstractHarm concept that is further used in the ISO 26262 package to specify risks when modelling a HARA.

The concepts of FailureMode, Cause, Effect and their related abstract concepts include dependability attributes describing the situation before and after application of mitigation or other controlling actions. For example, the FailureMode concept has two attributes related to detectability. The detectability property that defines the detectability of a failure mode after the application of a number of detecting actions (modelled using the Detection concept). The previousDetectabilityValues property that defines the detectability of a failure mode before or during the application of detecting actions (if the analysis requires an intermediate evaluation of the detectability attribute).

The Cause concept is characterized by two occurrence-based properties. The occurrence property shows the final probability and/or the final level of appearance of the analyzed cause after the application of controlling actions proposed during the analysis. The premitigationOccurences property allows the expert to define an initial probability and/or an initial level of occurrence before or during application of the appropriate controlling actions.

The Effect concept has two severity-related properties. The Severity property defines the final severity of the effect after application of controlling actions proposed during the analysis. The premitigationSeverities property defines the initial severity values of levels before or during the application of controlling actions.

The RiskSource concept is a generalization of the Hazard and Threat concepts. The former is widely used in safety domain in various hazard analysis methods and techniques. The latter comes from the security domain. According to ISO 27005 (ISO/IEC 2018), a threat is a potential cause of an incident that may result in harm of systems and organizations. The AbstrackRisk concept describes a generic risk and includes an abstract event, a risk source (for example, a hazard or a threat defined for safety and security contexts) and an abstract harm. The introduction of the RiskSource, Threat, Hazard and AbstractRisk concepts aims to provide a bridge between safety and security analysis. The AbstractRisk and RiskSource concepts are common to both safety and security fields, so they can be used as-is or extended to define safety and/or security related risks and risk sources.

Finally, the fault propagation mechanism as described by Avizienis, *et al*. (2004) is modelled using the Propagation, Cause and Activation dependencies.

## FMEA modelling

Most of the elements needed to model a FMEA according to the IEC 60812:2006 standard (IEC 2006a) belong to the general safety and reliability package, due to their generic nature.

The library used for modeling an FMEA is depicted in Figure 7. White elements belong to the general safety and reliability package. The only two elements that are needed in addition to what is provided by the general safety and reliability package are AbstractFMEAItem and FMEAItem. These represent

a row in a classical FMEA tables, aggregating causes, failure mode, and effects with dependability attributes, and store calculated RPN (risk priority number) values. The difference between AbstractFMEAItem and FMEAItem is that AbstractFMEAItem does not have the RPN value property type set, and FMEAItem has it set to Real as a default approach. Users who prefer to use "major", "medium", "low" (or other values) to rate their risks, can define their own FMEAItem-like elements with RPN values set to enumerations or a String type by specializing AbstractFMEAItem element. This provides customizability of the FMEA modelling capabilities to fit the user's own approach.

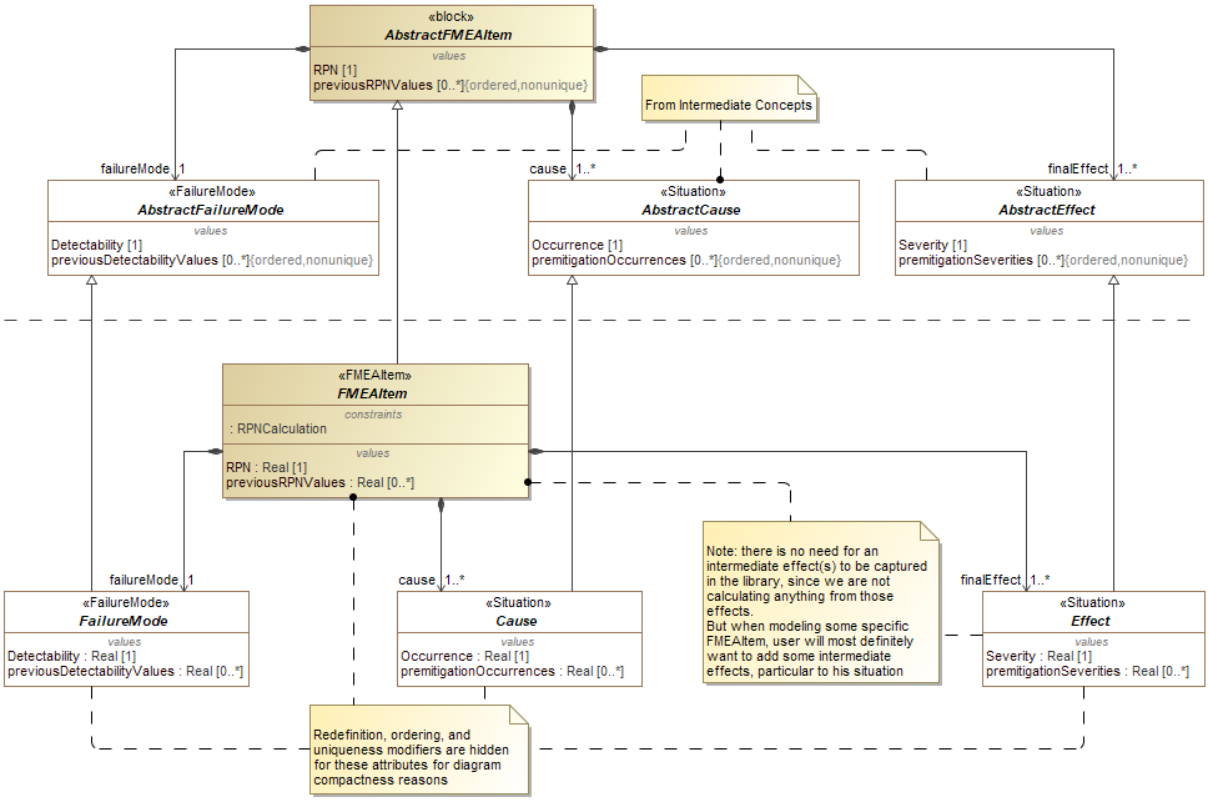


Figure 7. The FMEA library

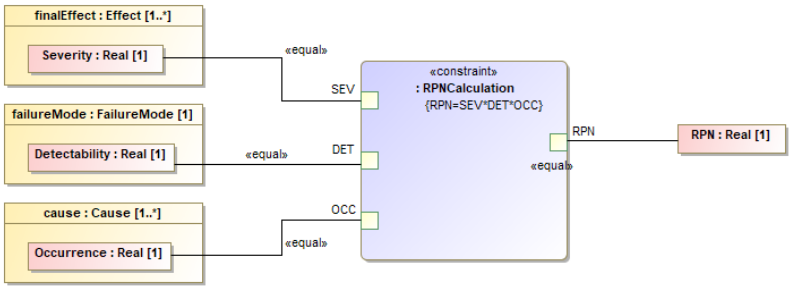


Figure 8. The parametric used to calculate the RPN value, from the FMEA library

A parametric model with a constraint property typed by the constraint block RPNCalculation is defined for FMEAItem block. This defines how the RPN value is calculated from occurrence, detectability, and severity values, for example as in the formula in the RPNCalculation block of Figure 8. As this model is based on SysML, it can be customized by specializing FMEAItem and redefining the constraint property to provide an alternative calculation according to the user's needs.

An FMEA example is given below. The system model (a glucose meter) is shown in Figure 9. As a methodological approach, we recommend introducing a simulation context (GlucoseMeterSimulation in this particular case), which is composed of all systems that are being analyzed from the risk perspective.

The simulation context includes an FMEA analysis (see Figure 10). This analysis is represented by a single GlucoseMeterFMEAItem in this example, but there could be as many FMEAAnalysisItems as necessary. This specific item violates a stakeholder requirement and as a result of the analysis a new safety requirement called “Alarm when battery has sunk” is introduced. A specific cause, failure mode, intermediate effect, and final effect are introduced by specializing Cause, FailureMode, and Effect elements from the FMEA library. Calculations from the FMEAItem parametrics are reused by inheriting and redefining cause, failure mode, finalEffect, and RPN value properties.

Figure 11 shows that causes, failure modes, intermediate effects and effects can be chained for fault propagation. This allows modelling an effect in the lower levels of a system as a cause of a failure in the upper levels of a system.

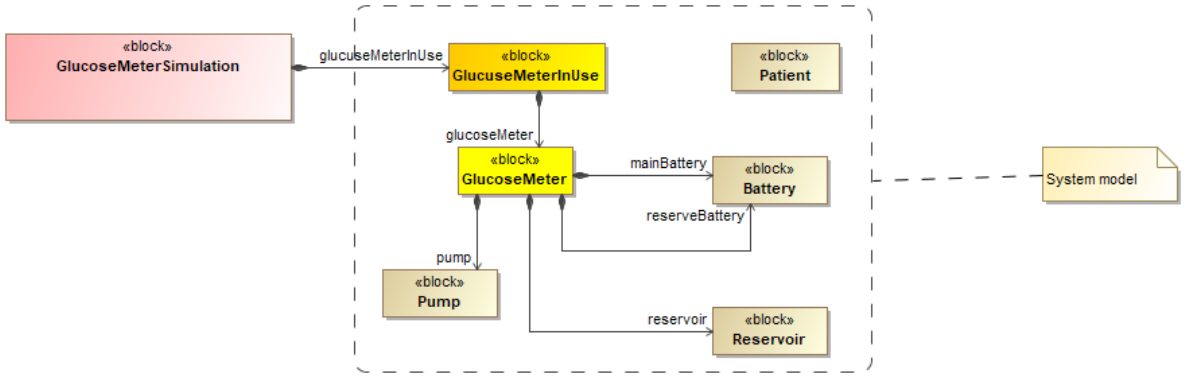


Figure 9. System model and simulation context

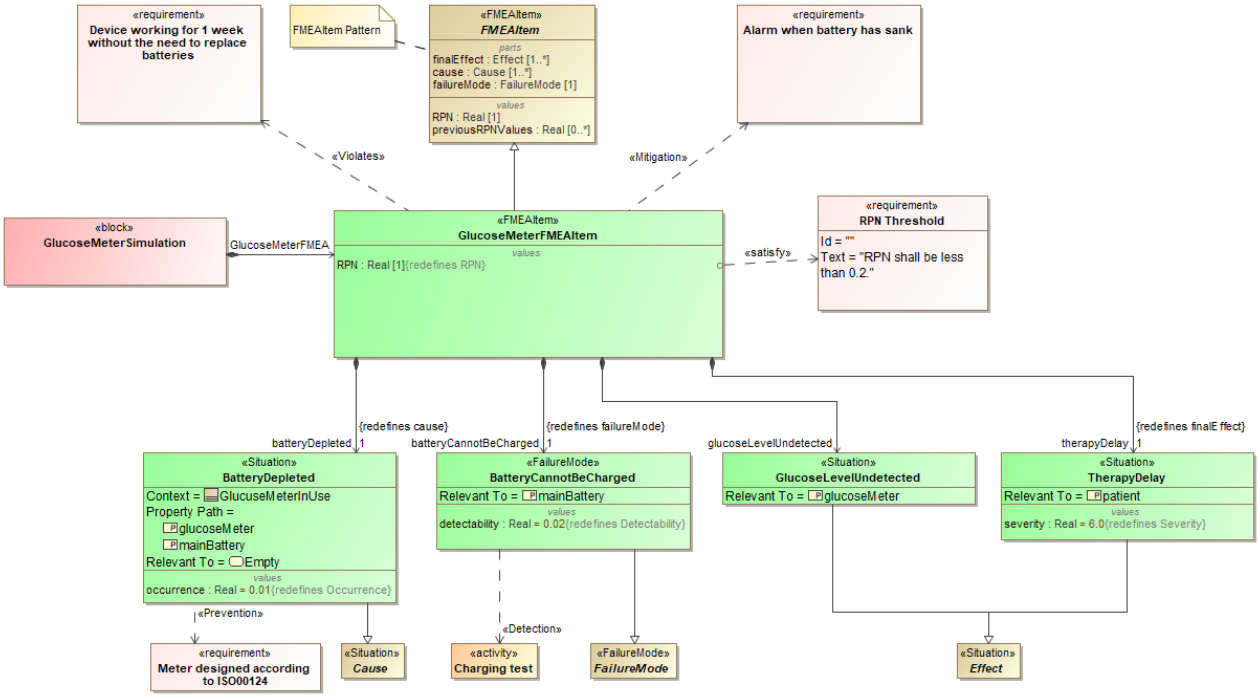


Figure 10. FMEA example (Block Definition Diagram)

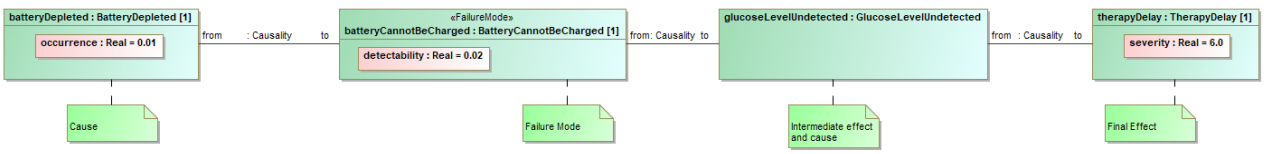


Figure 11. FMEA example (Internal Block Diagram)

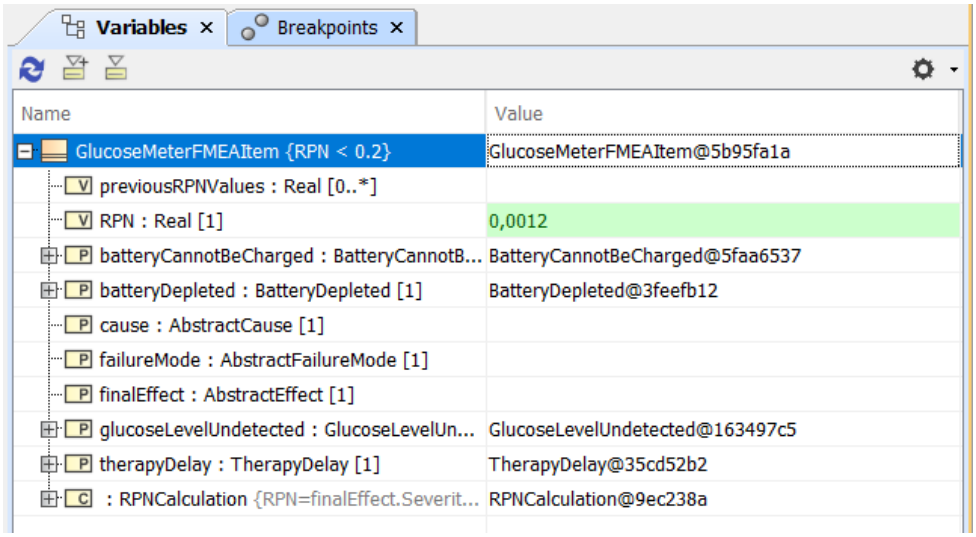


Figure 12. RPN calculation using simulation capabilities of a system modelling tool

Figure 12 demonstrates a commercial tool (No Magic/3DS MagicDraw) simulating the example model out of the box (no modifications to the tool) to calculate the RPN value. It is important to note that the model may look complicated, but tool vendors are expected to implement tabular representations of this model similar to classical FMEA tables to hide most of the complexity while keeping the benefits of employing a model-based approach and integrating with SysML.

## Fault Tree Analysis modelling

Support for Fault Tree Analysis (FTA) modelling is based on the IEC 61025:2006 standard (IEC 2006b). Using this standard ensures that the specification offers a form of FTA that is based on best practices and accepted by practitioners. In order for the package to fully meet the IEC 61025 standard, only a static fault tree analysis is necessary. We are aware that this decision does not cover all forms of fault tree analysis used in practice. However, starting from the analysis given by Ruijters and Stoelinga (2015), it is possible to perform the style of FTA used by the majority of practitioners. It is also possible for a user to extend the capabilities of the FTA package to enable, for example, dynamic fault tree analysis and component fault tree modeling while still remaining compatible with other information modelled using the specification.

The FTA library package is shown in Figure 13. FTA is a top down analysis that identifies possible failures leading to top events. Those top events are typically events that lead to system failure. During the analysis the system is iteratively examined, fault and error events are identified, their dependencies are described, and how they combine is modelled. Fault trees are described depending on the system state. The *FTAElement* shown in Figure 13 is derived from *AnySituation*, ensuring that the system state of the fault tree is fully described.

The gates shown in Figure 13 are used to describe the dependencies of events. For example, if different events lead independently to a system failure, the OR gate is used. If events must all occur, the AND gate is used. For a detailed description of the gates and their use, see the IEC 61025 standard. The calculations defined for the gates in IEC 61025 and applied in the package. A constraint is used for the calculation for each gate. Figure 14 shows the parametric diagram for the AND gate, showing how its output is calculated.

The calculation of all probabilities in a tree can become very complex with very large fault trees. To mitigate the strain this would place on the tool doing the calculation, the use of special tools is recommended for the computation of probabilities. Berres and Schumann (2014) demonstrate how this can be done. A further advantage when using external tools is the calculation of minimal cut sets, which are currently not defined in this specification but are a common output of an FTA.

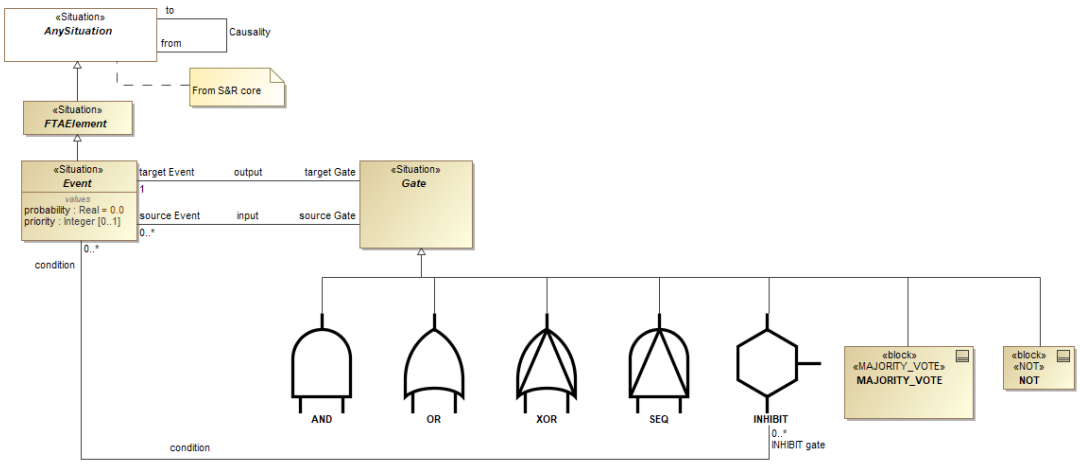


Figure 13. The FTALibrary package with the logic gates

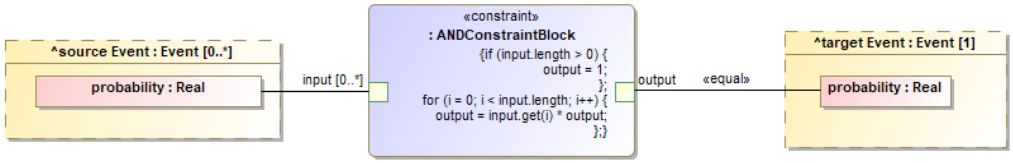


Figure 14. Constraint for the calculation of an AND Gate

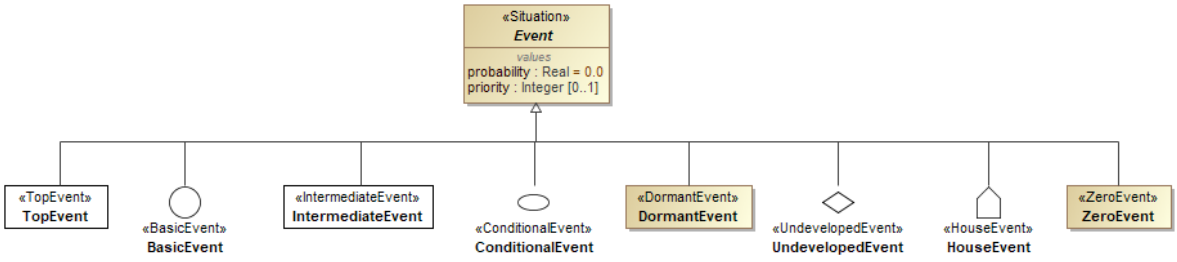


Figure 15. The event library in the FTALibrary package.

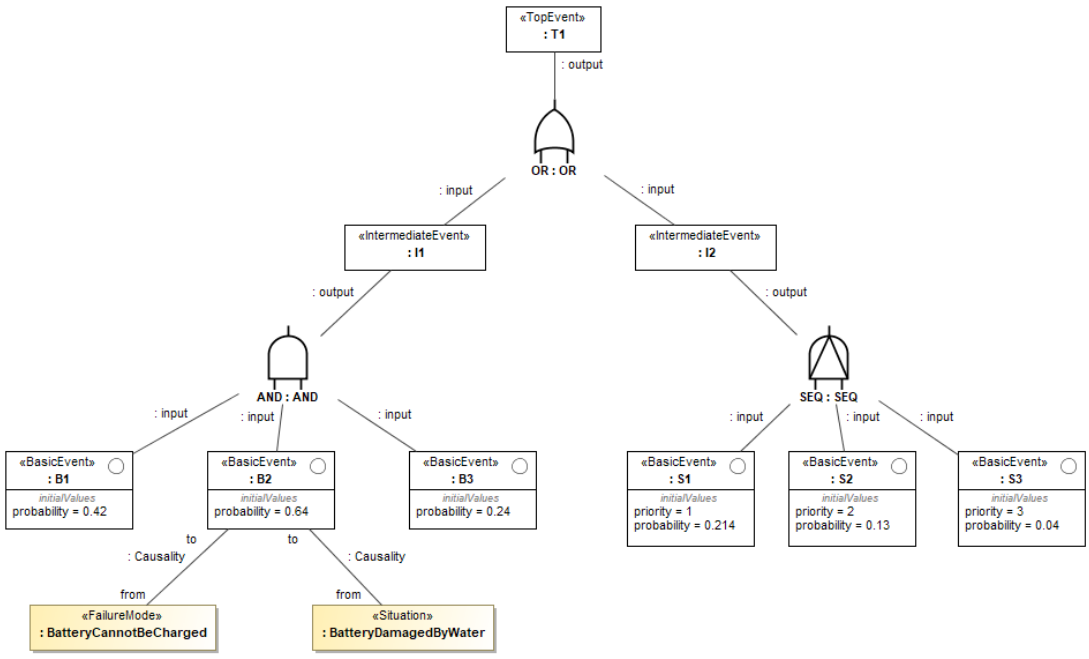


Figure 16 Example of a fault tree using the FTALibrary and FTAProfile packages

The set of events derived from the IEC 61025 standard is shown in Figure 15. All events are derived from Event, which is itself a Situation. The description property can be used to describe the event. Additionally, the probability property can be used to specify the occurrence of the event. The value of this probability can be given or calculated. In case of a given value, the related information source needs to be given, for example accident and incident databases. The priority property is only used by sequence gates to determine the order of events.

The output event of a gate is either an intermediate event or the top event. The top event is the root element of the fault tree and represents the event that leads to system failure. Both types of event are calculated by the modelling tool. The intermediate events are used as inputs to the gates of the tree. This can be seen in the example shown in Figure 16.

In order to combine a FMEA and a FTA, a connection between a failure mode and a fault tree event needs to be made. Therefore, the Cause of an FMEAItem can be interpreted as the event which leads to a failure of a system item. By combining FMEAs and FTAs, both analyses can be used to verify the analysis results. This may lead to a better understanding of the behavior of a system during erroneous behavior.

## Domain adaptation – ISO 26262

While approaches to analyzing and managing reliability tend to be similar across domains, the independent development of safety practices in different domains has led to a variety of approaches, differing acceptance of analysis methods, different concepts that are considered during system analysis and design, and even different terminologies for the same concepts. This fractured landscape makes it more difficult to create a profile that can support as many domains as possible.

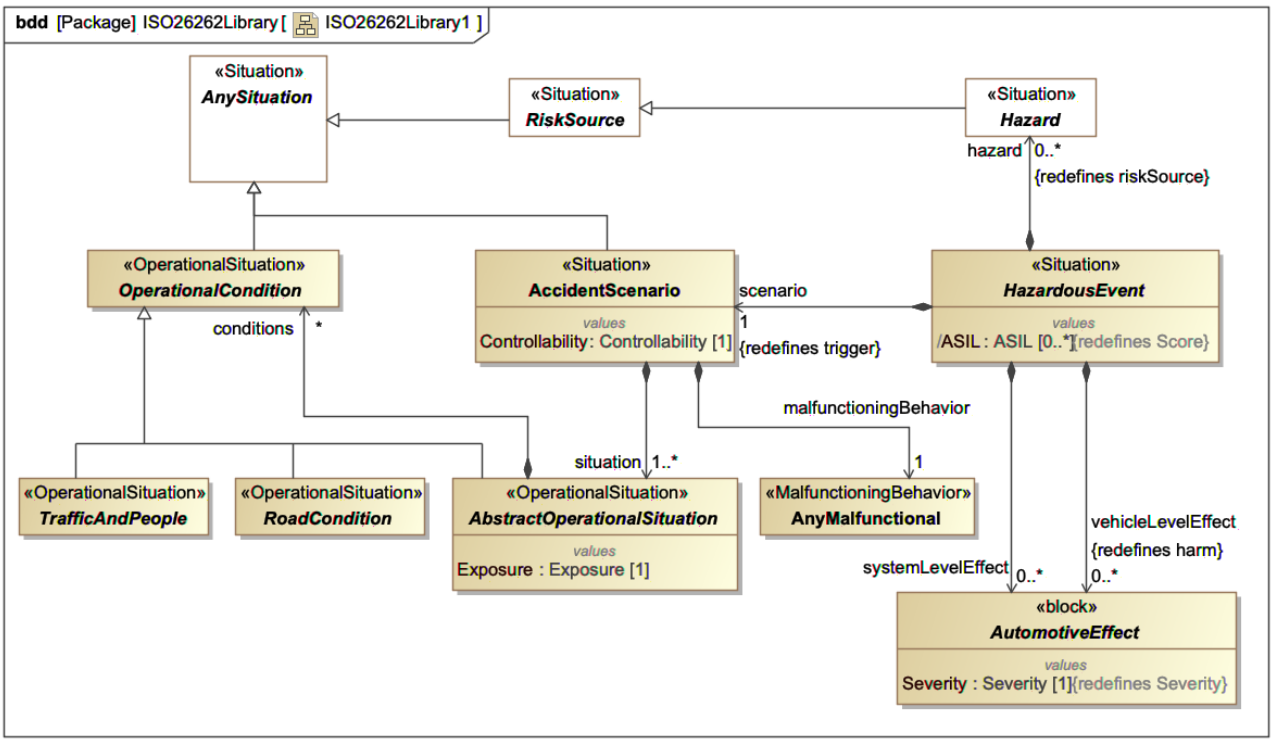


Figure 17: A selection of the automotive-specific concepts contained in the ISO 26262 package. Elements in white are defined in the core or general concepts packages.

The safety packages in the profile allow adaptation to different domains through extensions to the existing profile. The profile primarily supports this through the provision of the general concepts package. A new package can extend the generic safety concepts, altering terminology and relation-

ships and adding new elements as necessary for that domain. By extending the generic safety concepts, the new package remains compatible with the rest of the profile and information interchange with the rest of the specification is possible. An example of this approach is the specification's support for the automotive domain, based on ISO 26262 (ISO 2011). The ISO 26262 package contains elements supporting the analysis and requirement specification aspects of Functional Safety, as specified by ISO 26262 standard for automotive applications. ISO 26262 is a risk based standard derived from IEC 61508 (IEC 2010). The automotive package redefines or extends concepts from the core concepts package and the general concepts package. An illustrative selection of these is shown in Figure 17.

For example, the ISO 26262 package enables modelling a HAZOP, which is typically used to identify malfunctioning behaviors. The failure modes concept is used from the general concepts and specialized as a malfunctioning behavior. This allows the malfunctioning behavior to be related to the system behaviors through the HAZOP guidewords for construction of the HAZOP table. The risk analysis is performed by identifying Hazards that could result from the MalfunctioningBehavior, which in combination with a particular OperationalSituation could result in an AccidentScenario. This information is contained in the HazardousEvent which provides the risk level assessment for the event. Each of these concepts are modelled using elements defined in the ISO 26262 package as extensions of the core and general concepts. This means that the same elements can be used in other analyses in the model, such as in an FMEA.

ISO 26262 represents risk levels using Automotive Safety Integrity Level (ASIL) concept. The specification models this as redefining the Score value of the AbstractRisk general concept. Values that the ASIL may take are defined in an enumeration contained in the ISO 26262 package.

## Conclusions and roadmap

This article has described progress on creating a profile that integrates safety and reliability information into system models using SysML. The profile covers a range of reliability analysis methods, providing facilities for automating the calculations involved in an analysis. It provides modelling structures for storing, inspecting and working with safety information. These structures are adaptable to new domains to account for the variation in safety approaches and standards across domains. Most importantly, the profile provides a foundation upon which new model-based tools with safety-engineering-specific and reliability-engineering-specific user interfaces can be built, improving the efficiency of the work while also improving information consistency, maintainability and analyzability.

The profile is expected to be completed by early 2019, following which it will go through a year-long finalization phase (part of the OMG standardization process) to fix any problems found during implementation. Following publication in early 2020, we expect regular updates to the profile to be published that add new reliability and safety analysis methods and support for new safety domains. The profile has been structured in a way that allows these extensions to be made without disrupting the existing parts of the profile while still maintaining compatibility with them, ensuring the profile can continue to be used even as its usage domains expand.

## Acknowledgements

The authors would like to acknowledge the other participants of the submission team, and in particular Dave Banham of Rolls-Royce PLC, for their comments during the writing of this paper.

## References


Avizienis, A., Laprie, J. C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing, 1 (1), 11-33.

Berres, A., & Schumann, H., 2015, 'Closing the safety process gap: Early integration of safety assessment methods into systems engineering', Tag des Systems Engineering.

Biggs, G., Armonas, A., Juknevicius, T., & Post, K., 2018, 'Integrating Safety and Reliability Analysis into MBSE: Overview of the new proposed OMG standard', INCOSE International Symposium 28 (1).

Friedenthal, S., Moore, A., & Steiner, R., 2014, A Practical Guide to SysML, 3rd edn., Morgan Kaufmann, Waltham, MA.

Ruijters, E., & Stoelinga, M., 2015, 'Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools', Computer science review 15.

IEC, 2006a, Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA), (IEC 60812:2006), International Electrotechnical Commission, Geneva, Switzerland.

———, 2006b, Fault tree analysis (FTA), (IEC 61025:2006), International Electrotechnical Commission, Geneva, Switzerland.

———, 2010, Functional safety of electrical/electronic/programmable electronic safety-related systems, (IEC 61508:2010), International Electrotechnical Commission, Geneva, Switzerland.

ISO, 2011, Road vehicles - Functional safety, (ISO 26262:2011), International Organization for Standardization, Geneva, Switzerland.

ISO/IEC, 2018, *Information technology — Security techniques — Information security risk management*, (ISO/IEC 27005:2018), International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland.

OMG, 2017, Safety and Reliability for UML Request for Proposals, (ad/17-05-05), Object Management Group. Needham, MA (USA), from http://www.omg.org/cgi-bin/doc.cgi?ad/2017-3- 5.