Promoting MBA in the Rail Sector by Deriving Process-related Evidence via MDSafeCer

Barbara Gallina[a], Elena Gómez-Martínez[b,*], Clara Benac-Earle[c]

[a]*Mälardalen University, Västerås, Sweden*
[b]*School of Architecture, Computing and Engineering, University of East London, London, United Kingdom*
[c]*Babel Group. E.T.S. Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain*

Abstract

An EN 50129-compliant safety case should include process-related evidence in terms of quality as well as safety management. Potentially innovative engineering methods developed in academic settings could act as process-related evidence. However, to ease their acceptance within the rail industrial settings, the adequacy of these methods need to be justified. In this paper, we extend our previous work and we provide a broader justification including performance aspects aimed at showing that the entire MBA (Model-Based design methodology for Assessing performance and safety requirements of critical systems) is partly compliant with EN 50128.To do that, we tackle safety and performance process-related compliance as follows: we first manually check if MBA includes EN 50128-compliant process elements, then we model MBA in compliance with Software Process Engineering Meta-model 2.0, then, we derive process-based arguments from the MBA process model by using the MDSafeCer (Model Driven Safety Certification) method. By doing so, we provide a twofold contribution: we further validate MDSafeCer in the rail domain and we strengthen MBA.

Keywords: EN 5012x, model-driven, safety certification, process assessment

[*]Corresponding author

Email addresses: `barbara.gallina@mdh.se` (Barbara Gallina), `e.gomez@uel.ac.uk` (Elena Gómez-Martínez), `cbenac@fi.upm.es` (Clara Benac-Earle)

1. Introduction

Given the awareness concerning the impossibility of achieving absolute safety [1], due to epistemic and logic doubts [2], “safe enough is what we can aim for” [1]. There are several aspects to be considered when ensuring that a system is safe enough. In the railways domain, to the question: How safe is safe enough? the answer given in [3] is the following: “To ensure that the railway industry takes decisions with the proper balance of safety, performance and cost and that are consistent, legal, ethical and workable”. Therefore, different requirements with regards to safety, performance and cost, are needed and must be addressed via adequate development processes and methods.

According to the Comité Européen de Normalisation Electrotechnique (CENELEC) standard series, manufacturers of rail vehicles shall justify via a safety case that their vehicles are adequately safe for their intended applications. More specifically, the CENELEC EN 50129-compliant safety case should include arguments aimed at explaining why the included evidence (e. g., safety and quality management) is adequate to support the safety claims. Arguments should specifically refer to the appropriate Safety Integrity Level (SIL) since the stringency from one level to another changes.

Recently proposed and potentially innovative engineering methods could act as process-related evidence. However, to ease their acceptance within the rail industrial settings, the adequacy of these methods need to be justified. Model-Based Assessment (MBA) [4] a Model-Based methodology for Assessing performance and safety requirements of critical systems at early stages of the design phase, is a recently proposed and potentially innovative model-driven process for the design and verification of software architectures. MBA has been validated in research settings in cooperation with industry [5, 4]. The adoption of MBA in the rail domain, however, is not straightforward due to the current absence of compelling arguments concerning its adequacy. To formulate such arguments it is needed to explain why the selection of the process elements that compose MBA is compliant with the CENELEC requirements. Model-Driven Safety Certification (MDSafeCer) is a method aimed at speeding up the creation of process-based arguments, derived from process models, given in standardized process languages e.g., Software Process Engineering Meta-model (SPEM) 2.0 [6]. The usage and potential effectiveness of MDSafeCer has been illustrated in the automotive [7] and rail domain [8]. In this paper, we use MDSafeCer to derive part of the needed justification concerning the adequacy of MBA as safety and quality management evidence. By doing so we provide a twofold contribution: we further extend and validate MDSafeCer and we strengthen MBA by deriving safety case fragments aimed at showing its adequacy to design software sub-systems in compliance with the European standard EN 50128. More specifically, we consider the design of a door control management subsystem (within a specific train control monitoring system) in a suburban train. This subsystem is expected to have doors with a button that enables passengers to open them upon request. The malfunctioning of this system may endanger the system safety. The assumed Safety Integrity Level (SIL) is SIL 2. Given this system, we focus our attention on justifying adequacy with respect to SIL 2. Given the pattern-based nature of our justification, it can be flexibly changed to argue about a different level, where necessary.

In [9] we focused on safety, i. e., the part of the methodology we called Model-Based Assessment for Safety (MBASafe) [5], and showed that the methodology was partly compliant with the CENELEC EN 50128 standard. With this aim, we modelled MBASafe in compliance with the SPEM 2.0 and, then, we derived process-based arguments using MDSafeCer. In this paper, we show that the whole MBA methodology presented in [4], i.e., including performance, is partly compliant with the CENELEC EN 50128 standard.

The rest of the paper is organized as follows. In Section 2, we present essential background. In Section 3, we collect elements of EN 50128-compliance and we model in SPEM 2.0 the compliant portion of MBA. In Section 4, we derive safety case fragments for arguing that MBA partially meets EN 50128. Section 5 discusses our proposal. In Section 6, we present some related work. Finally, concluding remarks and future work can be found in Section 7.

2. Background

In this section we present the essential background on which our work is based on. In particular, in Section 2.1 we provide essential information concerning safety cases and their possible representation. In Section 2.2, we recall necessary information on the CENELEC standard series. In Section 2.3, we recall a subset of SPEM 2.0 concrete syntax. Finally, in Section 2.4, model-driven engineering principles and methods, relevant for this paper, are recalled.

2.1. Safety Cases and Safety Cases Representation

A safety case is defined as "a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment" [10]. Such argument typically includes process and product-based sub-arguments. To document safety cases, several approaches exist both graphical and textual. Goal Structuring Notation (GSN) [11] is one of the graphical ones and it is here selected because of its active community and its current level of maturity. GSN is a graphical notation, which permits users to structure their argumentation into flat or hierarchically nested graphs (constituted of a set of nodes and a set of edges), called goal structures. To make the paper self-contained, in Fig. 1, we recall a subset of the GSN concrete syntax used in Section 4. As Fig. 1 shows, all the nodes are characterized by an identifier (ID) and a statement, which is supposed to be written in natural language.

We recall that a *Goal* represents a claim about the system; a *Strategy* represents a method that is used to decompose a goal into sub goals; a *Solution* represents the evidence that a particular goal has been achieved; a *Context* represents the domain or scope in which a goal, evidence or strategy is given; *Supported by* represents an inferential (inference between goals) or evidential (link between a goal and the evidence used to substantiate it) relationship. Finally, *In context of* represents a contextual relationship. To create argumentation patterns, i.e., reusable goal structures, specific pattern constructs are at disposal, as shown in Fig. 1. Within patterns, in addition to the constructs presented in Fig. 1, curly brackets are also used to denote variables. Structured Assurance Case Metamodel (SACM) [12] is an Object Management Group (OMG) standard aimed at unify and standardize the graphical notations (including GSN) broadly used for documenting safety cases.

For sake of precision, it should be noted that at the time of writing, the stable version of SACM (1.1) only addresses a subset of GSN modeling elements. Pattern constructs, for instance, are not addressed yet. The to-be-finalized new version of SACM (2.0) [13] is expected to properly address patterns and other useful modelling elements, which are expected to be further developed within the AMASS project [14]. The current beta version of SACM 2.0 only provides initial support for patterns. More specifically, via the added boolean attribute, called *isAbstract* is possible to indicate whether a model element is considered to be abstract and consequently indicate whether an element is part of a pattern or not.

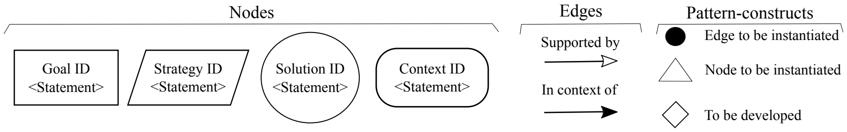


Figure 1: Subset of GSN concrete syntax.

2.2. The CENELEC EN 5012x

The CENELEC EN 5012x is a family of standards that contains requirements and recommendations concerning processes to be followed for the development and assurance of safety-critical systems. This family of standards is used for the certification of railway systems and signaling control-command equipment. As it was documented within the deliverable D6.1 of the MODSafe project [15], Light Rail, Metros, Trams are still characterized by a diversified landscape of safety requirements, safety models, roles and responsibilities, safety approval, acceptance and certification schemes. However, convergence towards the CENELEC standard series is evident.

In this section, we briefly present the portions of EN 50126, EN 50129 and EN 50128 that are necessary to understand Section 4.

2.2.1. EN 50126

EN 50126 [16] defines a fourteen-phase process to manage Reliability, Availability, Maintainability and Safety at system level. The Risk Analysis Phase is the third phase. The objective of this phase is multi-fold: 1) identification of the hazards associated with the system; 2) estimation of the risk associated with the hazards; 3) development of a process for risk management. One of the outcome of the Risk Analysis phase is the assignment of a SIL to any safety relevant function or system or sub-system or component. A SIL specifies a target level of risk reduction and is typically defined in components that operate in a safety-critical system. There are four discrete integrity levels associated with SIL with SIL 4 the most dependable and SIL 1 the least. The SIL allocation is made taking into account the rate of dangerous failures and tolerable hazard rate of the function, system, sub-system or component. The SIL of a system to be developed is determined on system level. The software "inherits" the SIL as any other part of the system through decomposition. Then, EN 50128 defines what must be done to develop SW functions with that SIL.

2.2.2. EN 50129

EN 50129 [17] defines the conditions that shall be satisfied in order that a safety-related electronic railway system/sub-system/equipment can be accepted as adequately safe for its intended application. These conditions are constituted of three types of evidence: Evidence of quality management, Evidence of safety management, and Evidence of functional and technical safety. The documentary evidence that these conditions have been satisfied shall be included in a structured safety justification document, known as the safety case. The safety case shall be structured in six parts. In this sub subsection we limit our attention to the following parts: Part 2 Quality Management Report, this shall contain the evidence of quality management, e.g., evidence of adequate organizational structures as well as evidence of adequate personnel competence and training; Part 3 Safety Management Report, this shall contain the evidence of safety management, e.g., evidence that the safety management process consists of a number of phases and activities, which are linked to form the safety life-cycle in compliance with EN 50126 and with EN 50128 at software sub-system level. The software architecture design phase should for instance be aligned with the system architecture design. Part 6 Conclusion, this shall summarize the evidence presented in the previous parts of the safety case, and argue that the relevant system/sub-system/equipment is adequately safe, subject to compliance with the specified application conditions.

It should be noted that the depth of the evidence presented and the extent of the supporting documentation should be appropriate to the SIL of the system/sub-system/equipment under scrutiny.

2.2.3. EN 50128

EN 50128 [18] focuses on processes for the development, deployment and maintenance of safety-related software for railway control and protection applications. EN 50128 does not mandate the use of a particular software development lifecycle. It only provides normative tables and recommendations concerning specific process elements, e.g., roles, work products, techniques, tools, tasks. Illustrative software route maps are indicated, however, a process engineer is responsible for the selection and composition of adequate process elements aimed at achieving the required software integrity level. To make the paper self-contained, we recall those process elements related to the Software Architecture & Design Phase that are in relation with MBA.

Tasks and related work products- The design task should receive in input the Software Requirements Specification and should deliver in output the Software Architecture Specification, the Software Design Specification, the Software Interface Specifications, the Software Integration Test Specification, the Software/Hardware Integration Test Specification, and the Software Architecture and Design Verification Report. It should be pointed out that to collect requirements systematically, IEEE 830 [19] and IEEE 1012 [20] standards, defined by the Institute of Electrical and Electronics Engineers (IEEE), are also typically used in combination with EN 50128. The verification task should receive in input all necessary system, hardware and software documentation and should deliver in output a Software Verification Plan a set of Software Verification Report(s), and a Software Quality Assurance Verification Report. The validation task should receive in input all necessary system, hardware and software documentation and should deliver in output a Software Validation Plan, a Software Validation Report and a Software Validation Verification Report.

Guideline- We limit our attention to Annex A. According to Table A.4, Software Design and Implementation, formal methods are recommended (R) for SIL 1 and SIL 2, and highly recommended (HR) for SIL 3 and SIL 4. More generally, modeling is HR for SIL1-4. According to Table A.5, Verification and Testing, formal proofs are R for SIL 1 and SIL 2 and HR for SIL 3 and SIL 4. According to Table A.17, Petri nets are R for SIL 1 and SIL 2 and HR for SIL 3 and SIL 4. Finally, according to Table A.22, Object Oriented Detailed Design is R for SIL 1 and SIL 2 and HR for SIL 3 and SIL 4.

Roles- We limit our attention to Annex B. According to Table B.2, a designer shall: transform specified software requirements into acceptable solutions; own the architecture and downstream solutions; define or select the design methods and supporting tools; apply appropriate design principles and standards; develop component specifications where appropriate; maintain traceability to and from the specified software requirements; develop and maintain the design documentation; ensure design documents are under change and configuration control. With respect to expected competencies, a designer shall be competent in: engineering appropriate to the application area, the safety design principles, design analysis & design test methodologies, and understanding the problem domain. Moreover, a designer shall understand: the constraints imposed by the hardware platform, the operating system and the interfacing systems and the relevant parts of EN 50128. Finally, (s)he shall be able to work within design constraints in a given environment.

According to Table B.5, a verifier shall be: competent requirements engineering and experienced in the applications domain and in the safety attributes of the applications domain. Moreover, a verifier shall understand: the overall role of the system and the environment of application; analytical techniques and outcomes; the applicable regulations; and the requirements of EN 50128.

Finally, according to Table B.7, a validator shall be competent in: the domain where validation is carried out as well as various validation approaches / methodologies and be able to identify the most suitable method or combination of methods in a given context. Moreover, he/she shall be: experienced in safety attributes of applications domain; capable of deriving the types of validation evidence required from given specifications bearing in mind the intended application as well as of combining different sources and types of evidence and synthesize an overall view about fitness for purpose or constraints and limitations of the application. A validator shall also have analytical thinking ability and good observation skills as well as overall software understanding and perspective including understanding the application environment. Finally, he/she shall understand the requirements of EN 50128. It should be also mentioned that the verifier and validator can be the same person in case of SIL1 and SIL2.

Performance-specific requirements- Annex A (more specifically A.5, A.13, A.17, and A.18) and Annex D (more specifically D.39 and D.40) provide requirements concerning performance modeling (R for SIL1 and SIL2; HR for SIL3 and SIL4) and verification via testing (HR for all levels except for SIL0). It should be noted, however, that specific techniques are not indicated.

2.2.4. Certification Liaison Process in the Railways Domain

In contrast with the avionic domain, in the context of the railways domain, a precisely stated process (which in the avionic domain is called Certification Liaison Process) to indicate the expected interactions between the applicant and the certification body is not present. However, based on the requirements stated in EN 5012x, a planning and an execution phase are also envisaged. Within EN 50128, a Software Quality Assurance Plan is expected to be developed. During the execution phase, this plan is expected to be respected. In this paper, we focus on the execution phase and we assume that the execution is conducted according to the corresponding plan.

2.3. SPEM 2.0

SPEM 2.0 [6] is the OMG's standard for systems and software process modelling. SPEM 2.0 supports the definition of reusable process content, i.e., work definition elements (e.g., tasks, etc.) as well as elements representing: who is responsible for the work (roles), how the work should be performed (guidance), what should be expected as in/output (work-products) and which tool should be used to perform the work. In Table 1, we recall a subset of SPEM 2.0 modelling elements, which can be interrelated to model static process structures. SPEM 2.0 does not support process enactment. Thus, process models cannot be executed and for this reason information regarding e.g., the exact start and end date of units of work cannot be modelled. To enact process models, SPEM2.0 extensions (e.g., EXE-SPEM [21]) have been proposed in the literature but their tool support is not mature yet. Despite its general-purpose nature, SPEM 2.0 implicitly permits process engineers to model non-functional concerns. A dedicated SPEM 2.0 extension [22] to explicitly model safety concerns (e.g. integrity levels) has been proposed. However, its maturity has not been demonstrated yet.

Table 1: Subset of SPEM 2.0 modelling elements

Task	**Role**	**WorkProduct**	**Tool**	**Guidance**
🏷	👤	📄	🔨	📐

Finally, we also recall that SPEM 2.0 offer modeling support for process-related patterns, via the *capability patterns.*

2.4. Model-driven Engineering Principles and Derived Methods

Model-driven Engineering (MDE) principles consist of the exploitation of models to capture characteristics at different abstraction levels of the development life-cycle. For automation purposes, vertical as well as horizontal model transformations are used to refine models (model-to-model transformations). A model transformation transforms a source model (compliant with one meta-model) into a target model compliant with the same or a different meta-model. A standard transformation can be defined as a set of rules to map source to the target. Each rule describes how to transform source instances to the identical target. In the remaining of this subsection, we recall necessary information on the two model-driven methods that are in focus in this paper.

2.4.1. MBA

Gómez-Martínez et al. [4] propose a Model-Based methodology for Assessing (MBA) performance and safety requirements of critical systems at early stages of the design phase. The methodology is constituted of four chained tasks, which can be iterated and are:

1. The *design* task (focus on the functional specification) is carried out by the designer and focuses on modeling the software system architecture by means of Unified Modeling Language (UML) diagrams, being these diagrams the outcome of this step.

2. The non-functional specification is in reality a composite task constituted of two sub-tasks, which can be executed in parallel: the non-functional *performance specification* and *safety specification*. These sub-tasks are carried out by the performance engineer and the safety engineer, respectively. The first one augments UML designs with performance annotations using MARTE (Modeling and Analysis of Real-Time Embedded Systems), the real-time extension to UML for specifying performance information. The second one, the safety specification task, consists of specifying safety requirements using Safety Contract Fragments (SCF) [23]. SFCs are in turn mapped into OCL constraints and included within the UML diagrams.

3. The *transformation* task is aimed at obtaining a formal architectural specification. This activity is carried out by a Petri net expert (Verifier) who translates the UML diagrams augmented with MARTE annotations and OCL constraints into Generalized Stochastic Petri nets (GSPN) [24]. This transformation is divided into two steps. During the first step the UML diagrams, including MARTE annotations, are automatically translated using the ArgoSPE plugin [25]. During the second step, OCL constraints are manually transformed following the rules described in [4], which are based on the guidelines given in [26]. The results of the two steps are then merged using the algebra tool of GreatSPN [27].

4. The *verification & validation* task is aimed at verifying via GreatSPN tool that the safety requirements are satisfied. In the case that the design does not meet the performance and safety requirements, systematized recommendations to improve the design are formulated and a new iteration is carried out.

2.4.2. MDSafeCer

MDSafeCer (Model-driven Safety Certification) [7, 9] is a method that adopts MDE principles to enable the semi-automatic generation of composable process-based argument-fragments within safety cases. Via MDSafeCer, process models compliant with a process modeling language meta-model (e.g., SPEM 2.0) are transformed into argumentation models compliant with SACM and presented via for instance GSN-goal structures. MDSafeCer generates process arguments based on a possible argumentation pattern, which is constituted of a top level claim stating that "the adopted *p* process is in compliance with the required {S} of standard- level {intLev}", where p, S, L are variables indicating respectively a specific process, a set of standards, a specific integrity level. This claim can be decomposed by showing that all the process activities have been executed and that in turn for each activity all the tasks have been executed and so on until an atomic process-related work-definition unit is reached. The pattern used within MDSafeCer, called *Process compliance*, is depicted in Fig. 2 and in Fig. 3. These figures are taken from [9]. The structure of *Process compliance* partially borrows from the the *Goal decomposition* pattern and incorporates the divide and conquer principle.

Fig. 2 represents a pattern that considers only a 3-layer work-breaking-down structure. A process is divided into phases, which in turn are divided into activities. A richer hierarchy could be considered by breaking activities down further into tasks and finally tasks into steps. The 3-layer granularity is however sufficient for this paper since MBA can be considered a 2-layer hierarchy, i.e., a phase constituted of five flattened activities. The five flattened activities are named tasks in accordance with SPEM 2.0 models.

For sake of clarity, it should be stated that the semantic mapping between SPEM 2.0 concepts and SACM was previously presented in [7]. However, given the recent introduction of the new but to-be-finalized version of SACM (SACM 2.0, Beta version) [13], the previous mapping is here updated and presented in Table 2.

The only difference consists of the substitution of *InformationElement* with *ArtefactElementCitation*, shortened with *AEC* in Table 2. The mapping related to the relationships aimed at connecting nodes of argumentation-structures remains unchanged. Thus, a relationship between a task and a role/tool/work product/guideline is still mapped into a *supportedBy* relationship in GSN and into an *AssertedEvidence* in SACM. This mapping could

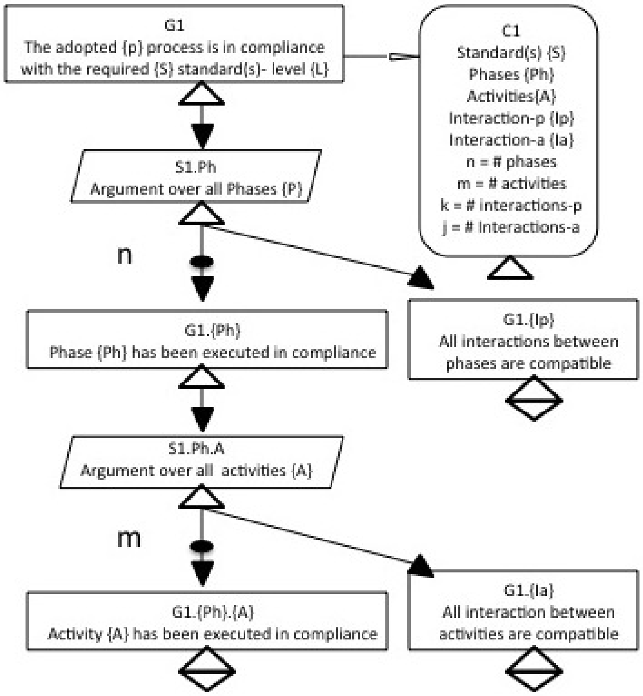


Figure 2: Goal structure representing the *Process compliance* argumentation pattern.

Table 2: Concepts mapping

SPEM 2.0	**GSN**	**SACM 2.0**
Task *ta*	Goal	Claim
Role *ro*	Solution	AEC
Work product *wp*	Solution	AEC
Tool *to*	Solution	AEC
Guidance *gu*	Solution	AEC

be further enriched to take into consideration the pattern-related modeling elements (from SPEM 2.0 capability patterns to SACM 2.0-patterns to be concretized via GSN goal structures). However, given the still unstable status of SACM 2.0, we prefer avoiding the inclusion of a premature and potentially hazardous mapping.

Concerning the rules, which were presented by Gallina 2014 [7] and Gallina et al 2016 [9], their validity is still confirmed since they were formulated considering directly GSN modeling elements.

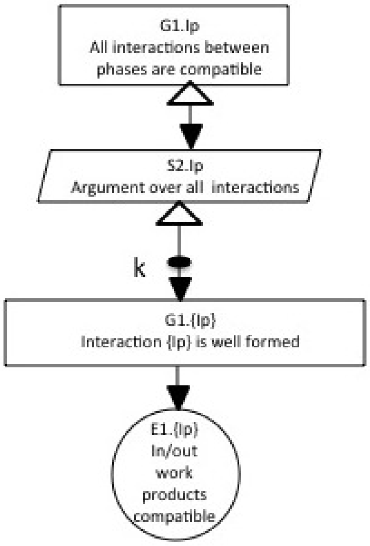

G1.Ip
All interactions between phases are compatible
S2.Ip
Argument over all interactions
k
G1.{Ip}
Interaction {Ip} is well formed
E1.{Ip}
In/out work products compatible

Figure 3: Goal structure, continuation of the pattern in Fig. 2.

Concerning tool-support for MDSafeCer, in the context of SafeCer [28], a prototype implementation of MDSafeCer was integrated within Workflow Engine for Analysis, Certification and Testing (WEFACT), which is a tool that offers a flexible infrastructure for defining and executing processes as well as integrating other tools for rendering purposes. More recently, another implementation in compliance with the SACM1.1 stable version was developed by Alajrami et al. 2016 [29]. However, this implementation does not take in input SPEM 2.0 models but EXE-SPEM [21] models since it was conceived for processes expected to be enacted on the Cloud.

3. Safety and Performance Methodology in SPEM 2.0

To partly act as safety and quality management evidence, needed for process assessment, MBA must be the result of the selection and composition of process elements that can be considered compliant with respect to the CENELEC series. MBA is a methodology to be used at design phase. Thus, first, it should be aligned with the Software Architecture & Design Phase. As recalled in Section 2, according to the CENELEC EN 50128, this phase should be carried out by appropriate roles, according to specific guidelines, be constituted of specific tasks, consume and produce specific work products.

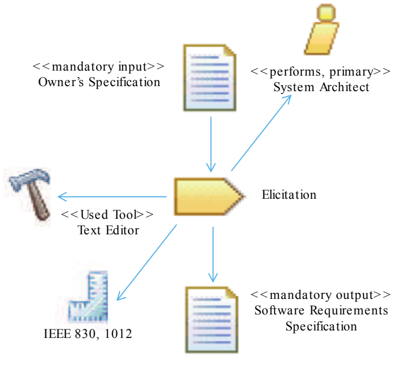
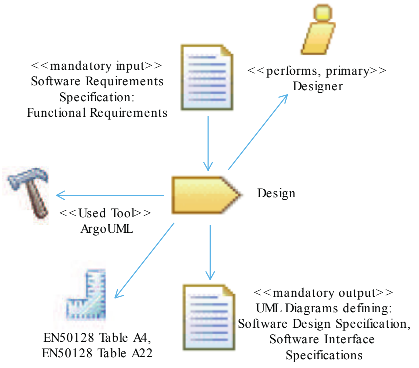
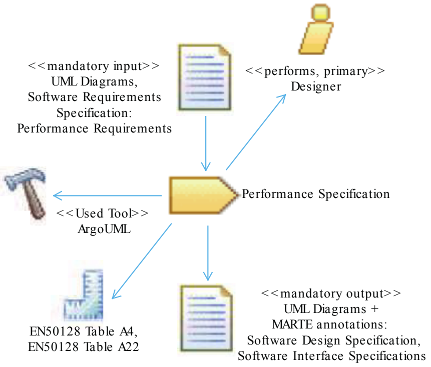
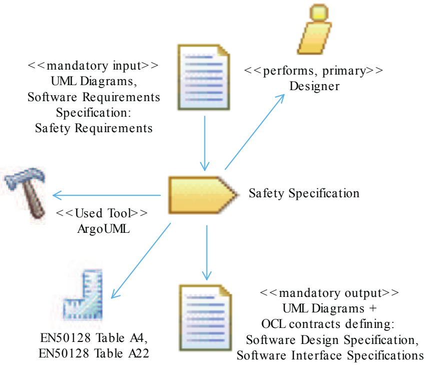
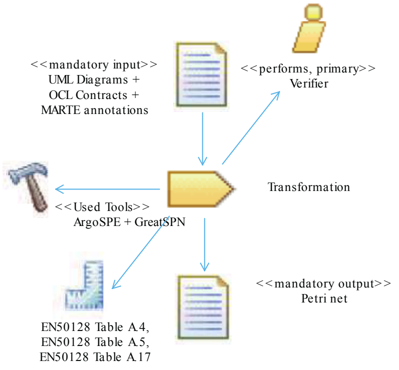
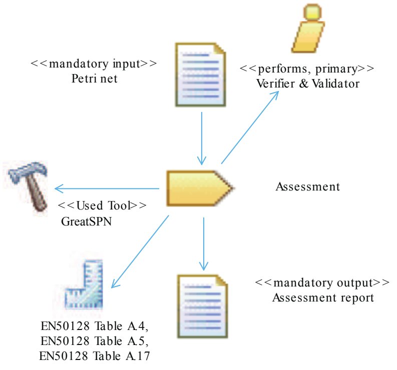
Since, as recalled in Section 2, MBA contains some of the required elements, its compliance can be partially argued about. More specifically, the following list highlights the process elements that meet the EN 50128 requirements: all the tasks that compose MBA can be considered aligned with he Software Architecture & Design Phase. However, not all the required tasks are included in MBA. This means that a company should be aware about what else should be performed. The task Transformation is not included in EN 50128 as a standalone task. It is implicitly expected to be executed (manually or automatically) in the case of usage of formal methods within the verification task. Also the current sets of MBA in/out work products can be aligned. However, the EN 50128 expected number of in/out work products is greater. MBA guidelines can be aligned. As seen in the background formal methods and more specifically Petri nets are among the techniques suggested to perform verification. With respect to roles, MBA does not pose enough emphasis. Nothing about qualifications is defined. Finally, the current tools (e.g., translator, model checker, etc.) that are proposed to perform the tasks do not offer satisfying evidence concerning their quality. Thus, MBA as it is cannot be adopted in real settings.

To enable its usage in real settings, the presentation of MBA should be enhanced and its alignment clearly made explicit. More specifically, all input/output work products should be specified and aligned with EN 50128. Concerning roles, vagueness in terms of their responsibility and degree of independence should be eliminated. Concerning tools, rational and adequate justifications in terms of their quality should be provided. In alternative, other tools should be suggested. In Table 3, we illustrate the SPEM 2.0 models representing the augmented MBA tasks. For sake of clarity is should be noted that the usage of SPEM2.0 could be considered arguable given its lack of support for process enactment as well as for modeling explicitly executed processes (e.g., start/end dates as mentioned in the background). In the context of our work, however, this limitation is not crucial since enactment is not required. Moreover, missing information could eventually be added in the description field of the specific process elements.

The augmented MBA, for instance, also considers the elicitation task, denoted as Task-0, according to standards IEEE 830 [19] and IEEE 1012 [20]. Thus, the augmented MBA is constituted of five plus one flattened tasks.

More specifically, Task-0 is carried out by the System Architect, who defines the owner's specification of the software product based on his/her personal expertise. This phase also encompass non-functional properties

Table 3: MBA tasks given in SPEM 2.0

Task-0	Task-1
 <<mandatory input>> Owner's Specification <<performs, primary>> System Architect Elicitation <<Used Tool>> Text Editor IEEE 830, 1012 <<mandatory output>> Software Requirements Specification	 <<mandatory input>> Software Requirements Specification: Functional Requirements <<performs, primary>> Designer Design <<Used Tool>> ArgoUML EN50128 Table A4, EN50128 Table A22 <<mandatory output>> UML Diagrams defining: Software Design Specification, Software Interface Specifications
Task-2	
 <<mandatory input>> UML Diagrams, Software Requirements Specification: Performance Requirements <<performs, primary>> Designer Performance Specification <<Used Tool>> ArgoUML EN50128 Table A4, EN50128 Table A22 <<mandatory output>> UML Diagrams + MARTE annotations: Software Design Specification, Software Interface Specifications	 <<mandatory input>> UML Diagrams, Software Requirements Specification: Safety Requirements <<performs, primary>> Designer Safety Specification <<Used Tool>> ArgoUML EN50128 Table A4, EN50128 Table A22 <<mandatory output>> UML Diagrams + OCL contracts defining: Software Design Specification, Software Interface Specifications
Task-3	Task-4
 <<mandatory input>> UML Diagrams + OCL Contracts + MARTE annotations <<performs, primary>> Verifier Transformation <<Used Tools>> ArgoSPE + GreatSPN EN50128 Table A.4, EN50128 Table A.5, EN50128 Table A.17 <<mandatory output>> Petri net	 <<mandatory input>> Petri net <<performs, primary>> Verifier & Validator Assessment <<Used Tool>> GreatSPN EN50128 Table A.4, EN50128 Table A.5, EN50128 Table A.17 <<mandatory output>> Assessment report

requirements that the system must meet, including safety, performance or security. Considering only those functional requirements from the software specification, the designer conceives the software design using the ArgoUML tool. This task produces a set of UML diagrams defining the software specification. Task-2 is divided into two subtasks, each of them focusing on each of the non-functional requirements: performance and safety. The separation is due to the fact that each of them uses different specification languages: performance requirements are described using MARTE annotations, and safety, OCL contracts. As outcome, the aforementioned UML diagrams are augmented with MARTE annotations and Object Constraint Language (OCL) contracts. These UML diagrams are then transformed into Petri nets by the Verifier, an expert in formal methods, during Task-3. Finally, Task-4 assesses the initial safety and performance requirements with the results obtained in the evaluation of the Petri Net.

By construction, these augmented MBA tasks contain process elements that are in compliance with EN 50128. To explain this compliance, in Section 4 we derive process-based arguments and we document them in GSN. Besides the enhancement of the presentation, to satisfy all the EN 50128, MBA should, however, be further developed or combined with another methodology offering complementary support. Thus, given the awareness developed thanks to the performed gap analysis, we also indicate the undeveloped goals.

4. Arguing about EN 50128 Compliance via MDSafeCer

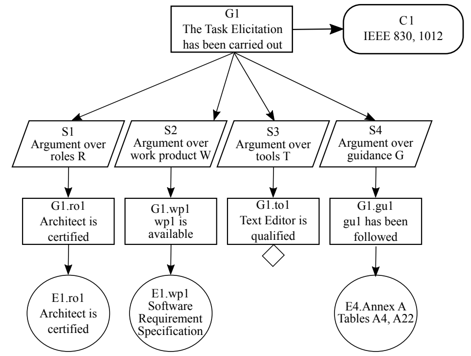
The aim of this section is to derive a process-based argument for arguing about MBA compliance with EN 50128. More specifically, our derived argument given in GSN argues that MBA is partially compliant with the EN 50128 requirements related to the design phase for a SIL2 subsystem. To derive such argument, we proceed compositionally and from the process models given in Table 3, by using MDSafeCer, we first derive sub-arguments that argue about compliance at task level. The derived sub-arguments are depicted in Table 4. Such arguments could be further developed to indicate the missing evidence (e.g., the missing work products).

By aligning MBA-hierarchy with the pattern hierarchy (presented in Section 2.4.2) and by manually following the generation rules, we can easily derive the argument at the phase level, depicted in Fig. 4 and in Fig. 5 (note that Fig. 5 does not present all the developed goals related to all the relations among tasks).

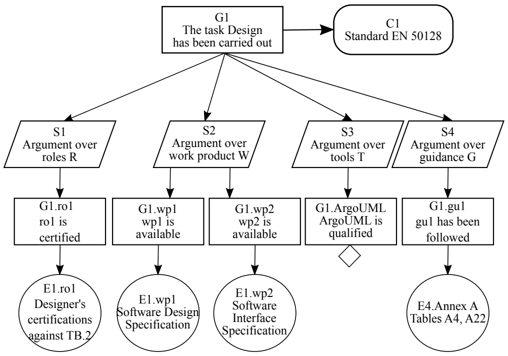
This argument can be easily composed with the sub-arguments, which were illustrated in Table 4. The compositional nature could be presented

Table 4: Task-based arguments

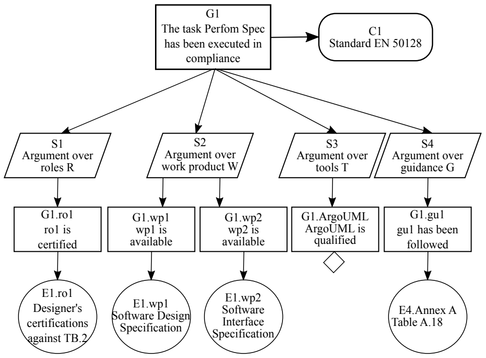
T0-based argument

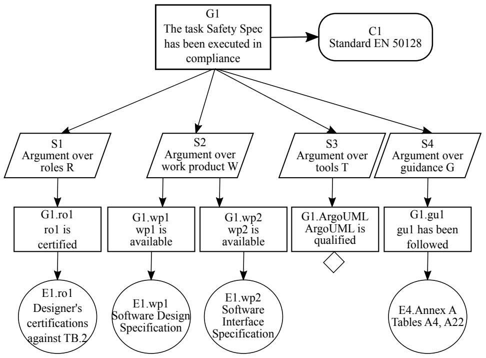


T1-based argument

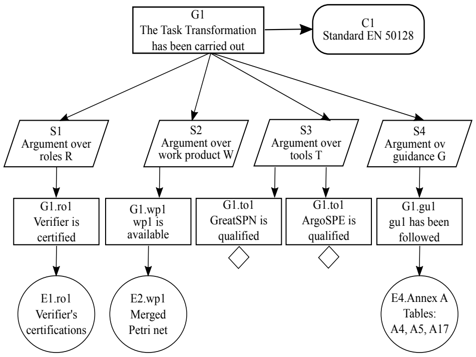


T2-based argument

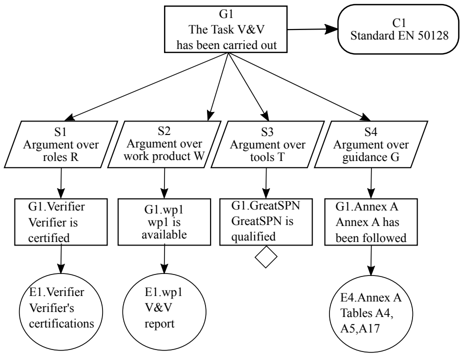




T3-based argument



T4-based argument



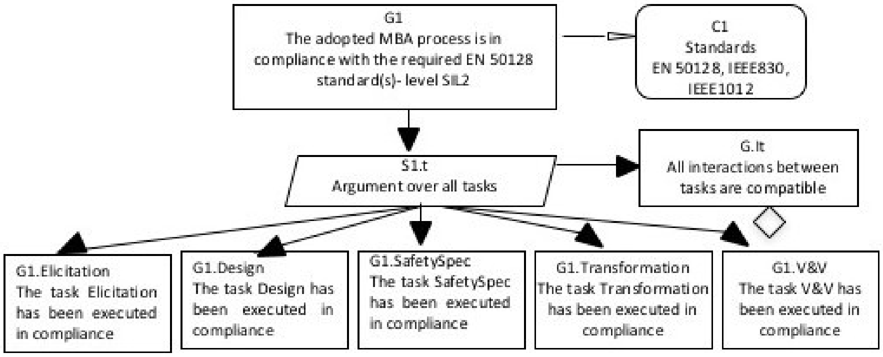

G1
The adopted MBA process is in compliance with the required EN 50128 standard(s)- level SIL2
C1
Standards
EN 50128, IEEE830, IEEE1012
S1.t
Argument over all tasks
G.It
All interactions between tasks are compatible
G1.Elicitation
The task Elicitation has been executed in compliance
G1.Design
The task Design has been executed in compliance
G1.SafetySpec
The task SafetySpec has been executed in compliance
G1.Transformation
The task Transformation has been executed in compliance
G1.V&V
The task V&V has been executed in compliance

Figure 4: Goal structure representing the argumentation pattern instance.

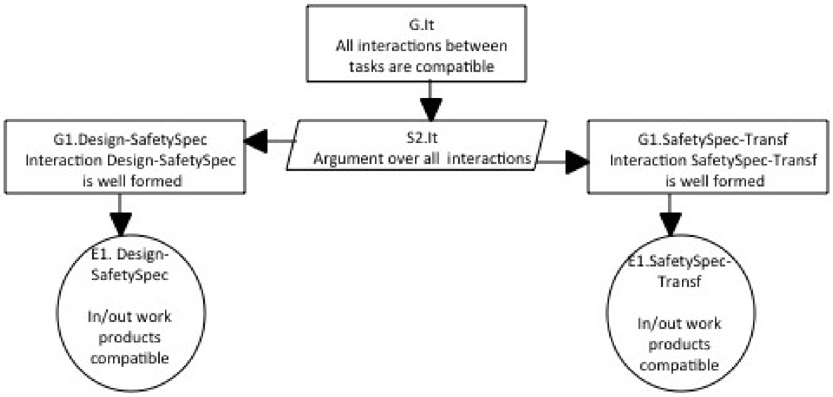

G.It
All interactions between tasks are compatible
S2.It
Argument over all interactions
G1.Design-SafetySpec
Interaction Design-SafetySpec is well formed
G1.SafetySpec-Transf
Interaction SafetySpec-Transf is well formed
E1. Design-SafetySpec
In/out work products compatible
E1.SafetySpec-Transf
In/out work products compatible

Figure 5: Goal structure, continuation of the argumentation pattern instance.

in a more advanced way by using modularized goal structures. Similarly, contracts could be used to clearly state the assumptions and guarantees that may exist between two sub goal-structures. In the context of distributed and heterogeneous management, where the responsibility for the provision of the different justifications might also be distributed and then integrated, contract-based goal structuring could be a winning solution.

5. Discussion

As previously stated, the derived argument given in GSN in Section 4 argues that MBA is partly compliant with the EN 50128 requirements related to the design phase for a SIL 2 subsystem. The part which is not compliant is shown as undeveloped goals in Table 4: the tools, used in MBA, which are not qualified. The main reason why they are not qualified is that they are the result of research carried out by academia and developed as proof of concept.

One possible way of addressing this limitation is to submit the tools to a qualification process. In [30] a qualification method that takes a systems approach on qualifying software tools as parts of tool chains is suggested. The idea is to focus on the relevant parts of tool chains in regard to safety in relation with the development environment they are to be deployed in. Thus, by following this approach, all the tools used in MBA could be qualified as a tool chain. A different approach is given in [31], where the experiences of qualifying one stand-alone tool according to ISO/DIS 26262 standard are described. The authors explain that ISO 26262 allows for different levels of qualification, including a self-qualification (1st party qualification). However, to increase confidence into the proposed approach, they decided to submit the tool qualification approach to an accredited certification body for review and approval. Due to their reputation for software tool certifications / qualifications according to various standards, TÜV SÜD Automotive GmbH was chosen for the tool qualification assessment. We could follow a similar approach and choose the railway company Grupo CAF, S.A., with whom we have worked previously, to assist with the tool qualification assessment, which would be carried out by AENOR, the Spanish Association for Standardization and Certification.

Another possibility is to substitute the aforementioned tools by qualified tools. To the best of our knowledge, there are no qualified tools that can be directly substituted in the MBA methodology. A possible solution is to modify the MBA methodology in such a way that there exist qualified tools to support the underlying formalism.

6. Related Work

The current certification framework is traversing a crisis phase due to the growing complexity associated to the safety justifications that are required by the standards [32]. A balance between process and product-based justification is still not clear. Despite its necessity, process-based justification is proportionally less investigated. Bender et al [33] in their work on the certification nature, conclude that for the time being process adherence (including personal qualifications), classified as indirect evidence, must be provided. They however do not propose any process-related argument. Nair et al. [34] recognise the relevance of process-based argumentation and similarly to what proposed by Gallina [7] argue about the core process elements. Nair et al. call the process-based argument as secondary confidence argument. The work from Nair et al. is further developed by Hawkins et al. [35], who, differently from Gallina [7], uses a process-related meta-model, which was developed within the OPENCOSS project [36], as part of the OPENCOSS Common Certification Language. To enable pattern-based model-driven certification in military aeronautical settings, the MIMOSA (Means of engIneering for MOdelling and analysis of modular embedded aeronautic Systems and Architectures) framework [37] is proposed. This framework introduces argumentation patterns, given in a new notation, which highly borrows from GSN. The main goal of MIMOSA is to contribute to the assessment of the architectures proposed by industrial companies with respect to certification standards. MIMOSA is not focused on the design processes but on their outcomes.

As related work, we also mention the recent work presented by Knight et al. [38]. In their effort aimed at seeking a rational within the currently proposed standards, the authors also contribute to the identification of a possible solution for the certification crisis. Instead of developing process-based arguments to be used by manufacturers, they develop process-based arguments to be used by standardisation bodies to justify their sets of requirements ("an explicit rationale that justifies the contents of the standard").

In its effort aimed at strengthening via process-based evidence an existing method that targets provision of product-based evidence, our work represents a novelty. Our work also contributes to the achievement of the right balance between process and product based justification. Moreover, the possibility, in a long term, of deriving semi-automatically process-based arguments related to MBA will free time to be dedicated to the provision of product-based arguments. This possibility will also allow manufacturers to save money during the negotiation process with certification bodies since rational and explanations concerning process-based solutions will be available.

Given the relevance of providing process-based arguments in order to strengthen existing methods, a similar effort was conducted within the avionic domain by Gallina et al. [39] by arguing against model-based development-related requirements stated in DO-178C [40]/DO-331 [41]. The focus of the derivation is on the planning phase, while in this paper the focus is on the execution phase. Another effort related to process assessment was recently conducted in the nuclear domain [42], where authors propose a mapping between SPICE (Software Process Improvement and Capability Determination)-oriented process assessment and argumentation modeling elements. This work is not rooted in state of the art methodological solutions based on meta-modelling principles as ours is. It, however, provides an interesting SPICE-oriented perspective.

7. Conclusion and Future Work

Since newly proposed and potentially innovative engineering methods suffer of low acceptance in rail industrial settings due to the requirements of the certification process, methods aimed at speeding up the provision of process-based arguments can be beneficial. In this paper, we have used MDSafeCer to show that MBA can be partly used as quality and safety management evidence within a safety case. More specifically, we have focused on specific portions of the CENELEC standard series related to software process compliance and we have argued by using GSN about compliance with EN 50128-related design with special focus on safety and performance requirements.

In the future, two lines of further development are envisioned: one aimed at empowering MBA and the other aimed at empowering MDSafeCer. Concerning the first line of future development, to empower MBA we aim developing more compelling arguments aimed at sustaining that safe processes will lead to safe software, more specifically, safe design processes will lead to safe design. As known [43], providing such arguments is a challenging task.To achieve a more compelling process argument related to the execution of the MBA, we will further develop MBA according to the findings. Ideally, all undeveloped goals should be replaced by well-founded and explained goals. Moreover, similar to the work by Varkoi et al. 16, which was discussed in Section 6, we are also interested in expanding our argumentation by using SPICE-oriented assessment. To do this, similarly to what done in this paper with EN 50128, we will analyse SPICE to evaluate MBA's maturity. Moreover, we are also interested in developing arguments concerning the planning phase in order to argue about fulfilment of the requirements related to the preparation of Software Quality Assurance Plan (aligned with the Safety Plan at system level).

As future work, we also aim at focusing on evidence related to the system/subsystem behaviour, i.e., technical evidence. To do that, we plan to derive product-based arguments by building on top of work presented by

Sljivo et al. [44]. Moreover, considering the ongoing development of standards aimed at providing guidance concerning cybersecurity (see for instance the IEC 62443 series of standards, which target industrial communication networks - Network and system security), MBA could be further evaluated against them in order to strengthen it and determine its capability in designing *security-informed* [45] safety-critical systems. To do that, we plan to build on top of the work aimed at aligning safety and security standards (e.g., in the avionics [46] and in the railway [47]) in order to avoid duplicating the effort.

Concerning the second line of future development, we are interested in empowering MDSafeCer's tool-support. To do that, we plan to make evolve the current implementations. The initial goal of their evolution is to provide evidence with respect to the effectiveness of the approach in terms of time reduction (manual vs. semi-automatic work). Once the evidence is achieved, the intention is to provide an industry-friendly tool support.

Acknowledgements

This work has been partially supported by the ARTEMIS project nSafeCer [28], the Gen&ReuseSafetyCases project [48], by the EU and VINNOVA via the ECSEL JU project AMASS (No 692474) [14]. The research of Clara Benac Earle was also supported by Spanish MINECO project STRONGSOFT (TIN2012-39391-C04-02) and by the Madrid Regional Government project nGreens (S2013/ICE-2731).

References

[1] N. Leveson, White Paper on The Use of Safety Cases in Certification and Regulation (2012).

[2] J. Rushby, Logic and epistemology in safety cases, in: SafeComp 2013: Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security, Vol. 8153 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 1–7.

[3] Rail Safety and Standards Board, Engineering safety management, in: The Yellow Book, Rail Safety and Standards Board on behalf of the UK rail industry, 2012, p. 361.

[4] E. Gómez-Martínez, R. J. Rodríguez, C. Benac Earle, L. Etxeberria, M. Illarramendi, A Methodology for Model-based Verification of Safety Contracts and Performance Requirements, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliabilitydoi:10.1177/1748006X16667328.

[5] E. Gómez-Martínez, R. J. Rodríguez, L. Etxeberria Elorza, M. Illarramendi Rezabal, C. Benac Earle, Model-based verification of safety contracts, in: C. Canal, A. Idani (Eds.), Software Engineering and Formal Methods, Vol. 8938 of Lecture Notes in Computer Science, Springer, 2015, pp. 101–115.

[6] Object Management Group, Software & Systems Process Engineering Meta-Model (SPEM), v2.0. Full Specification formal/08-04-01 (2008).

[7] B. Gallina, A model-driven safety certification method for process compliance, in: 2nd Int. Workshop on Assurance Cases for Software-intensive Systems (ASSURE), 2014, pp. 204–209.

[8] B. Gallina, L. Provenzano, Deriving Reusable Process-based Arguments from Process Models in the Context of Railway Safety Standards, AUJ 36 (4).

[9] B. Gallina, E. Gómez-Martínez, C. B. Earle, Deriving Safety Case Fragments for Assessing MBASafe's Compliance with EN 50128, in: Proceedings of 16th International Conference on Software Process Improvement and Capability Determination SPICE 2016, Vol. 609 of Communications in Computer and Information Science, Springer, 2016, pp. 3–16.

[10] Interim Defence Standard 00-56 Part 1 - Issue 5, in, UK MOD (2014).

[11] GSN, Goal Structuring Notation Community Standard Version 1 (2011).

[12] SACM1.1, Structured Assurance Case Metamodel (SACM).
URL http://www.omg.org/spec/SACM/1.1

[13] SACM2.0, Structured Assurance Case Metamodel (SACM).
URL http://www.omg.org/spec/SACM/2.0/Beta1/

[14] AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems), http://www.amass-ecsel.eu.

[15] MODSafe Modular Urban Transport Safety and Security Analysis, Survey of current safety lifecycle approaches, DEL D6.1 TRIT WP6 100531 V1.0, Tech. rep., European Union (2010).

[16] BS EN50126, Railway applications: The specification and demonstration of Reliability. Availability, Maintainability and Safety (RAMS) (1999).

[17] BS EN50129, Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling (2003).

[18] BS EN50128, Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems (2011).

[19] IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications, Tech. rep., Institute of Electrical and Electronics Engineers (Oct. 1998). doi:10.1109/ieeestd.1998.88286.

[20] IEEE Computer Society, IEEE Standard for Software Verification and Validation, Tech. rep., Institute of Electrical and Electronics Engineers (Oct. 2004).

[21] S. Alajrami, B. Gallina, A. Romanovsky, EXE-SPEM: towards cloud-based executable software process models, in: MODELSWARD, SciTePress, 2016, pp. 517–526.

[22] B. Gallina, K. R. Pitchai, K. Lundqvist, S-TunExSPEM: Towards an Extension of SPEM 2.0 to Model and Exchange Tunable Safety-oriented Processes, in: R. Lee (Ed.), Software Engineering Research, Management and Applications, Springer International Publishing, Heidelberg, 2014, pp. 215–230.

[23] A. Söderberg, R. Johansson, Safety Contract Based Design of Software Components, in: IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2013, pp. 365–370. doi:10.1109/ISSREW.2013.6688922.

[24] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, Wiley Series in Parallel Computing, John Wiley and Sons, 1995.

[25] E. Gómez-Martínez, J. Merseguer, ArgoSPE: Model-based Software Performance Engineering, in: Proc. 27th Int. Conf. on Applications and Theory of Petri Nets and Other Models of Concurrency (ICATPN), Vol. 4024 of Lecture Notes in Computer Science, Springer, 2006, pp. 401–410.

[26] T. S. Liu, S. B. Chiou, The application of Petri nets to failure analysis, Reliab. Eng. Syst. Safe. 57 (2) (1997) 129–142. `doi:10.1016/S0951-8320(97)00030-6`.

[27] S. Baarir, M. Beccuti, D. Cerotti, M. De Pierro, S. Donatelli, G. Franceschinis, The GreatSPN tool: recent enhancements, SIGMETRICS Perform. Eval. Rev. 36 (4) (2009) 4–9.

[28] ARTEMIS-JU-269265, SafeCer-Safety Certification of Software-Intensive Systems with Reusable Components, `http://www.safecer.eu/`.

[29] S. Alajrami, B. Gallina, I. Sljivo, A. Romanovsky, P. Isberg, Towards cloud-based enactment of safety-related processes, in: SafeComp 2016: Proceedings of the 35th International Conference on Computer Safety, Reliability, and Security, Vol. 9922 of Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 309–321.

[30] F. Asplund, J. El-khoury, M. Törngren, Qualifying Software Tools, a Systems Approach, in: SafeComp 2012: Proceedings of the 31st International Conference on Computer Safety, Reliability, and Security, Vol. 7612 of Lecture Notes in Computer Science, Springer, 2012, pp. 340–351. `doi:10.1007/978-3-642-33678-2\_29`.

[31] M. Conrad, P. Munier, F. Rauch, Qualifying Software Tools According to ISO 26262, in: Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme VI, Schloss Dagstuhl, Germany, 2010, Tagungsband Modellbasierte Entwicklung eingebetteter Systeme, fortiss GmbH, München, 2010, pp. 117–128.

[32] B. Gallina, How to increase efficiency with the certification of process compliance. In the 3rd Scandinavian Conference on SYSTEM & SOFTWARE SAFETY, Stockholm, March 24-25 (2015).

[33] M. Bender, T. Maibaum, M. Lawford, A. Wassyng, Positioning verification in the context of software/system certification, in: 11th International Workshop on Automated Verification of Critical Systems (AVOCS), Newcastle upon Tyne (UK), Sept. 12-15, 2013, pp. 1–15.

[34] S. Nair, N. Walkinshaw, T. Kelly, J. L. de la Vara, An evidential reasoning approach for assessing confidence in safety evidence, in: IEEE 26th International Symposium on Software Reliability Engineering (ISSRE), 2015, pp. 541–552. doi:10.1109/ISSRE.2015.7381846.

[35] R. Hawkins, T. Richardson, T. Kelly, Using process models in system assurance, in: SafeComp 2016: Proceedings of the 35th International Conference on Computer Safety, Reliability, and Security, Vol. 9922 of Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 27–38.

[36] FP7 OPENCOSS, Open Platform for EvolutioNary Certification Of Safety-critical Systems, http://www.opencoss-project.eu.

[37] P. Bieber, F. Boniol, G. Durrieu, O. Poitou, T. Polacsek, V. WIELS, G. Martinez, MIMOSA: Towards a model driven certification process, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), TOULOUSE, France, 2016, pp. 131–139.
URL https://hal.archives-ouvertes.fr/hal-01289704

[38] J. C. Knight, J. Rowanhill, The indispensable role of rationale in safety standards, in: SafeComp 2016: Proceedings of the 35th International Conference on Computer Safety, Reliability, and Security, Vol. 9922 of Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 39–50.

[39] B. Gallina, A. Andrews, Deriving Verification-related Means of Compliance for a Model-based Testing Process, in: Proceedings of the 35rd IEEE Digital Avionics Systems Conference, DASC, 2016.

[40] DO-178C (EUROCAE ED-12C), Software Considerations in Airborne Systems and Equipment Certification, RTCA Inc, Washington DC (2013).

[41] DO-331, Model-Based Development and Verification Supplement to DO-178C and DO-278A, RTCA Inc, Washington DC (2011).

[42] T. Varkoi, T. Mäkinen, R. Nevalainen, Process Assessment in A Safety Domain, in: Proceedings of the 10th International Conference on the Quality of Information and Communications Technology- Track: Quality Aspects in Safety Critical Systems, QUATIC, 2016.

[43] O. Nordland, Presenting a safety case — a case study —, in: SafeComp 2001: Proceedings of the 20th International Conference on Computer Safety, Reliability, and Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 56–65.

[44] I. Sljivo, B. Gallina, J. Carlson, H. Hansson, Generation of Safety Case Argument-Fragments from Safety Contracts, in: 33rd International Conference on Computer Safety, Reliability, and Security, Vol. 8666 of Lecture Notes in Computer Science, Springer, 2014, pp. 170–185.

[45] R. Bloomfield, P. Bishop, Safety and assurance cases: Past, present and possible future–an Adelard perspective, in: Making Systems Safer, Springer, 2010, pp. 51–67.

[46] B. Gallina, L. Fabre, Benefits of security-informed safety-oriented process line engineering, in: 34th Digital Avionics Systems Conference (DASC), 2015, pp. 8C1–1–8C1–9.

[47] J. Braband, What's Security Level got to do with Safety Integrity Level?, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), TOULOUSE, France, 2016, pp. 410–419.
URL https://hal.archives-ouvertes.fr/hal-01289437

[48] Gen&ReuseSafetyCases-SSF,
http://www.es.mdh.se/projects/393-GenReuseSafetyCases.