Deriving Reusable Process-based Arguments from Process Models in the Context of Railway Safety Standards

Barbara Gallina
Mälardalen University, P.O. Box 883, SE- 721 23 Västerås, Sweden;
barbara.gallina@mdh.se
Luciana Provenzano
Bombardier Transportation, Östra Ringvägen 2, 722 14 Västerås, Sweden;
luciana.provenzano@rail.bombardier.com

Abstract

In the railway domain, standards such as the EN5012x family prescribe processes to be followed for the management and certification of safety-critical systems. This results in a need to model processes and retrieve process-based arguments to prove that the system achieved the required safety level in order to reduce time and cost spent in the certification process. In this paper, we present the application of the MDSafeCer, i.e. a model-driven safety certification method, for railways. In particular, we model in SPEM 2.0 the safety requirements process according to what described in the safety plan, and we show how it is possible to extract safety evidence to prove the compliance of this process to the EN50128 standard.

Keywords: railway, safety certification, process modelling

1 Introduction

In the context of safety-critical railway systems engineering, various standards (i.e. EN5012x) play a crucial role in prescribing process reference models at system (i.e. EN50126 [5]) as well as at sub-system level (i.e. EN50128 [6]). These models define sets of partially ordered tasks that have to be executed to develop safety-critical railway systems (such as entire vehicle, signalling components, etc.). As also observed in the automotive domain [4], to these partially ordered tasks other core process elements are directly or indirectly associated namely roles, work-products, and guidelines. These core process elements allow process engineers to establish responsibilities by defining roles (who) for producing specified work products (what). Moreover, for the execution of the tasks well-defined principles and techniques supported by guidelines are applied. The rigor and stringency required during the application of these reference models vary with respect to the criticality of the systems, and are subject to interpretations. Compliance with the process reference models constitutes a mandatory requirement for certification purposes in which process-related deliverables are fundamental. Within EN50129 [7], a safety case is defined as a structured justification document that includes the required evidence, i.e. evidence of quality management, evidence of safety management (compliance with the EN50126 RAMS process [5]), and evidence of functional and technical safety. Evidence of quality as well as safety management represents process-related evidence. The provision of such evidence is time-consuming and costly, especially if reuse [3] and semi-automatic generation is not enabled.

To reduce time and cost, we apply MDSafeCer, which was introduced by Gallina [2] in the context of the SafeCer project [1] SYNOPSIS [14]. MDSafeCer is a model-driven certification method for the (semi) automatic generation of process-related deliverables. In this paper we consider a portion of the safety plan, we model it in SPEM (Software Process Engineering Meta-model) 2.0 [11], and then we show how process-based fragments in form of GSN (Goal Structuring Notation)-compliant goal structures [9] of a safety case can be derived from the safety plan model.

The remainder of this paper is organized as follows. Essential background information is recalled in Section 2. The application of MDSafeCer to the safety-requirements process defined within a railway project is described in Section 3. Concluding remarks and perspective for future work are presented in Section 4.

2 Background

In this section, we shortly recall some background on which this work is based. In particular, in Section 2.1, we provide a quick survey of the CENELEC EN5012x family of European standards applicable for the management and certification of safety-critical railway systems. In section 2.2 we recall the main SPEM 2.0 process elements that will

be used further in this paper to model the safety requirements process. In section 2.3, we briefly introduce the GSN graphical notation used to build the safety case fragment. Finally, in section 2.4 we introduce MDSafeCer method that we will apply in the railway domain.

2.1 EN5012x standards

The European group of standards EN5012x defines processes that enable the implementation of a consistent approach for the management of safety-critical railway systems. The three main standards are:

- EN50126, which describes a process for the specification and demonstration of the RAMS (Reliability, Availability, Maintainability, Safety) requirements [5]
- EN50129, which defines a process for safety acceptance and approval [7]
- EN50128, which focuses on processes for the development, deployment and maintenance of safety-related software for railway control and protection applications [6].

Figure 1 shows the scope of the above-mentioned standards compared to the railway product or system under development and/or maintenance.

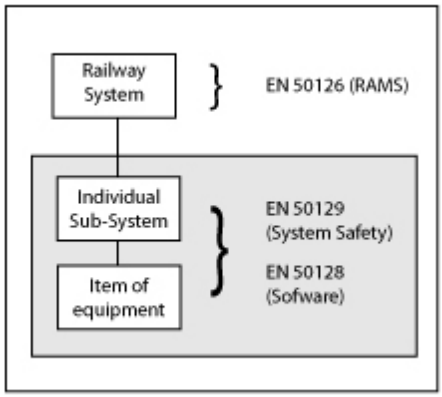

Figure 1 Scope of the EN5012x standards

In order to obtain the safety approval for a given safety-critical railway system or product, the EN50129 standard prescribes that an independent safety assessment is performed based on documentary evidence. The documentary evidence includes the so-called Safety Case, i.e. the documented demonstration that the product complies with the specified safety requirements [7]. The Safety Case addresses the conditions that shall be satisfied to prove that the necessary level of safety has been achieved, i.e. evidence of quality management, evidence of safety management, and evidence of functional and technical safety.

The Software Requirement Phase is part of the life-cycle model (Figure 2) required by the Software Quality Assurance activities described in chapter 6.5.4.5 of the EN50128 standard [6]. In particular, the standard states that quality concerning the life-cycle model shall address as a minimum:

- activities and elementary tasks consistent with the plans, e.g. Safety Plan, that have been established at the System level;
- entry and exit criteria of each activity;
- inputs and outputs of each activity;
- major quality activities;
- the entity responsible for each activity

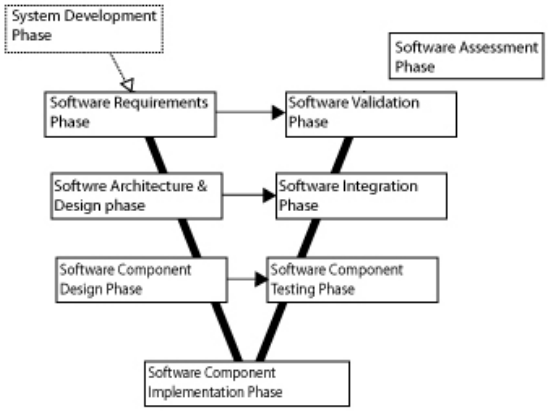

Figure 2 Life-cycle phases for a development project extracted from the V-model defined in the EN50128

Moreover, section 7.2 “Software Requirements” of the same standard defines the artifacts that shall be produced at the end of the Software Requirements Phase, i.e.:

- Software Requirements Specification
- Overall Software Test Specification
- Software Requirements Specification Report

By reading these recommendations, it is clear that a process shall be defined which complies with the standard.

2.2 Process modelling through SPEM 2.0

SPEM 2.0 [11] is the OMG standard for systems and software process modelling. Despite it is a general-purpose language, its elements implicitly enable to model safety concerns, as explained by Gallina et al. in [3] and [4].

The following table (Table 1) shows a subset of SPEM 2.0 modelling elements, in particular the ones we will use in Section 3 to model task, roles, guidance, tools and work-products related to the safety requirements process.

Task	TaskUse	Role	WorkProduct	Tool	Guidance
					

Table 1 Icons denoting method content (use) elements

2.3 Safety case documentation

As summarized by Dardar et al. [12], a safety case can be documented in textual or graphical languages (refer to [8]). GSN [9] is a graphical notation that allows organizing the

safety argumentation into flat or hierarchically nested graphs called goal structures. Figure 3 shows the syntax of the core GSN modelling elements that we will use in Section 3.

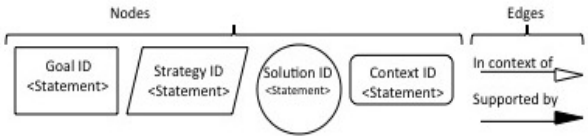

Figure 3 Subset of GSN concrete syntax

SACM (Structured Assurance Case Metamodel) [13] is an OMG standard that represents an effort to unify and standardize the graphical notations, namely GSN and CAE (Claim Argument Evidence) [10], broadly used for documenting safety cases. By providing a meta-model that defines the abstract syntax of a unified argumentation language, SACM thus constitutes a step towards the formalization of these notations.

2.4 Model-Driven Safety Certification

In this subsection we recall essential information on the Model-Driven Safety Certification (MDSafeCer) method [2]. MDSafeCer allows the (semi) automatic generation of process-based evidence from process models. MDSafeCer consists of three iterative tasks in succession, as shown in Figure 4.

The main idea is that a process is modelled (refer to the first task "Safety-process modelling") according to the best practices and the applicable standards. Once the process model is ready, a process-based argument can be generated (refer to the second task "Process-based argument generation") via a model to model transformation. The generated argument may need to be rectified, resulting in iterations back to the previous tasks, and/or completed by a safety argumentation expert (refer to the third task "Process-based argument Check&Completition") [2].

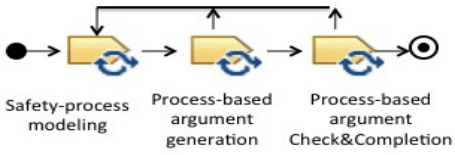

Figure 4 MDSafeCer overview specified in SPEM 2.0

3 EN50128-compliance via MDSafeCer

In this section, we apply MDSafeCer in the context of safety-critical railway systems to:

- Model the safety requirements process
- Build evidences required to prove the compliance of this process with what described in chapter 7.2 "Software Requirements" of the EN50128:2011 railway standard. These evidences will compose the Safety Case, as explained in Section 2.1.

In particular, we model the process of building the Software Requirements Specification artifact within the Software Requirements Phase.

3.1 Safety-requirements process modelling

The first step is to model in SPEM 2.0 the task concerning the writing of the Software Requirements Specification document (refer to task "Safety-process modeling" in Section 2.4). This task shall be compliant with chapter 7.2 of the EN50128 standard.

To perform this activity, all process elements linked to this task shall be specified, as required by the SPEM 2.0 process elements recalled in Section 2.2. For example, we shall define the work-product associated to this task, the role in charge of creating and maintaining this document, and so on. This information should be described in the Project Safety Plan and/or in other project plans, such as the Quality Assurance plan, etc. that are referred in the Safety Plan. The following table (Table 2) shows the definition of the process elements related to the safety requirements process and in which project plan we find the needed information.

SPEM2.0 Process element	**Process element description**	**Information found in…**
Work product	Software Requirements Specification	Sub-chapter "Safety life-cycle" of chapter "Safety Management" within the Safety Plan
Role	Requirement Manager	Engineering Project Plan (EPP) that is referenced in sub-chapter "Roles and Responsibilities" of chapter "Safety Management" within the Safety Plan
Tool	IBM DOORS	Sub-chapter "Safety Requirements" of chapter "Safety Management" within the Safety Plan.
Guidance	Software Safety Requirement Guidelines	Requirement Management Plan that is referenced in sub-chapter "Safety Requirements" of chapter "Safety Management" within the Safety Plan

SPEM2.0 Process element	Process element description	Information found in…
Task	Software Requirements Specification	Sub-chapter "Safety life-cycle" of chapter "Safety Management" within the Safety Plan

Table 2 Process elements description

Figure 5 depicts the final result of the modeling in SPEM 2.0 of the task Software Requirements Specification.

It is worth noting that SPEM 2.0 also enables the process engineer to define via stereotypes some additional information for each process element (e.g., <<performs, primary>>). This makes possible the addition of important pieces of information, necessary to support the safety justification.

Moreover, in the case of the process element "role", it is possible to specify that the Requirements Manager's competence is substantiated through CV and course attendance certificates. These pieces of information are then included in the final justification, as shown in Section 3.2.

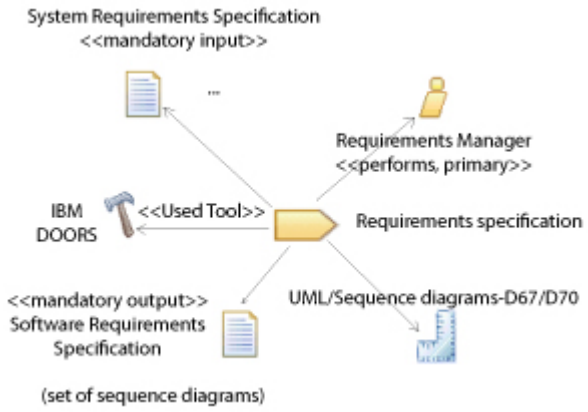

Figure 5 SPEM 2.0 modeling of Software Requirements Specification in EN50128

The same logic used to model in SPEM 2.0 the Software Requirements Specification can be applied to model the remaining work-products within the Software Requirements Phase, i.e. the Overall Software Test Specification and the Software Requirement Specification Report.

3.2 Process-based argument compliance

Based on the information defined in the model in Figure 5 and by applying the transformation rules [2], we can create the safety argumentation (refer to task "Process-based argument generation" in Section 2.4) in the GSN notation, as depicted in Figure 6.

As discussed in [8], the documentation style is a matter of taste and inclination. Text-inclined safety experts/assessors might prefer reviewing textual documentation. To satisfy text-inclined argument-readers, instead of a model to model transformation, a model to text transformation should be provided aiming at generating a safety justification in the shape of a structured prose, as shown in this example:

"This argument establishes the following claim: the task requirement specification has been planned, within the context of EN50128. To establish the top-level claim, four strategies are adopted: (1) argues about roles; (2) argues about work-products; (3) argues about guidelines; (4) argues about tools.

To argue about roles, one sub-claim is established: (1) the requirement manager is certified. This sub-claim is supported by direct evidence in form of CV and course attendance. Etc..."

The above-written text-based argumentation is equivalent to the one given in GSN.

Once the argumentation is available, it is used by a safety expert (refer to task "Process-based argument Check&Completion" in Section 2.4) as basis for creating the final document to be submitted to the authority. The safety expert may improve the confidence of an argument by adding more assumptions and justifications, modify existing goals or develop new goals, as explained in [2].

Once the safety argumentation is entirely checked and finalised, it can be used to prove the compliance of the safety-requirements process with the EN50128 standard and, as a result, included in the Safety Case.

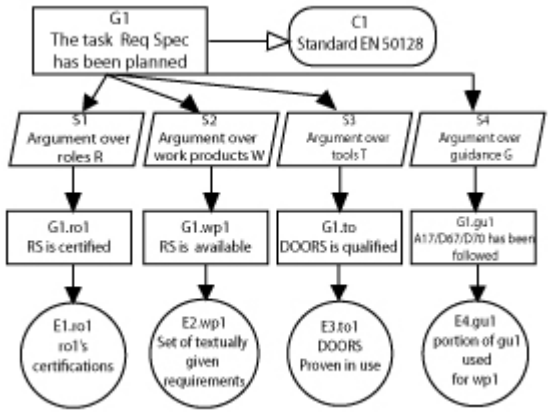

Figure 6 GSN structure arguing about process compliance

4 Conclusions and future work

In this paper, we presented the application of MDSafeCer to railway standards.

In railways, safety standards (EN5012x) prescribe processes to be followed for the management and certification of safety. This requires the definition of well-defined processes and their application and monitoring throughout the entire project life-cycle. Moreover, process-based safety arguments shall be extracted to show process compliance. These activities can be time-consuming, expensive and error-prone if not supported by a structured

process modelling and systematic reuse. For these reasons, we explored the possibility of applying MDSafeCer to model the safety requirements process and to build the process-based argument needed to show the compliance of this process with the EN50128 standard.

The first result we obtained is that MDSafeCer can be successfully used for this purpose. By drawing generalizations from this result, we can conclude that MDSafeCer can be applied to the whole life-cycle defined in EN50128. The proposed usage of SPEM2.0 resulted to be promising. SPEM2.0 can be used to model the whole life-cycle defined in EN50128 in a rather intuitive way. This outcome is also valid for all other safety railway standards.

We also observed that the use of MDSafeCer can significantly improve the process quality at very early stage of the project. In fact, MDSafeCer enables to highlight missing information about work-products, roles, responsibilities, etc. by giving an opportunity of tuning the process in the right way. To generate the argument-fragment, MDSafeCer needs to transform process elements into argumentation elements. Whenever process elements are missing MDSafeCer is expected to notify the user. From this perspective we think that MDSafeCer will reduce time and cost for the production of the safety evidence.

In the future, in cooperation with assessors, we plan to fully define a pattern for arguing about process compliance in the context of railways standards. Moreover, we also expect to automate the generation of the argument by using the prototype tool support currently available within the AIT WEFACT tool [15].

Acknowledgements

This work has been partially supported by the Swedish SSF SYNOPSIS project [14].

References

[1] ARTEMIS-JU-269265, *SafeCer - Safety Certification of Software-Intensive Systems with Reusable Components*.

[2] B. Gallina (2014), *A Model-driven Safety Certification Method for Process Compliance, 2nd IEEE International Workshop on Assurance Cases for Software-intensive Systems (ASSURE)*, joint event of ISSRE, Naples, Italy.

[3] B. Gallina, I. Sljivo, O. Jaradat (2012), *Towards a Safety-oriented Process Line for Enabling Reuse in Safety Critical Systems Development and Certification*, Post-proceedings of the 35th IEEE Software Engineering Workshop (SEW-35).

[4] B. Gallina, S. Kashiyarandi, H. Martin and R. Bramberger (2014), *Modelling a Safety- and Automotive-oriented Process Line to Enable Reuse and Flexible Process Derivation*, Proceedings of the 8th IEEE International Workshop on Quality-Oriented Reuse of Software (QUORS), joint workshop at COMPSAC conference, IEEE Computer Society, doi: 10.1109/COMPSACW.2014.84, pp. 504-509,Västerås (Sweden).

[5] BS EN50126-1 (1999), *Railway applications – The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*

[6] BS EN50128 (2011), *Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems*

[7] BS EN50129 (2003), *Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling*

[8] C. Holloway (2008), *Safety case notations: Alternatives for the non-graphically inclined?*, Proceedings of the 3rd IET International Conference on System Safety, IET Press, pp. 1-6.

[9] GSN (2011), *Community Standard Version 1*.

[10] L. Emmet and G. Cleland (2002), *Graphical notations, narratives and persuasion: A pliant systems approach to hypertext tool design*, in Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia, ser. HYPERTEXT '02. New York, NY, USA: ACM, pp. 55–64.

[11] Object Management Group (2008), *Software and Systems Process Engineering Meta-Model (SPEM)*, v2.0. Full Specification formal.

[12] R. Dardar, B. Gallina, A. Johnsen, K. Lundqvist, M. Nyberg (2012), *Industrial Experiences of Building a Safety Case in Compliance with ISO 26262*, Proceedings of the 2nd IEEE WoSoCER, joint event of the 23rd International Symposium on Software Reliability (ISSRE), Dallas (Texas), IEEE Computer Society, ISBN 978-1-4673-5048-8, USA.

[13] SACM, *http://www.omg.org/spec/sacm/1.0.*

[14] SYNOPSIS-SSF-RIT10-0070: *Safety Analysis for Predictable Software Intensive Systems,* Swedish Foundation for Strategic Research.

[15] *WEFACT: Workflow Engine for Analysis, Certification and Test*, http://www.ait.ac.at/research-services/research-services-digital-safety-security/verification-and-validation/methods-and-tools/wefact-workflow-engine-for-analysis-certification-and-test/?L=1