BENEFITS OF SECURITY-INFORMED SAFETY-ORIENTED PROCESS LINE ENGINEERING

Barbara Gallina, Mälardalen University, Västerås, Sweden
Laurent Fabre, Critical Systems Labs, Vancouver, Canada

Abstract

Nowadays, given the growing aircraft connectivity, security-informed safety is crucial. To certify aircrafts, safety as well as security standards need to be taken into consideration. In this context, a process engineer has to succeed in mastering the growing complexity of the standards interplay. To support process engineers, we propose to: first, consider a common terminological framework, aimed at reconciling security and safety within dependability; then identify and systematize commonalities and variabilities between the processes. To enable this systematization we introduce Security-informed Safety-oriented Process Line Engineering (SiSoPLE), which extends SoPLE to address security concerns. To show the effectiveness and benefits of SiSoPLE, we apply this new process line engineering to two aerospace standards, SAE ARP 4761 (Safety) and RTCA DO-326A (Security). We then provide our lessons learned and concluding remarks. Finally, we sketch some perspectives for future investigation.

Introduction

Nowadays, given the growing aircraft connectivity, security-informed safety is crucial. The recently published security standard, called RTCA DO-326A [1], applies to aircraft and aircraft systems. RTCA DO-326A focuses on security aspects that may affect aircraft airworthiness: more precisely, it only addresses security aspects that could impact flight safety. This standard specifies a top down risk assessment process with a generic set of activities and is intended to be compatible with other aerospace standards (e.g., SAE ARP 4754A [2], ARP 4761 [3]) dedicated to aircraft system certification. To ensure multi-concern assurance, more specifically to ensure security and safety assurance, while planning the (software) system development process, security-oriented and safety-oriented processes should be aligned. In this context, a process engineer has to succeed in mastering the growing complexity of the standards interplay [4]. More efficiently, overlapping process elements should be identified to facilitate reuse. Currently, however, reuse in the context of security-informed safety is hindered due to the presence of terminological differences between the security and safety community.

To overcome the terminological obstacle, in this work, we propose to consider a common terminological framework aimed at reconciling security and safety within dependability, which is the umbrella term that is typically used to embrace all the attributes that deal with trustworthiness. As it was observed in a previous work [5], dependability was indeed introduced as an umbrella term aimed at embracing a handful of attributes [6]. Decade after decade, dependability has grown and has constantly (explicitly or not) renewed itself [7] in terms of not only attributes (e.g. cyber-security), but also threats (e.g. vulnerabilities), and means (e.g. protections). Despite this renewal, essential principles can be identified. By revisiting those principles, reuse potential is revealed. Syntactical differences, which ontologically denote semantic equivalencies, should not prevent process engineers from recognizing ways of optimizing the standards interplay.

Once the common terminological framework is re-established, we propose to identify and systematize commonalities and variabilities between the processes. To systematize process-related commonalities and variabilities, in the context of safety standards, the notion of Safety-oriented Process Lines (SoPL) [8] and a corresponding SoPL Engineering method [9] have been introduced in the course of previous work. SoPLE consists of a two-phase method. The first phase is aimed at engineering the domain from a process perspective i.e., identifying and systematizing process-related commonalities and variabilities in order to concurrently engineer a set of processes. The second phase is aimed at deriving single processes via selection and composition of commonalities and variabilities. SoPLE has been promisingly applied to

engineer intra (automotive-centered) [10] as well as cross-domain (automotive and avionics) [11] SoPLs.

To address security-informed safety, we adopt SoPLE and we propose to extend it to address security. We call the new approach Security-informed Safety-oriented Process Line Engineering (SiSoPLE). Similarly to SoPLE, SiSoPLE is a method constituted of two phases: one aimed at engineering reusable security-informed safety process-related commonalities and variabilities and the other one aimed at engineering single security-informed safety processes via selection and composition of previously engineered reusable process elements.

To show the effectiveness and benefits of SiSoPLE, we apply it to engineer a SiSoPL based on RTCA DO-326A and SAE ARP 4761. The application will be limited to the initial phases of the two processes. More specifically, we will systematize commonalities and variabilities by comparing two activities: the Aircraft Functional Hazard Assessment within the Safety assessment process and the Preliminary Aircraft Security Risk Assessment within the Security Risk Assessment. We then provide our lessons learned and discuss related work. Finally, we provide our concluding remarks and sketch some perspectives for future investigation.

Background

This section recalls the essential background on which the presented work is based: safety and security, and security-informed safety; RTCA DO-326A; ARP4761; safety-oriented process lines engineering (SoPLE); and SoPL modeling.

Safety, Security, and Security-informed Safety

Aviezienis et al [6] introduced a terminological framework aimed at characterizing dependability in terms of its attributes, threats (faults, errors, and failures) and means. Dependability is usually indicated as an umbrella term, which embraces various aspects (attributes) related to trustworthiness. Safety and security are two dependability attributes.

Safety is defined as absence of catastrophic consequences on the user(s) and the environment. Security is defined as composite attribute constituted of availability, integrity, and confidentiality. Availability is defined as readiness for correct service. Integrity is defined as absence of improper system alterations. Finally, confidentiality is defined as absence of unauthorized disclosure of information.

Security-informed safety is an expression that has been recently introduced [12] to indicate an old truth: “For a system to be safe, it also has to be secure”. To guarantee an agreed upon level of safety/security, besides knowing what can go wrong, a risk assessment is needed.

Despite the existence of the dependability terminological frameworks and despite the awareness related to the above-stated truth, the security and safety communities have progressed by following different development paths. For instance, they define risk in a slightly different way. The safety community defines risk as the evaluation of the effect of a failure condition. This assessment takes into consideration the probability and severity and thus enables the judgment with respect to acceptability.

The security community defines risk [13] as *threat x vulnerability x consequence*, where consequence takes into consideration the attacker capability, the asset (i.e., aircraft if the risk is assessed at aircraft level) exposure and thus enables the judgment with respect to acceptability.

Further to terminological differences, process differences exist between the security and the safety domains. However there are strong reasons to align the safety and the security processes. For the purpose of this work four main reasons are identified: (1) security assessment should be mostly focused on safety-critical and safety-related functions. If security assessment is performed without the knowledge of failure conditions it may be performed inadequately and potentially not completely. Therefore safety assessment should feed inputs to the security risk assessment process to highlight functions of importance to the security analysis. (2) safety decisions regarding requirements and architecture should ideally not interfere with similar security decisions. In the worst case, safety measures could conflict with security measures or one domain could limit technical solutions for the other domain. Architecture or equipment decision rather than being taken unilaterally should be taken in a collaborative manner between safety and security, (3) Once security threats are identified, they may need to be fed back into the safety process to show the relationship between threat conditions and failure

conditions (4) finally a common picture of risk assessment encompassing security and safety will likely be preferred by certification authorities. Certification authorities may accept separate system assessments for safety and security. However the certification authorities will expect to see a global understanding of these risks and their influence on system design.

RTCA DO-326A/ED-202A

RTCA DO-326A/ED-202A [1] is a joint product of two industry committees: the EUROCAE Working Group WG-72, titled "Aeronautical Systems Security" and the RTCA Special Committee SC216, also titled "Aeronautical Systems Security". DO-326A is a document that provides guidance to handle the threat of intentional unauthorized electronic interaction to aircraft safety. More specifically, it defines a set of partially ordered activities that need to be performed in support of the airworthiness process to handle such threat. This set of partially ordered activities is known as Airworthiness Security Process. This process is constituted of a set of activities: Plan for Security Aspects of Certification (PSecAC), Security Scope Definition, Preliminary Aircraft Security Risk Assessment, Security Risk Assessment, Security Development related activities, Security effectiveness assurance, Communication of evidence (via PSecAC Summary). These activities are in turn composed of various tasks. In this section, we focus on a single activity, called Preliminary Aircraft Security Risk Assessment (PASRA), which belongs to the risk assessment set of activities. PASRA is aimed at identifying threat conditions and threat scenarios and assessing all security risks at aircraft level. PASRA takes as input the architecture under consideration, failure conditions and severity (which are established during the execution of the system development process described in ARP4761) and the information related to the security environment and perimeter, defined during the Security Scope Definition. Based on the input received, the following set of tasks is performed within the PASRA task: Threat Condition Identification and Evaluation, Threat Scenario Identification, Security Measure Characterization, and Level of Threat Evaluation. The final outcome of PASRA is the Preliminary Security Effectiveness Objectives, based on identified & evaluated threat conditions. DO-326A describes what security-related activities need to be performed but does not provide much guidance about how to perform these activities. DO-326A is expected to be used in conjunction with its companion document DO-356 [14], which provides guidance and methods for accomplishing the activities identified in DO-326A in the areas of Security Risk Assessment and Effectiveness Assurance.

ARP4761 Including its Expected Evolution

ARP4761 [3] Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment is an Aerospace Recommended Practice from SAE International. ARP4761 is a document that provides guidance to perform safety assessment. More specifically, defines a set of partially ordered activities that need to be performed in support of the airworthiness process to handle hazardous events (system and equipment failure or malfunction that may lead to hazard). This set of partially ordered tasks is known as Airworthiness Safety Assessment Process. This process, as newly stated in ARP4754A, is constituted of: Functional Hazard Assessment (FHA), performed at aircraft and system level, Preliminary Aircraft Safety Assessment (PASA), Preliminary System Safety Assessment (PSSA), System Safety Assessment (SSA) and, Aircraft Safety Assessment (ASA). In this paper, we focus on Aircraft-level FHA. Aircraft-level FHA is aimed at identifying failure conditions and assessing all safety risks at aircraft level. Aircraft-level FHA takes in input the list of top-level functions plus the initial design decisions (architecture), the aircraft objectives and requirements. Based on the input received, the following set of steps is performed within the Aircraft-level FHA task: Identification of all functions and corresponding failure conditions; determination of effects of the failure conditions and classification of the determined effects. The final outcome of Aircraft-level FHA is the safety objectives and the derived safety requirements, based on identified & evaluated failure conditions.

Safety-oriented Process Lines Engineering

Safety-oriented process lines (SoPLs) [8] represent sets of safety-oriented processes that may exhibit: full commonalities (equal process elements), partial commonalities (structured process elements that are partially equal), and variabilities. Variabilities denote

elements that may vary e.g., optional process elements or process elements that represent variants and can be chosen instead of others at specific variation points. The fundamental process elements to be interconnected to model processes are: tasks (which represent broken down units of work), work products (e.g., deliverables), roles, guidance, and tools. Additional information on SoPLs as well as SoPLs Engineering (SoPLE) can be found in [7-10].

Safety-oriented Process Line Modeling

As we discussed in [15], currently, no language is available to model safety-oriented process lines. Recently, two relevant extensions of SPEM 2.0 [16] have been proposed: vSPEM [17], to model process lines and S-TunExSPEM [18] to model and exchange safety-oriented processes.

For the modeling purposes of this paper, an extended combination of these two extensions could represent an interesting solution. More precisely, S-TunExSPEM could be extended with vSPEM constructs plus additional constructs to model security concerns. Thus, in this subsection, we recall essential information related to these extensions. More specifically, with respect to S-TunExSPEM, we briefly recall its safety-tunability, which is supported by the language constructs depicted in Table 1. As Table 1 shows, ordinary process elements (task, role, etc.) can be decorated with a safety hat, which indicates the criticality level and, as extensively explained in [18], can be in different colors.

Table 1. Subset of S-TunExSPEM Icons

Task	Role	Tool	Work product	Guidance
				

With respect to vSPEM, we recall its support for variability by focusing on the concrete syntax. As Table 2 shows, vSPEM basically introduces the possibility to model: 1) variation points, by decorating SPEM2.0 icons with empty circles; and 2) variants, by decorating SPEM2.0 icons with a V.

Table 2. Subset of vSPEM Icons

Concept	Variation point	Variant
Task		

To connect a variant (optional/alternative/etc. process element) to a variation point, vSPEM provides the *occupation relationship arrow*, which is an arrow having a filled circle on the opposite side.

Common SiS-related Terminology

As recalled previously, safety and security are two dependability attributes. Thus, the threats that hinder dependability, the threats' causation chain and the means that can face those threats can be equivalent for safety as well as security engineering.

Based on Avizienis et al [6], we can infer that an *incompetence fault* can represent a *vulnerability* (e.g., a programmer introduces non deliberately a weakness). An *external fault* can represent an *attack*, which, if performed in the presence of a vulnerability and if not mitigated, can lead to a failure (*threat condition*), which may have security as well as safety-related catastrophic consequences. As discussed in [19], if the causation chain is developed independently by different teams to satisfy the requirements of two distinct certification body, it is likely that the corresponding means will be duplicated.

Security-related threats can be mapped to a subset of Avizienis et al. faults. The term threat (Intentional Unauthorized Electronic Interaction) is defined in RTCA DO-326A/ED-202A. Similarly, Security-related threat-conditions (or failure conditions) can be mapped to conditions that are fulfilled as the consequence of the occurrence of a subset of Avizienis et al. failures.

As recognized in [12], a "lingua franca" is needed. A common SiS-terminology or a clear semantic mapping between safety and security is necessary to reveal the commonality. Obviously, in this section, we did not aim at being exhaustive. The objective is rather to continue paving the way towards succeeding in being persuasive with respect to the potential return on investment that such unified terminology could entail.

Towards Modeling SiSoPLs

In this subsection, we propose an extension of the combination of S-TunExSPEM and vSPEM. We call this extension SiS-TunvExSPEM, since it offers tunability capabilities for families of SiS-oriented processes. The SiS-support is concretized by

decorating safety hats with locks. Similarly to safety hats, security locks can be in different colors to represent different criticality levels.

Table 3 shows one single SiS-process element: a SiS-task. The other process elements would be modeled in the same way: by decorating with hats and locks in case of SiS-process elements or by decorating with only hats or only locks in case of safety-process elements or security-process elements respectively.

Table 3. Task-related Icon for Modeling SiS-tasks

SiSoPLE

In this section we present our proposal, called SiSoPLE, which stands for Security-informed Safety-oriented Process Line Engineering. Similarly, to SoPLE, after a scoping phase, the domain engineering phase is performed, during which commonalities and variabilities are identified. To do that, for each standard, we take the following actions:

• identification of certification-relevant process elements (e.g., activities and tasks);

• identification of the order in which activities and tasks should be performed;

• identification of the way in which tasks are grouped to form activities;

• identification of the way in which activities are grouped to form phases.

Then, we compare activities with activities, tasks with tasks, etc. We also compare the order of execution. To ease this comparison, we try to overcome several aspects such as: irrelevant terminological differences; irrelevant ordering differences; irrelevant grouping differences. More specifically, to overcome irrelevant terminological differences we build on top of the dependability-related terminological framework.

Overcoming irrelevant terminological differences or identifying significant points of variations is crucial since it permits (process) engineers to reduce the complexity of the systems to be engineered as well as the complexity of the certification process.

Once the commonalities and variabilities are known, we model them by using SiS-TunvExSPEM. To engineer single processes, aimed at satisfying a single certification body, process elements are selected and composed: all the commonalities are selected, jointly with the required variants, selected at corresponding variation points.

A security informed safety process line is expected to enable the alignment of security and safety standards. As discussed in the background, there are strong reasons to enable such alignment since, if the alignment is not performed, the resulting safety assessment conclusions may be incomplete, the technical solution might be less than ideal and more engineering effort might be required to harmonize both security requirements and architecture with the safety requirements late in the design phase.

While there is also potential for re-use between the security and the safety processes, these aspects mostly highlight that without some level of synergy between the security and the safety process, an organization may not produce a safe system or encounter resource and/or technical challenges.

Applying SiSoPLE

In this section we give initial findings related to the application of SiSoPLE. First of all we scope our SiSoPL. We select two standards, DO-326A/ED-202A and ARP4761, in the avionics domain and we limit our attention to two units-of-work, the Preliminary Aircraft Security Risk Assessment (PASRA) and the Aircraft-level Functional Hazard Assessment (AFHA), which are respectively defined in those standards. Within standards, units of work are often called differently: steps/phases/activities. In this section, we consider them as SPEM2.0-like tasks. Based on the description given in natural language in the background, we compare PASRA and AFHA and we identify and model commonalities and variabilities. Since both tasks are characterized by similar steps, a partial commonality-task (called PASISRA) can be identified and modeled. PASISRA is then characterized by a variation point that takes into consideration the variability. Since both tasks are expected to produce in output a work product indicating the identified and evaluated conditions,

such output can be seen as a partial commonality, characterized by a variation point that takes into consideration the variability. Similarly, partial commonalities can be identified with respect to the remaining process elements. The result of our comparative work is depicted in Figure 1.

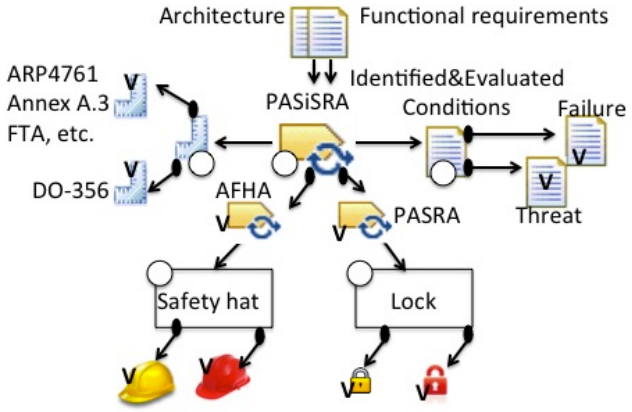

Figure 1 Security-informed Safety Task Line

More precisely, Figure 1 shows the PASiSRA task line modeled using TunvExSPEM. PASiSRA combines in a single model the AFHA and the PASRA. This combination is justified by the recognition that both tasks deal with risk assessment.

For sake of clarity, it should be stated that Figure 1 could be further enriched with additional process elements/information. In real settings, a task is performed by a role with well-defined competences. Moreover, since some tasks/steps are typically automated, tools can be specified.

At the time being, the aim of Figure 1 is not be complete and exhaustive (ready for certification purposes). The aim of Figure 1 is rather to show the potential of synthesizing/aligning the two tasks within a single model to enable concurrent engineering of a set of processes. By selecting and composing adequate process elements (we assume that a set of cross-cutting concerns have been expressed), it is possible to derive AFHA, as shown in Figure 2, and PASRA, as shown in Figure 3.

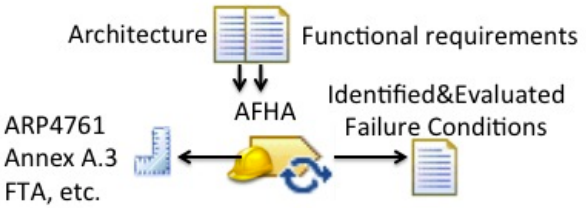

Figure 2 AFHA

Figure 3 PASRA

At this stage of development this model only supports the alignment of the two tasks. However, in the case of in-depth development related to the semantic mapping between safety and security, the number of variants could be reduced since terminological differences could be overcome. Ideally, a unified PASiSRA could probably be modeled as shown in Figure 4.

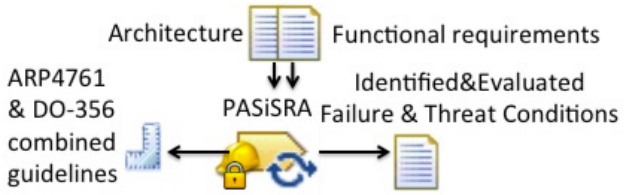

Figure 4 Potential Semantically Unified PASiSRA

Lessons Learnt

From the application of SiSoPLE, despite the simplicity of our illustration, we can draw the following lessons.

General soundness- SiSoPLE is sound since commonalities and manageable variabilities can be identified and modeled. SiSoPLE is also beneficial since commonalities enable reuse. More specifically, a single plan for aspects of certifications could be authored, instead of two (PSAC and PSecAC). SiSoPLE enables the alignment of multiple standards within a single model and thus it offers a means for the introduction of synergies between safety and

security experts, avoiding the potential issues, which were discussed in the background.

Scalability- SiSoPLE is scalable. Since the modeling language sketched represents an extension of SPEM2.0, a (family of) process(es) model can be structured by using components. Thus, scalability is guaranteed via the "divide and conquer" principle.

Effectiveness: It is yet unknown how much engineering effort saving will directly result in executing a PASiSRA rather than a separate AFHA and a PASRA. It is still likely that a security engineer and a safety engineer will still be required to execute the PASiSRA. However at the project level savings will result because conflicts between safety and security will be dealt with early in the lifecycle and because the risk of re-work later in the development cycle is reduced. This alignment will also produce a safer product since security has been considered in the context of safety. Overall SiSoPL provides a framework where potential conflicts between safety and security can be resolved early. This alignment will avoid re-work later on in the development life-cycle and ensures a safe system is being engineered.

Applicability- SiSoPLE is in its initial development stages: currently, it only offers a vision-solution but no mature and tool-supported approach. Thus it is not yet applicable in realistic industrial settings. However, given its soundness, it could/should be manually applied in research-oriented industrial settings to unveil its potential and attract a substantial critical mass, needed to recognize the emergent necessity of reconcile and cross-fertilize the safety and security communities in order to speed up the semantic mapping that is fundamental to save time and cost as well as reduce complexity.

Related Work

The relevance of harmonizing and cross-fertilizing the safety and security communities is well known. Within the MAFTIA project [20], researchers have worked on a common terminological framework. Littlewood et al.[21] have investigated if and how diversity-based fault tolerant strategies typically used for reliability and safety engineering can be used also for security engineering. They have concluded that some basic insights from probabilistic modelling in reliability and safety apply to examples of design for security.

The necessity of combining safety and security-related certification processes in order to save time and money as well as complexity via reduction of duplication has been recognised for more than a decade. Within the SafSec project [22] a common methodology for security accreditation and safety assurance was developed [19, 23]. This methodology is based on the recognition that both safety and security processes recommend risk-driven development processes. Safety hazards and security threats are both analysed via a unified risk management process. The approach, known as dependability by contract [24], is aimed at developing a unique dependability case incorporating both security and safety concerns via a compositional contract-based approach.

Our work on SiSoPLE may be considered as an additional ring of the chain aimed at enabling time and cost reduction via reuse within safety and security certification. Our focus, within this paper, is on reuse of process elements. Our work differs from the above-cited related work since it proposes to systematize reuse via the adoption and extension of product line engineering principles. We do not propose a contract-based approach, even if eventual (cross-cutting) constraints between process elements could be expressed as contracts.

Conclusion and Future Work

As discussed throughout the paper, nowadays, to certify safety-critical systems, security-related standards have to be considered in addition to safety-related standards. All these standards suggest assessment and development processes. To speed-up the process-related certification and master the growing complexity of these standards, the interplay among these various standards needs to be properly understood and optimized. In this paper, we have proposed to first of all recognize that security and safety are two attributes of dependability and that thus a common terminological framework can be adopted. Once the terminological framework is adopted, identification and systematization of common and variable process elements becomes easier. To perform such identification and systematization we have proposed SiSoPLE. Then, we have shown how to engineer a SiSoPL and discussed its benefits and current limits.

An important theme addressed in this paper is that the benefits of SiSoPLE will be obtained at the

project level if not directly at the task level. Individual safety / security tasks still need to be performed with some level of synergy between them. However the more important benefits are for the project: identification and resolution of safety / security issues early in the development life-cycle, and therefore minimizing the need for re-work late in the development cycle. Finally SiSoPLE also brings more confidence that a safer system has been produced and a faster certification process can be expected.

In the future, we aim at further developing SiSoPLE in various directions. First of all, jointly with safety and security certifiers, we aim at clearly scoping and fully engineer our SiSoPL. Then, by building on top of currently available metrics for product lines [25], we aim at defining metrics that allow process engineers to evaluate the reduction in terms of time and cost enabled by the systematization of reuse. We also aim at further investigating modelling capabilities targeting SiSoPLs. Finally, as done with SoPLE [15, 26-27], we aim at enabling model-based certification and automate the generation of process-based security and safety assurance cases semi-automatically. In a long-term future, the semi-automatic generation could embrace also product-based arguments by further developing Anti-Sisyphus [28].

References

[1] RTCA DO-326A, 2014, Airworthiness Security Process Specification, RTCA.

[2] ARP4754A, 2010, Guidelines for Development of Civil Aircraft and Systems, SAE International.

[3] ARP-4761, 1996, Guidelines and Methods for Conducting the Safety Assessment process on Civil Airborne Systems And Equipment.

[4] Ferrell, T., U., Ferrell, 2014, Assuring Avionics - Updating the Approach for the 21st Century, Proceedings of SafeComp-Workshops, the International Conference on. Computer Safety, Reliability and Security (SafeComp), Lecture Notes in Computer Science, Vol. 8696, pp. 375-383, Springer, Florence Italy.

[5] Gallina, B., September 9th, 2014, Critical Infrastructure Protection: the Eternal Return of Dependability-related Essential Principles. Proceedings of the 1st International Workshop on Reliability and Security Aspects for Critical Infrastructure Protection (ReSA4CI), Springer, LNCS 8696, ISBN 978-3-319-10505-5, Florence Italy.

[6] Avizienis, A., J.-C., Laprie, B., Randell, C., Landwehr, 2004, Basic concepts and taxonomy of dependable and secure computing. In: IEEE Trans. Dependable Sec. Comput. 1(1): 11-33.

[7] Laprie, J.-C., 2008, From Dependability to Resilience. LAAS Report no. 08001. LAAS-CNRS, Toulouse, France.

[8] Gallina, B., I. Sljivo, and 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), IEEE Computer Society, ISBN 978-1-4673-5574-2, Heraclion, Crete, Greece.

[9] Gallina B. et al, 2014, nSafeCer, D121.1: Generic process model for integrated development and certification.

[10] Gallina, B., and S. Kashiyarandi, and H. Martin and R. Bramberger, 2014, Modeling 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), IEEE Computer Society, Västerås, Sweden.

[11] Gallina, B., and S. Kashiyarandi, and K. Zugsbrati and A. Geven, September 8, 2014, Enabling Cross-domain Reuse of Tool Qualification Certification Artefacts. Proceedings of the 1st International Workshop on DEvelopment, Verification and VAlidation of cRiTical Systems (DEVVARTS), Springer, LNCS, Florence, Italy.

[12] Bloomfield, R., R. Stroud, 2013, Security-Informed Safety "If it's not secure, it's not safe". Marc-Olivier Killijian. Proceedings of the International Conference on. Computer Safety, Reliability and Security (SafeComp) FastAbstract, Toulouse, France. pp.NC. <hal-00926459>.

[13] Gil-Casals, S., P., Owezarski, G., Descargues, 2012, Risk Assessment for

Airworthiness Security. Proceedings of the International Conference on. Computer Safety, Reliability and Security (SafeComp), Magdeburg, Germany. pp.8,. <hal-00698523>

[14] RTCA DO-356, 2014, Airworthiness Security Methods and Considerations, RTCA.

[15] Gallina, B., K. Lundqvist and K. Forsberg, October 5-9, 2014, THRUST: A Method for Speeding Up the Creation of Process-related Deliverables. IEEE 33rd Digital Avionics Systems Conference (DASC-33), doi:10.1109/DASC.2014.6979489, Colorado Springs, CO, USA.

[16] OMG, 2008, Software & systems Process Engineering Meta-model (SPEM), v 2.0. Full Specification formal/08-04-01, Object Management Group.

[17] Martınez-Ruiz, T., F. Garcia, and M. Piattini, 2014, Towards A SPEM v2.0 Extension to Define Process Lines Variability Mechanisms. Book Chapter, DOI: 10.1007/978-3-540-70561-1_9.

[18] Gallina, B., and K. R. Pitchai and K. Lundqvist, 2014, S-TunExSPEM: Towards an Extension of SPEM 2.0 to Model and Exchange Tuneable Safety-oriented Processes. 11th International Conference on Software Engineering Research, Management and Applications (SERA), SCI 496, Springer, ISBN 978-3-319-00947-6, Prague, Czech Republic, August 7-9, 2013.

[19] Lautiere, S., D., Cooper, and D., Jackson, February 2005, SafSec: Commonalities Between Safety and Security Assurance, Proceedings of the 13th Critical Systems Symposium, Southampton, England.

[20] The MAFTIA project, http://research.cs.ncl.ac.uk/cabernet/www.laas.research.ec.org/maftia/

[21] Littlewood, B., L., Strigini, 2004, Redundancy and Diversity in Security. Proceedings of the European Symposium on Research in Computer Security (ESORICS), LNCS, Springer-Verlag, pp. 423-438.

[22] Praxis High Integrity Systems, SafSec: Integration of Safety & Security Certification, November 2006.

[23] Lautiere, S., D., Cooper, D., Jackson, T., Cockram, 2004, Assurance Cases: how assured are you?, supplemental volume to DSN-2004, Proceedings of the International Conference on Dependable Systems and Networks..

[24] Dobbing, B., S., Lautiere, 2007, Dependability-by-Contract. In The Safety of Systems (pp. 35-51). Springer London.

[25] Torkamani, M. A., April, 2014, Metric suite to Evaluate Reusability of Software Product Line, International Journal of Electrical and Computer Engineering (IJECE), Vol. 4, N.2, pp. 285-294.

[26] Gallina, B, November 3-6, 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, doi: 10.1109/ISSREW.2014.30, pp. 204-209.

[27] Gallina, B., L. Provenzano, June, 2015, Deriving Reusable Process-based Arguments from Process Models in the Context of Railway Safety Standards. 20th International Conference on Reliable Software Technologies-Industrial Presentation- (Ada-Europe), Madrid, Spain.

[28] Gallina, B., May 19th, 2015, Towards Enabling Reuse in the Context of Safety-critical Product Lines. 5th International Workshop on Product LinE Approaches in Software Engineering (PLEASE), Florence, Italy.

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

Acknowledgements

This work has been partially supported by the Swedish SSF SYNOPSIS project [29].