Enabling Cross-domain Reuse of Tool Qualification Certification Artefacts

Barbara Gallina[1], Shaghayegh Kashiyarandi[1], Karlheinz Zugsbratl[2], and Arjan Geven[2]

[1] MRTC, IDT,
Mälardalen University, P.O. Box 883, SE-72123 Västerås, Sweden
name.surname@mdh.se
[2] TTTech, Wien, Austria
name.surname@tttech.com

Abstract. The development and verification of safety-critical systems increasingly relies on the use of tools which automate/replace/supplement complex verification and/or development tasks. The safety of such systems risks to be compromised, if the tools fail. To mitigate this risk, safety standards (e.g. DO-178C/DO330, IEC 61508) define prescriptive tool qualification processes. Compliance with these processes can be required for (re-)certification purposes. To enable reuse and thus reduce time and cost related to certification, cross-domain tool manufacturers need to understand what varies and what remains in common when transiting from one domain to another. To ease reuse, in this paper we focus on verification tools and model a cross-domain tool qualification process line. Finally, we discuss how reusable cross-domain process-based arguments can be obtained.

Keywords: Tool qualification processes, safety cases, process-based arguments, safety standards, DO-178C, ISO 26262, IEC 61508, Software Process Engineering Meta-model (SPEM) 2.0, Goal Structuring Notation (GSN)

1 Introduction

In the context of safety-critical systems engineering, software is increasingly developed and verified (semi)-automatically. Tools for code generation as well as for verification are introduced to (semi)automate/replace/supplement complex tasks. Since safety might be compromised if such tools fail, safety standards (e.g. IEC 61508 [1]) prescribe tool qualification processes (which represent process reference models for tool qualification). More recently DO-178C [2], which is going to become the de-facto standard for certifying avionic software, and more precisely its supplement DO330 has entered the scene with new requirements on the tool qualification process. This supplement provides a very detailed process which has been conceived to be used for cross-domain certification, assumed that the domain-specific documents confirm its applicability. As a consequence, since compliance with the DO330 process reference model may constitute a mandatory requirement for certification purposes, companies (including TTTech) used to develop tools in compliance with either DO-178B [3] or IEC 61508 have to quickly perform a gap analysis in order to introduce adequate changes in their processes for being prepared for efficient re-certification.

In the automotive domain and within the context of intra-domain certification, we face similar circumstances such as the introduction of new standards and thus new requirements related to processes. For this case, we proposed to exploit the time for the gap analysis to reach a solution that goes beyond adhoc and temporary patches. More specifically, to enable flexible but compliant development processes, we proposed (and presented in [4]) to adopt a safety-oriented process line approach and model the set of prescriptive processes as a process line. The time for the gap analysis was thus used to identify and model the commonalities and variabilities among processes in order to enable reuse of process elements. The experience gathered in the automotive domain is exploited and further developed in this work. More specifically, in this paper we do not only enable reuse of process elements by modeling a cross-domain tool qualification process line, but we also enable reuse of certification artifacts by relating the process line with the corresponding family of process-based arguments related to process compliance. To do that, we show how reusable process-based arguments can be obtained from a process line. The need of harmonizing qualification guidance amongst standards is clearly stated in the perspectives discussed in [5]. The demand for reusing certification data related to the tool qualification process is explained in [6], while the motivation of providing a knowledge base concerning qualification effort is described in [7]. Our proposal for enabling reuse of process-related artifacts contributes to the satisfaction of these abovementioned needs and addresses the current problems as stated in related work and faced in practice. More specifically, the tool qualification process line contributes in engineering the harmonization of the standards. It systematizes the comparative study we performed on the set of tool qualification processes. With this, the relation between the process line and the set of corresponding process-based arguments enables reuse of certification artifacts and at the same time constitutes a knowledge base of certification strategies.

The rest of the paper is organized as follows. In Section 2, we provide essential background information. In Section 3 we present our cross-domain safety-oriented process line constituted of tool qualification processes. In Section 4, we give an intuition concerning the derivation of reusable process-based arguments from the process line. In Section 5 we discuss related work. Finally, in Section 6 we present some concluding remarks and future work.

2 Background

In this section, we present the background information on which we base our work. In particular, in Section 2.1 we provide essential information concerning prescriptive tool qualification processes. In Section 2.2, we briefly present

SPEM 2.0, the process modeling language used to model the tool qualification process line. In Section 2.3, we briefly present Goal Structuring Notation (GSN), the graphical notation used to argue about process compliance.

2.1 Tool qualification processes

To ensure that tools behave correctly concerning the imposed safety requirements, safety standards define tool qualification processes. These processes are typically constituted of three phases: classification, qualification, and usage [8]. During the classification phase, the tools are classified according to the level of confidence that is required to ensure their behavior is in-line with the safety requirements. Levels are named differently from one standard to another: *tool confidence levels* in the ISO 26262 [9], *tool criteria* in the DO-178C and *tool classes* in the IEC 61508. If a tool is considered to be harmless, it can be used without requiring any qualification. During the qualification phase, the tools that were considered potentially harmful, have to be qualified, i.e. manufacturers have to show absence of hazardous events (failures that might lead to accidents). Finally during the usage phase, tools can be used within the specified restrictions.

Tool qualification processes embrace two categories of tools: development tools and verification tools. In the context of this paper the focus is put on verification tools. More specifically, the work has been performed having in mind the tool qualification process related to the *TTE-Verify* tool, a verification tool of *TTEthernet* networks. As a result, those parts of the standards which deal with the active contribution to the development, e.g. code generation, are not covered in this work.

2.2 Safety-oriented process lines and SPEM 2.0

Safety-oriented process lines [10] represent sets of safety-oriented processes that exhibit: full commonalities (equal process elements), partial commonalities (structured process elements that are partially equal), and some variabilities (e.g. optional process elements). Safety-oriented process lines can be modeled by adopting a two-phase approach consisting of a first phase aimed at modeling the domain and a second phase aimed at modeling the single processes.

SPEM (Software Process Engineering Meta-model) 2.0 [11] is the OMG's standard for systems and software process modeling. The selection of SPEM 2.0 for modeling process lines was extensively motivated in [10]. SPEM 2.0 offers support for the definition of reusable process content. Process engineers are enabled to define reusable work definition elements (e.g. tasks) as well as other process elements. An additional package called Method Plugin supports the creation of repositories for reuse of process content. SPEM 2.0 also offers support for variability modeling enabling the specification of (safety-oriented) process lines, as explored in [10] and in [4]. In Table 1, we recall some of the SPEM 2.0 graphical modeling elements that can be interrelated to model the process dynamics. In the table, we focus on the elements that we subsequently use in Section 3.

Table 1. Icons denoting Method Content Use elements

Task	TaskUse	WorkProduct

As discussed in [12], these elements could be extended to better model safety aspects. However, currently this extension does not embrace cross-domain needs. In the context of this work, we thus take the standardized SPEM 2.0 and provide SPEM 2.0 models by using Eclipse Process Framework Composer [13], which is a SPEM 2.0-compatible open source tool for authoring development method content and publishing processes.

2.3 Process compliance and GSN

Safety cases are contextualized structured arguments containing process and product-based sub-arguments. These sub-arguments are aimed at linking evidence with claims regarding system safety. In this paper, we focus on process-based arguments and more specifically on these process-based arguments that are used to show that the verification tools used to verify the software have been developed in compliance with the tool qualification process mandated by the standard. To document process-based arguments, we use the graphical notation called GSN [14]. The selection of GSN for documenting safety cases was extensively motivated in [15, 16]. GSN 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, we recall the concrete syntax of the GSN core modeling elements used in Section 4 in Figure 1. The following list provides their informal semantics:

- Goal: represents a claim about the system.
- Strategy: represents a method that is used to decompose a goal into sub goals.
- Context: represents the domain or scope in which a goal, evidence or strategy is given.
- Supported by: represents an inferential or evidential relationship. Inferential relationships declare that there is an inference between goals in the argument. Evidential relationships declare the link between a goal and the evidence used to substantiate it.
- In context of: represents a contextual relationship.

As Figure 1 shows, all the nodes are characterized by an identifier (ID) and a statement which is supposed to be written in natural language. Beyond the modeling elements presented in Figure 1, we also make use of the diamond-shaped element to characterize *to-be-developed* argumentation branches. Curly brackets within statements are used to denote variables.

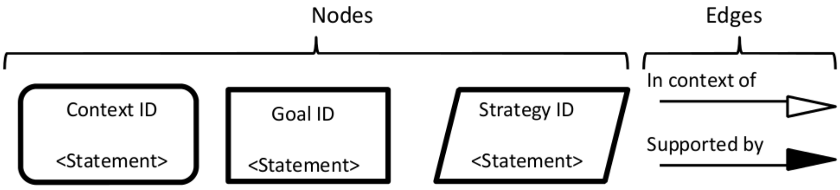


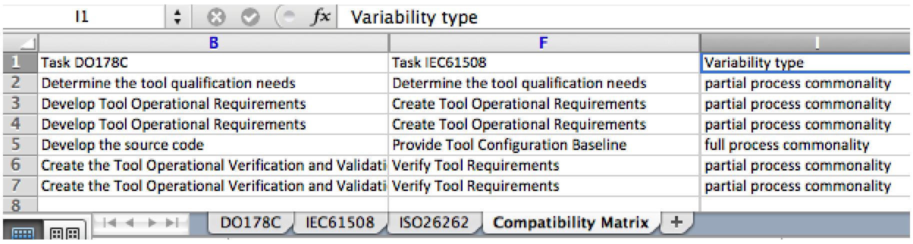
Fig. 1. Partial concrete syntax of GSN.

3 A cross-domain tool qualification process line

As discussed in [10], whenever prescriptive processes mandated by the standards exhibit evident similarities they can be treated as a safety-oriented process line. This fosters reuse of process elements thanks to the systematic engineering of commonalities and variabilities between processes. To identify commonalities and variabilities between tool qualification processes, the guidelines provided in [10] are followed. Thus, for each standard and for each phase, the following actions are taken:

- identification of activities, tasks, steps;
- 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.

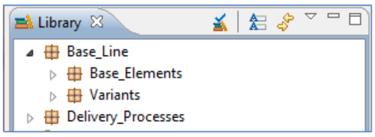
This identification requires a very detailed analysis of each of the explicit and implicit process-related pieces of information provided in the standard. Similarly to what has been done in [4], the gathered information has been documented in a spreadsheet (depicted in Figure 2) and then used to model the cross-domain process line in EPF Composer/SPEM2.0 according to the methodological framework proposed in [4].


I1 Variability type

	B	F	I
1	Task DO178C	Task IEC61508	Variability type
2	Determine the tool qualification needs	Determine the tool qualification needs	partial process commonality
3	Develop Tool Operational Requirements	Create Tool Operational Requirements	partial process commonality
4	Develop Tool Operational Requirements	Create Tool Operational Requirements	partial process commonality
5	Develop the source code	Provide Tool Configuration Baseline	full process commonality
6	Create the Tool Operational Verification and Validati	Verify Tool Requirements	partial process commonality
7	Create the Tool Operational Verification and Validati	Verify Tool Requirements	partial process commonality
8			

Fig. 2. Cut of the spreadsheet documenting the comparative analysis.

The compatibility matrix only compares DO-178C/DO330 and IEC 61508 because no classification is required anymore. The reason is that ISO 26262-8:11.4.6 states that a tool developed according to the DO330 standard can be considered sufficient for being suitable for ISO 26262 ASIL-D projects. The interested reader may refer to [17] for further details on the standards comparison. Figure 4 represents the SPEM 2.0/EPF-based safety oriented tool qualification process line. We create this process line in SPEM/EPF by following the methodological approach introduced in [4]. We thus make use of the package Method Plugin and we define a series of plug-ins aimed at containing base elements. As Figure 3 shows, we then organize them by using two logical packages (Base and Processes).

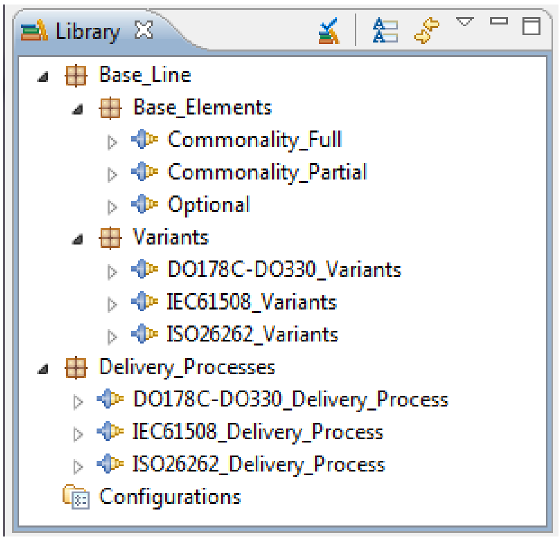


Library

Base_Line
Base_Elements
Variants
Delivery_Processes

Fig. 3. Top-level view of the SPEM2.0/EPF-based tool qualification process line.

We use Base (respectively Processes) for organizing plugins related to the Domain (Process) engineering phase. More specifically, we define one plug-in for each type of commonality (either full or partial) and variability (i.e., optional). We also define a plug-in for all the variants that are related to either partial commonalities or variabilities. In this paper, the naming convention used for tasks classified as partial commonality is that the name of DO330 is used.

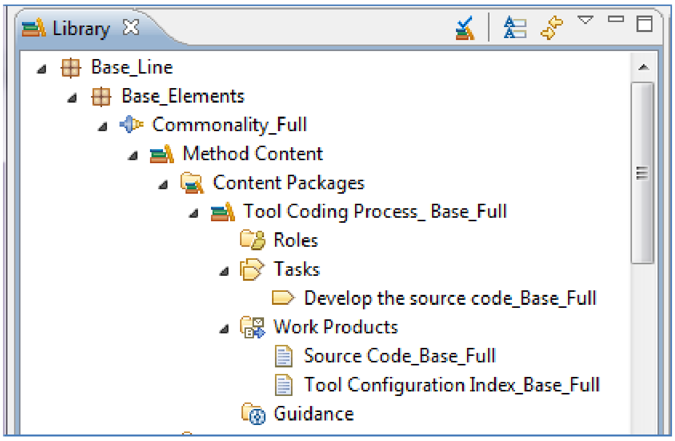


Library

Base_Line
Base_Elements
Commonality_Full
Commonality_Partial
Optional
Variants
DO178C-DO330_Variants
IEC61508_Variants
ISO26262_Variants
Delivery_Processes
DO178C-DO330_Delivery_Process
IEC61508_Delivery_Process
ISO26262_Delivery_Process
Configurations

Fig. 4. Lower-level view of the SPEM2.0/EPF-based tool qualification process line.

Figure 5 details the process elements contained in the plugin related to the full commonalities. It is in compliance with the information initially collected in the spread sheet.



Library

- Base_Line
  - Base_Elements
    - Commonality_Full
      - Method Content
        - Content Packages
          - Tool Coding Process_ Base_Full
            - Roles
            - Tasks
              - Develop the source code_Base_Full
            - Work Products
              - Source Code_Base_Full
              - Tool Configuration Index_Base_Full
            - Guidance

Fig. 5. SPEM2.0/EPF-based tool qualification process line.

From Figure 5, it clearly emerges that the task named *Develop the source code* is the only full commonality.

Once a cross-domain safety-oriented process line constituted of tool qualification processes is available, (partial) commonalities as well as variabilities are clearly systematized and single processes can be easily derived. Figure 6 and Figure 7 represent the single-processes derived from the safety oriented process line by selecting and composing desired process elements.

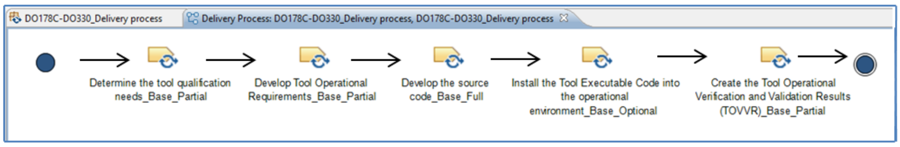

DO178C-DO330_Delivery process
Delivery Process: DO178C-DO330_Delivery process, DO178C-DO330_Delivery process
Determine the tool qualification needs_Base_Partial
Develop Tool Operational Requirements_Base_Partial
Develop the source code_Base_Full
Install the Tool Executable Code into the operational environment_Base_Optional
Create the Tool Operational Verification and Validation Results (TOVVR)_Base_Partial

Fig. 6. Derived DO330-compliant tool qualification process.

More specifically, to create single processes and thus populate the logical package *Delivery Processes*, full and partial commonalities must be selected. Finally to characterize single processes eventual additive as well as optional elements must also be selected. Besides selection, ordering of the process elements is necessary. This is done by setting the predecessor (as shown in Figure 8).

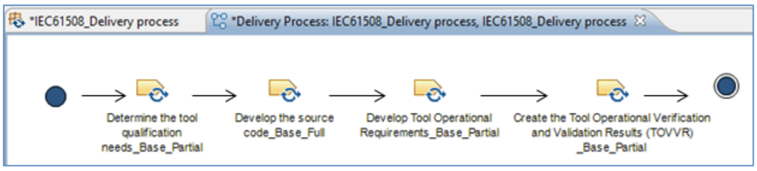
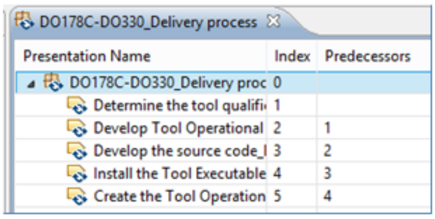

*IEC61508_Delivery process
*Delivery Process: IEC61508_Delivery process, IEC61508_Delivery process
Determine the tool qualification needs_Base_Partial
Develop the source code_Base_Full
Develop Tool Operational Requirements_Base_Partial
Create the Tool Operational Verification and Validation Results (TOVVR) _Base_Partial

Fig. 7. Derived IEC 61508-compliant tool qualification process.


DO178C-DO330_Delivery process

Presentation Name	Index	Predecessors
DO178C-DO330_Delivery proc	0	
Determine the tool qualifi	1	
Develop Tool Operational	2	1
Develop the source code_I	3	2
Install the Tool Executable	4	3
Create the Tool Operation	5	4

Fig. 8. Task ordering.

As it can be seen by comparing the two Figures 6 and 7, the two derived processes exhibit few variabilities in terms of tasks and thus the effort performed to be compliant with IEC 61508 can be rather easily reused to obtain the certification stamp by certification authorities responsible for checking compliance with DO330.

4 Enabling reuse of certification artifacts

To (re)certify tools, process compliance is required. Manufacturers have to show that the qualification process mandated by the standard has been performed. When moving from one domain to another, it is crucial to reuse certification data in order to reduce time and cost. To do that, the first necessary step is the recognition that certification data related to a process line exhibits commonalities and variabilities. Thus, a compositional approach based on product line-oriented practices enabling the selection and composition of commonalities and variabilities is the key solution for showing process compliance.

Typically, a company has to provide structured arguments which can be expressed graphically or in natural language to show compliance. In this section, based on the process line presented in Section 3 and on GSN recalled in Section 2.3, we give an intuition about how such compositional and reusable process-based arguments could look like. Our goal is thereby to illustrate how reuse can be enabled and accelerated via the tool qualification process line, thus we do not show a complete process-based argument.

Figure 9, in particular, shows how the sub-goal structure (fragment of the process-compliance argumentation) can reflect the tool qualification process line.

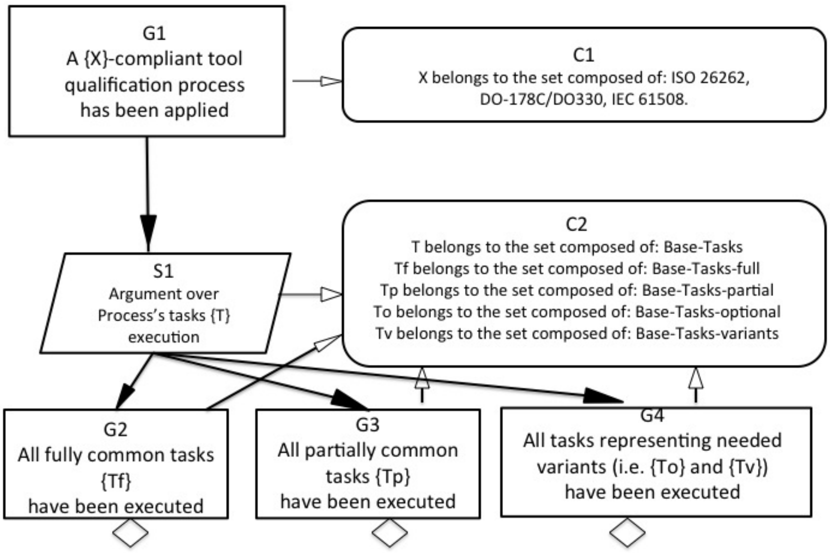

G1
A {X}-compliant tool qualification process has been applied
C1
X belongs to the set composed of: ISO 26262, DO-178C/DO330, IEC 61508.
S1
Argument over Process's tasks {T} execution
C2
T belongs to the set composed of: Base-Tasks
Tf belongs to the set composed of: Base-Tasks-full
Tp belongs to the set composed of: Base-Tasks-partial
To belongs to the set composed of: Base-Tasks-optional
Tv belongs to the set composed of: Base-Tasks-variants
G2
All fully common tasks {Tf} have been executed
G3
All partially common tasks {Tp} have been executed
G4
All tasks representing needed variants (i.e. {To} and {Tv}) have been executed

Fig. 9. Goal structure fragment representing a process-based argument.

From Figure 9 we retrieve the following argument fragment: the process is compliant with the process mandated by the standard under consideration. To support this top-level claim (G1 in Figure 9), a strategy (S1) is used to decompose it into sub-claims (G2-G4) which step by step can be more easily supported by evidence. The strategy focuses on a specific process element (i.e. task) and argues that compliance is achieved because all the common tasks and all the standard-specific tasks have been performed. From this argument fragment that only considers the initial break-down structure of the entire argumentation, it clearly emerges that:

- G2, once fully developed, can be easily fully reused.
- G3, once fully developed, can be easily partially reused.
- G4, once fully developed, cannot be reused.

Thus, the main effort during re-certification is expected to be limited to the development of G4.

5 Related Work

The necessity of ensuring compliance with the standards as well as the demand for reducing time and cost related to the certification process is currently providing the motivation for several research projects (e.g. [18, 19] and [20]).

To ensure compliance as well as reduce time and cost, different solutions (compliance checking, reuse, etc.) are being investigated under different perspectives, most of them product-based ones. Exceptions to this product-based focus are the contributions presented in [21–23].

In [21], the authors propose a workflow-based approach to provide: 1) reference models for the safety processes mandated by the standards and 2) automatic compliance checking capabilities of user-defined processes against reference models. However, the authors focus on single standards and do not investigate reuse possibilities.

In [22], the authors propose future research directions to address reuse issues in the context of cross-domain certification as well as in the context of evolutionary products. Their intention is to provide a common certification framework.

In [23], the authors propose a meta-model to capture entities (e.g. certification objectives with respect to the safety level) involved in software product line certification. Their proposal aims at representing the first step towards certifiable software product lines. It, indeed, has a potential to solve reuse issues at the process level but does not discuss reuse issues at the argumentation level.

6 Conclusion and Future Work

In this paper, we have presented a novel approach to reduce cost and time during the tool certification process. We have shown that by modeling the family of tool qualification processes via a safety-oriented process line, it is possible to identify reusable process elements and thus speed up the re-certification process when tools are expected to be used in different domains. We have also shown that these reusable process elements are reflected in the process-based arguments and thus not only qualification data (evidence) has the potential to be reused but also process-based sub-arguments. The main attention in this paper was given to the verification tools, however the approach can be extended to other tool categories as well as other kinds of safety-related processes. Due to space reasons, we also focused on process-related tasks and work products. As extensively discussed in [24], reuse also embraces all the other crucial process elements (namely, roles, work products, and guidance).

In a medium-term future, we aim at further developing our approach. First of all we will start to define a pattern for process compliance targeting cross-domain tool qualification processes. Then, we will work on providing an adequate tool-support allowing for semi-automatic generation of process-based and pattern-based arguments from process models. A master thesis on this research direction is already ongoing [25]. Finally, we also plan to introduce metrics to measure the real gain that our approach introduces.

Acknowledgments. This work has been partially supported by the European Project ARTEMIS SafeCer [18] and by the Swedish SSF SYNOPSIS project [19].

References

1. IEC61508: Functional safety of electrical/electronic/programmable electronic safety-related systems (2010)
2. RTCA Inc: Software Considerations in Airborne Systems and Equipment Certification, RTCA DO-178C (EUROCAE ED-12C), Washington DC. (2013)
3. RTCA Inc: Software Considerations in Airborne Systems and Equipment Certification, RTCA DO-178B (EUROCAE ED-12B), Washington DC. (1992)
4. Gallina, B., Kashiyarandi, S., Martin, H., Bramberger, R.: Modeling a safety- and automotive-oriented process line to enable reuse and flexible process derivation. In: 8th IEEE International Workshop Quality-Oriented Reuse of Software. (July 2014)
5. Camus, J.L., Dewalt, M.P., Pothon, F., Ladier, G., Boulanger, J.L., Blanquart, J.P., Quere, P., Ricque, B., Gassino, J.: Tool qualification in multiple domains: Status and perspectives. In: Embedded Real Time Software and Systems, Toulouse, France, 5-7 February. Volume 7991., Springer (2014)
6. Kornecki, A.J., Zalewski, J.: Design tool assessment (December 15 2003)
7. Kornecki, D.A.J., Zalewski, D.J.: The qualification of software development tools from the DO-178B certification perspective. CrossTalk - The Journal of Defense Software Engineering (April 2006)
8. Slotosch, O.: Model-based tool qualification - the roadmap of eclipse towards tool qualification. In Cerone, A., Persico, D., Fernandes, S., Garcia-Perez, A., Katsaros, P., Shaikh, S.A., Stamelos, I., eds.: SEFM Satellite Events. Volume 7991 of Lecture Notes in Computer Science., Springer (2012) 215–228
9. ISO26262: Road vehicles Functional safety. International Standard, November (2011)
10. Gallina, B., Sljivo, I., Jaradat, O.: Towards a Safety-oriented Process Line for Enabling Reuse in Safety Critical Systems Development and Certification. In: Post-proceedings of the 35th IEEE Software Engineering Workshop. SEW-35, Greece (2012)
11. Object Management Group: Software & Systems Process Engineering Meta-Model (SPEM), v2.0. Full Specification formal/08-04-01. (2008)
12. Gallina, B., Pitchai, K.R., Lundqvist, K.: S-TunExSPEM: Towards an Extension of SPEM 2.0 to Model and Exchange Tuneable Safety-oriented Processes. In Lee, R., ed.: 11th International Conference on Software Engineering Research, Management and Applications (SERA), August 7, 2013, Prague, Czech Republic. Volume 496/2014., Springer SCI (2014)
13. Eclipse Process Framework: http://www.eclipse.org/epf/
14. GSN: Community Standard Version 1. November. http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf (2011)
15. Dardar, R., Gallina, B., Johnsen, A., Lundqvist, K., Nyberg, M.: Industrial experiences of building a safety case in compliance with iso 26262. In: IEEE 23rd International Symposium on Software Reliability Engineering Workshops (ISSREW). (2012) 349–354
16. Gallina, B., Gallucci, A., Lundqvist, K., Nyberg, M.: VROOM & cC: a Method to Build Safety Cases for ISO 26262-compliant Product Lines. In: SAFECOMP Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR), HAL / CNRS report (September 2013)
17. Gallina B. et al.: nSafeCer, D121.1: Generic process model for integrated development and certification (2014)

18. ARTEMIS-JU-269265: SafeCer-Safety Certification of Software-Intensive Systems with Reusable Components. http://www.safecer.eu/ (2013)
19. SYNOPSIS-SSF-RIT10-0070: Safety Analysis for Predictable Software Intensive Systems. Swedish Foundation for Strategic Research
20. FP7 OPENCOSS: Open platform for evolutionary certification of safety-critical systems
21. Chung, P.W.H., Cheung, L.Y.C., Machin, C.H.C.: Compliance flow - managing the compliance of dynamic and complex processes. Know.-Based Syst. **21**(4) (May 2008) 332–354
22. Espinoza, H., Ruiz, A., Sabetzadeh, M., Panaroni, P.: Challenges for an open and evolutionary approach to safety assurance and certification of safety-critical systems. In: First International Workshop on Software Certification (WoSoCER). (2011) 1–6
23. Braga, R.T.V., Trindade Junior, O., Castelo Branco, K.R., Neris, L.D.O., Lee, J.: Adapting a Software Product Line Engineering Process for Certifying Safety Critical Embedded Systems. In: Proceedings of the 31st international conference on Computer Safety, Reliability, and Security. SAFECOMP, Springer-Verlag (2012) 352–363
24. Kashiyarandi, S.: Reusing Process Elements in the Context of Safety Critical Systems Development and Certification. Master's thesis, Mälardalen University, School of Innovation, Design and Engineering, Sweden (to appear)
25. Asghar Ali, E.: Deriving reusable process-based arguments from process models in the context of safety critical systems development and certification. Master's thesis, Mälardalen University, School of Innovation, Design and Engineering, Sweden (ongoing)