A Tool Suite for Assurance Cases and Evidences: Avionics Experiences

Alejandra Ruiz(✉), Xabier Larrucea, and Huascar Espinoza

ICT-European Software Institute Division, Tecnalia, Derio, Spain
{alejandra.ruiz,xabier.larrucea,huascar.espinoza}@tecnalia.com

Abstract. This paper describes a specification and an implementation of a flexible tool platform for assurance and certification of safety-critical systems. This tool platform is built upon a comprehensive conceptual assurance and certification framework. This conceptual framework is composed of a common information model called CCL (Common Certification Language) and a compositional assurance approach. Our tool platform allows an easy integration with existing solutions supporting interoperability with existing development and assurance tools. The ultimate goal of our platform is to provide an integrated approach for managing assurance cases and evidences resulting from a safety project.

Keywords: Tooling platform · Compliance · Standards · Argumentation · Evidence management

1 Introduction

Assurance [1] and safety certification[2] are among the most expensive and time-consuming tasks in the development of safety-critical embedded systems. Innovation and productivity in this market is curtailed by the lack of affordable certification and especially recertification approaches [3]. A common situation in safety-critical industrial domains is the fact that developers or manufacturers of a safety-critical system are required to demonstrate with evidences that their products are acceptably safe in a given context before it is formally approved for release into service. Conceptually, this means that all potential system hazards [4] – operational misbehaviour or conditions which might lead to an accident leading to injury or loss of human life or to damage to the environment – are either prevented or mitigated. The manufacturer is obliged to demonstrate the absence of risks and to increase the assessor's confidence with respect to the system's safety. In fact they must explicitly provide evidence of the system's conformance to relevant standards or reference models. This includes prescription that rigorous analysis, checking, and testing are carried out.

The identification of evidences [5] for the effectiveness of existing certification schemes is hard to come by. Typically a safety-critical application and its accompanying set of evidences are monolithic, based on the whole product, and a major problem arises when evolutions to the product came into play. Those evolutions become costly and time consuming because they entail regenerating the entire evidence-set.

R.V. O'Connor et al. (Eds.): EuroSPI 2015, CCIS 543, pp. 63–71, 2015.
DOI: 10.1007/978-3-319-24647-5_6

This paper is structured as follows; section 2 highlights current gaps in industrial environments. Section 3 presents the main concepts of our conceptual approach while on section 4 we discuss the principal functionalities over a particular example. Section 5 presents the benefits from using our tool suite and the main problems we try to give support to. Finally, section 6 indicates some main conclusions extracted from this work and the future work we are planning to deal with tool evolution.

2 Related Work

Practitioners face different situations during development and certification processes. One of them is to clearly define and maintain a chain of evidence adequate for safety certification. The identification and management of these evidences increase development time and costs. Different tools have been developed in order to support argumentation and evidence management efforts [12]. The arguments are usually packed into a safety case which can be 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*" [6].. There are some graphical notations which include the main concepts for argumentation such as the GSN (Goal Structuring Notation) [10] or CAE (Claim Arguments and Evidence) [112]. Both graphical notations facilitate the understanding of argumentations performed by the reviewer or assessor during an assurance case assessment. One step forward is the Structured Assurance Case Metamodel (SACM) [7] which is a standard developed by the Object Management Group (OMG) to model the different concepts that come up while exposing an argument. It has a richer set of concepts than the ones made explicit in the GSN. In fact the notion of how a particular claim is used in an argument - e.g. as supporting or indirect, and umbrella types of element in the argument. In addition, it has some extra concepts such as counter-evidence or assumed claims. However, it also lacks of GSNs features such as modularity and some forms of patterning to provide argumentation templates. SACM does not prescribe a specific graphical notation but proposes the use of the existing ones are possible graphical notation for its concepts. Therefore GSN notation could be used for describing SACM models. Regarding argumentation tools, there has been some work from NASA working on Advocate tool which uses GSN notation for the argumentation [13]. D-CASE [14] is a tool created by DEOS project where argumentation pattern functionality is linked with the use of parameters. D-CASE and Advocate approaches do not present clear relations to standards requirements or evidence management. NOR-STA platform [16] also provides argumentation support based on TRUST-IT method but they do not provide an interface to describe standards based requirements, and to generate automatically a report on compliance with respect to these standards.

Some initiatives integrate information from different sources and tools. This is the case of ModelBus [15] which offers support for creating an integrated tool environment. Nevertheless, ModelBus is focused on data integration from different tools. This paper integrates in a consistent and meaningful way project safety information and standard compliance information.

3 Conceptual Platform

This paper is framed under a European project called OPENCOSS (Open Platform for EvolutioNary Certification Of Safety-critical Systems) which is a large-scale collaborative project of the EU's Seventh Framework Program. OPENCOSS focuses on the harmonization of safety assurance and certification management activities for the development cyber-physical systems in automotive, railway and aerospace industries. This paper presents a conceptual approach dealing with the aforementioned identified situations. This work is based on assurance cases and evidences approaches and we have identified the following challenges in safety critical systems [8]:

- **Unawareness of the certification process.** The lack of awareness on the certification aspects is a frequent problem in the current practice, in large part arising due to poor visibility into the architecture of systems, their design rationale, how components were verified and integrated, and finally how the system components and the system as a whole were certified.
- **Data exists in many places, with different formats, multiple copies and versions.** Usually, engineers submit paper-based reports and do not know where the reports go and are unable to follow up. Quality and safety managers assess and classify information. Excel and Word documents are often exchanged, of which multiple copies and versions exist.
- **Time-consuming to compile reports, artefacts and difficult to retrieve.** Often paper-based reports are filled, which are time-consuming to aggregate. It is painful to generate trend analysis reports because the organizations do not have easy access to data, reports and policies.
- **Difficulties in interpretations of argumentation.** Determining the degree of compliance with specified standards or practices for the different safety-critical market and technological domains is a challenging task. There are a variety of definitions of evidence, and how to evaluate it or derive it in regard the technology used, which makes cross-acceptance difficult.

System's safety is usually demonstrated by compliance to standards, processes, or generally accepted checklists [2]. In some industries, manufacturers are required to produce argumentation in the form of an explicit safety case, in order to demonstrate that all of the hazards have been prevented or mitigated and that the system is acceptably safe to operate in its intended context of use [9]. These argumentations are not just part of a set of requirements defined by standards. In fact a safety engineer must assure that all evidences are made explicit in order to have a confidence level enough to determine that a system is safe.

We have design a platform in order to give an answer and support a feasible approach to deal with all the mentioned issues. Fig 1 shows a general view of the functional decomposition of our conceptual platform. Our conceptual framework contains the following functionalities:

- **Prescriptive Knowledge Management:** Functionality related to the management of standards information as well as any other information derived from them, such

as interpretations about intents, mapping between standards, etc. This functional group maintain a knowledge database about "standards & understandings".

- **Assurance Project Lifecycle Management:** This functionality factorizes aspects such as the creation of safety assurance projects. This module manages a "project repository", which can be accessed by the other modules.
- **Safety Argumentation Management:** This group manages argumentation information in a modular fashion. It also includes mechanisms to support compositional safety assurance, and assurance patterns management.
- **Evidence Management:** This module manages the full life-cycle of evidences and evidence chains. This includes evidence traceability management and impact analysis. In addition, this module is in charge of communicating with external engineering tools (requirements management, implementation, V&V, etc.)

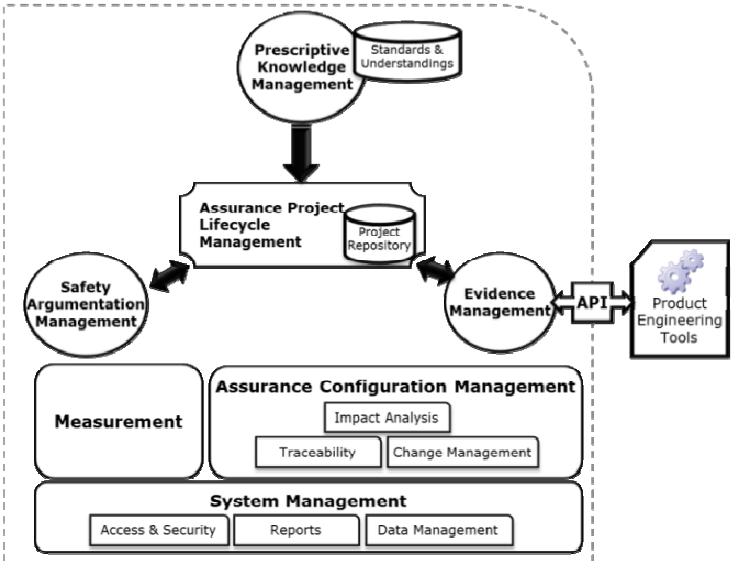

Fig. 1. Functional decomposition of our platform

- **Assurance Configuration Management:** This is an infrastructure functional module. This includes functionality for traceability management, change management, impact analysis.
- **System Management:** It includes generic functionality for security, permissions, reports, etc.
- **Measurement:** This module contains functionality related to safety indicators. Our first approach is based on basic indicators such as Mean Time Between Failures (MTBF).

4 Avionics Experience

It will be taken the example of Execution Platform (Computing Unit and Operating System) to build a scenario where complete Execution Platform will be installed in an IMA (Integrated Modular Avionics) platform using the aforementioned tool support. The execution platform is considered as an independent item for which a qualification dossier will be built. This qualification dossier consists of plans, technical documents, and certification documents. Technical documents are specifications, validation and verification life cycle data. The certification documents are configuration index documents and accomplishment summaries.

The execution needs to comply with DO178c [17] standard and ARP 4754 [18] guidelines. In addition the resulting software will be integrated into an IMA (Integrated Modular Avionics) platform [19].

Fig 2 depicts a process flow representing our tool platform. Each swim lane has its own editor and they support a predefined set of activities which has been tested for avionics software. The Standards editor, assurance projects editor, argumentations editor and evidences specification tool are closely integrated.

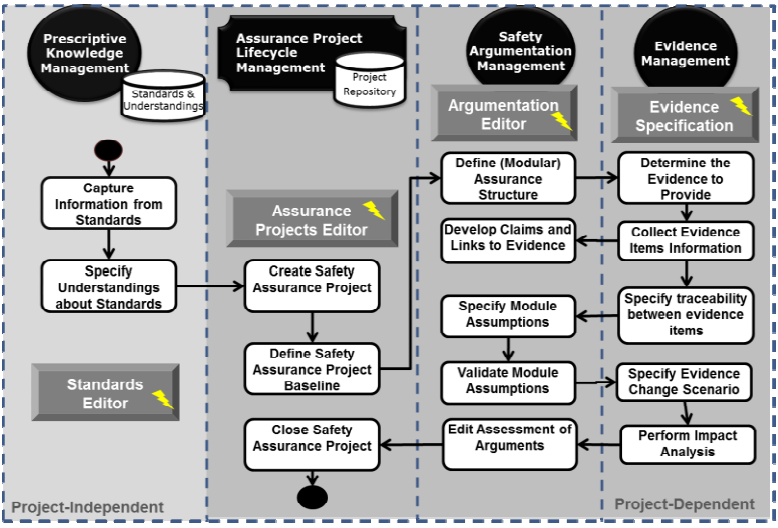

Fig. 2. Typical scenario for the tools

First we model our selected standards during the preliminary phase. The standard's editor offers services to retrieve, to digitalize and to store standards, recommendations, compliance means, intents and interpretations. All these standards are part of the so-called "reference framework". This reference framework is created just once, when the tools are deployed into the company. The reference framework of our case study includes the DO178c standard and the ARP 4754 guidelines.

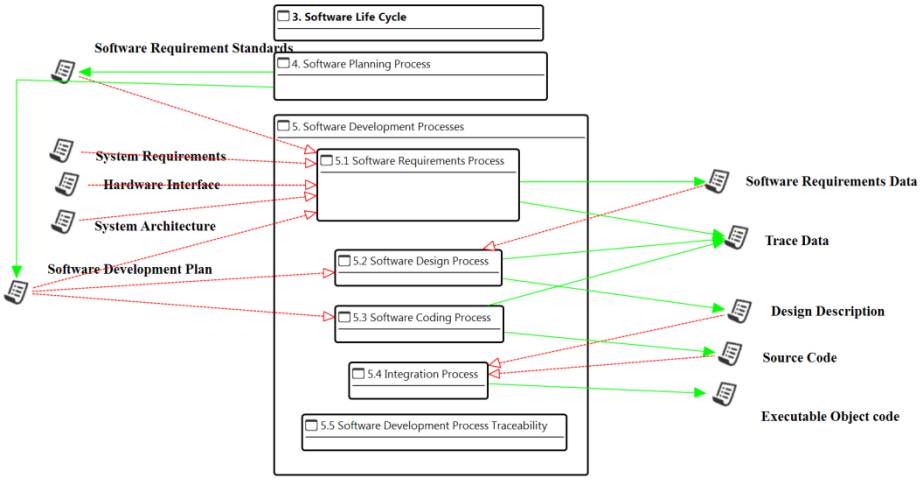

Fig. 3. Excerpt of the standard DO178C modelled with our tool

Once we have defined our reference framework, we need to define an assurance project. The new project is linked to the mentioned reference framework and it can be tailored to specific project requirements. One of the changes that can be done at this phase is defining which specific tools will be used on a defined activity just in the scope of this project or the role involved of a specific activity.

During the safety argumentation phase the argumentation editor is used to define an argumentation model compliant to SACM [7] using the GSN graphical notation [10]. Argumentation deals with (a) direct technical arguments of safety, required behavior from components, (b) compliance arguments about how prevailing standard has been sufficiently addressed and (c) backing confidence arguments about adequacy of arguments and evidence presented (e.g. sufficiency of Hazard and Risk Assessment). In order to support the argumentation creation, the arguments related to the standard compliance are automatically generated from the information selected on the baseline.

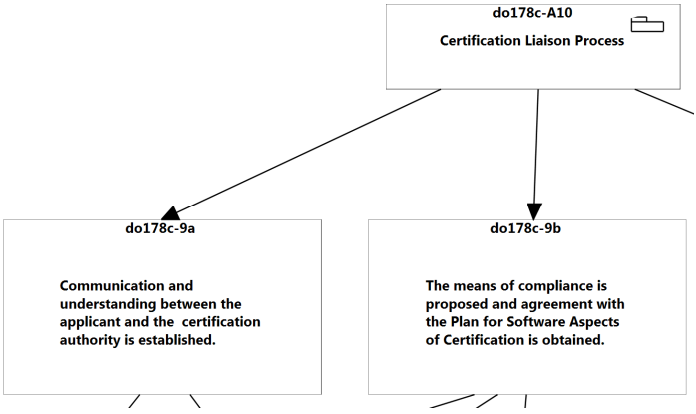

Fig. 4. Excerpt of the compliance argumentation

On the argumentation editor, we offer the possibility to take advantage of best practices by using argumentation patterns. The argumentation editor is able to re-use predefined patterns just by "drag and drop" the pattern into the working area. The use of the SACM model provides a semi-formal way for structuring.

In our avionics use case, our assurance case refers to required data such the PSAC (Plan for Software Aspects of Certification) or SAS (Software accomplishment summary) which are used as evidence for a certification process. However we do not only link these documents with the pieces of argumentation that they support but also to trace their evolution and evaluate our confidence on their safety. In addition we have implemented the following functionalities:

- Evidence storage: it provides a mean to determine, specify, and structure evidence. Evidence can be stored either locally on the system or on any revision management system as Subversion.
- Evidence traceability: it offers the possibility to specify and maintain the evidence relationships, like the relation between a specific document used as evidence and all the versions of that document that evolution thought the project lifecycle, of the relation between evidence and how it is used to support a specific claim. We are able to trace the evidence(s) used to comply with a specific requirement on one standard
- Evidence evaluation: we keep track of the evidence assessment for completeness and adequacy.

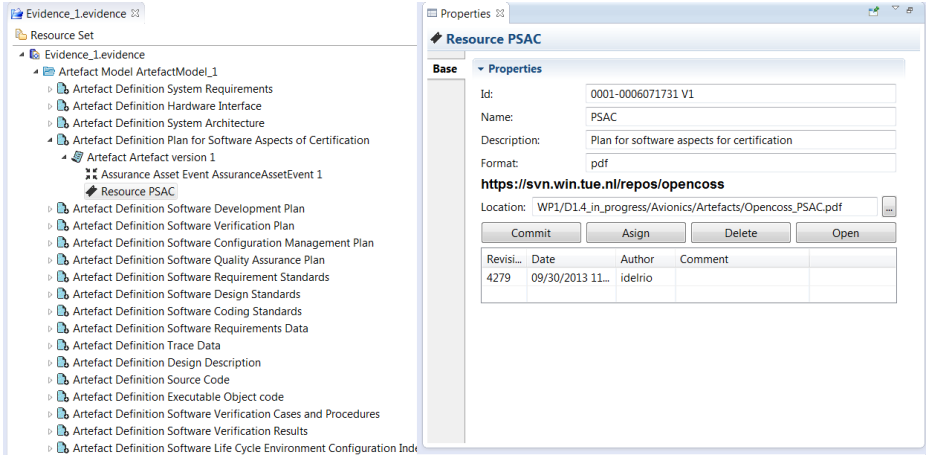

Fig. 5. View of the evidence model

We have also used the compliance maps functionality in order to define which and how all pieces of evidences stored do comply with the different aspects of the standard as it was capture on the reference framework. As a result of this we are able to show the compliance report.

5 Benefits from Using this Platform

We have used this platform for the implementation of differen case sutides along the Opencoss project. As result of users interviews we have identified the following set of benefits resulting from the use of this approach:

- **Centralized management of safety assurance assets.** Our tool infrastructure traces evidences with certification requirements
- **The Safety Case concept provides a comprehensible compilation of safety argumentation and evidence.** This approach promotes safety certification as a judgment based on a body of material that, explicitly, should consist of three elements: claims, evidence, and argument. To this end, we need to be able to propagate satisfaction from the fine-grained claims arrived at through decomposition to the higher-level claims. Supporting such propagation first and foremost requires elaborating the decomposition strategies to be used in different domains.
- **Harmonized and synchronized agreements in interpretations.** Without an upfront agreement between the system supplier/OEM and the certifier/assessor about the details of the arguments/evidences that need to be collected, there will invariably be important omissions, which need to be remedied after the fact and at significant costs. The presented tool suite support for negotiating detailed agreements about the required arguments/evidences to avoid unnecessary cost overheads during certification. This is achieved by exhaustively going through the concepts and their relations in the (abstract) arguments/evidences specifications for the standards and specializing these concepts and relations according to the needs of the underlying system.

6 Future work and Conclusions

Awareness of compliance and the certification process are some of the most expensive activities in a safety critical context. Cost-efficient system certification demands a continuous compliance-checking process by enhancing integration of certification goals and development workflow. The goal is to provide engineers with guidance about how to comply with standards and regulations and allow developers to assess where they are with respect to their duties to conform to safety practices and standards.

Our tool provides a centralized management of safety assurance assets. This tool infrastructure allow faster certification by automating most of the activities required for certification, so every change triggers a complete run of these activities, signaling those that need to be performed manually. This also includes facilitating integration with state-of-the-art engineering tools (e.g., DOORS, Simulink, safety analysis tools, etc.). In addition we provide a comprehensible compilation of safety argumentation and evidence. A key aspect of the certification language to be developed in a near future is to define the semantics of an argumentation language. We also need to support compositional certification by the use of a contract based approach and the possibility to validate the content of these contracts during runtime.

Acknowledgment. The research leading to these results has received funding from the FP7 programme under grant agreement n° 289011 (OPENCOSS) and nº608945 (Safe Adapt). We would also like to mention Angel López, Idoya del Río and Mª Carmen Palacios from Tecnalia for their effort developing some of the functionalities we have explained.

References

1. Hawkins, R., Habli, I., Kelly, T., McDermid, J.: Assurance cases and prescriptive software safety certification: A comparative study. Saf. Sci. **59**, 55–71 (2013)
2. Dodd, I., Habli, I.: Safety certification of airborne software: An empirical study. Reliab. Eng. Syst. Saf. **98**(1), 7–23 (2012)
3. Wilson, A., Preyssler, T.: Incremental certification and integrated modular avionics. In: 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, pp. 1.E.3–1–1.E.3–8 (November 2008)
4. Vinodkumar, M.N., Bhasi, M.: A study on the impact of management system certification on safety management. Saf. Sci. **49**(3), 498–507 (2011)
5. Baumgart, S., Froberg, J., Punnekkat, S.: Towards efficient functional safety certification of construction machinery using a component-based approach. In: 2012 Third International Workshop on Product LinE Approaches in Software Engineering (PLEASE), pp. 1–4 (2012)
6. Defence Standard 00-56, Safety Management Requirements for Defence Systems, Issue 4, Part 1: Requirements, Ministry of Defence, Glasgow, UK (2007)
7. OMG, Structured Assurance Case Metamodel, (SACM) (2013)
8. Larrucea, X., Combelles, A., Favaro, J.: Safety-Critical Software [Guest editors' introduction]. IEEE Softw. **30**(3), 25–27 (2013)
9. Basir, N., Denney, E., Fischer, B.: Deriving Safety Cases for the Formal Safety Certification of Automatically Generated Code. Electron. Notes Theor. Comput. Sci. **238**(4), 19–26 (2009)
10. Goal Structuring Notation Working Group, GSN Community Standard (November 2011). Retrieved from http://www.goalstructuringnotation.info
11. Adelard, L.: (n.d.). Claims, Arguments and Evidence. Retrieved from http://www.adelard.com/asce/choosing-asce/cae.html
12. OPENCOSS project, D6.2_Detailed requirements for evidence management of the OPENCOSS platform_final (November 2012)
13. Denney, E., Pai, G., Pohl, J.: AdvoCATE: an assurance case automation toolset. In: Ortmeier, F., Daniel, P. (eds.) SAFECOMP Workshops 2012. LNCS, vol. 7613, pp. 8–21. Springer, Heidelberg (2012)
14. Matsuno, Y., Takamura, H., Ishikawa, Y.: A dependability case editor with pattern library. In: HASE, pp. 170–171 (2010)
15. Blanc, X., Gervais, M.-P., Sriplakich, P.: Model Bus: Towards the Interoperability of Modelling Tools. In: Aßmann, U., Akşit, M., Rensink, A. (eds.) MDAFA 2003. LNCS, vol. 3599, pp. 17–32. Springer, Heidelberg (2005)
16. Górski, J., Jarzębowicz, A., Miler, J., Witkowicz, M., Czyżnikiewicz, J., Jar, P.: Supporting assurance by evidence-based argument services. In: Ortmeier, F., Daniel, P. (eds.) SAFECOMP Workshops 2012. LNCS, vol. 7613, pp. 417–426. Springer, Heidelberg (2012)
17. RTCA DO-178/EUROCAE ED-12, Software Considerations in Airborne System and Equipment Certification
18. SAE ARP4754/EUROCAE ED-79, Certification Considerations for Highly Integrated or Complex Aircraft Systems
19. RTCA DO-297/EUROCAE ED-124 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations