The AMASS Approach for Assurance and Certification of Critical Systems

Jose Luis de la Vara
Computer Science and Engineering Department
Universidad Carlos III de Madrid
Leganes, Spain
jvara@inf.uc3m.es

Alejandra Ruiz
ICT Division
TECNALIA Research and Innovation
Derio, Spain
alejandra.ruiz@tecnalia.com

Barbara Gallina
Division of Computer Science and Software Engineering
Mälardalen University
Västerås, Sweden
barbara.gallina@mdh.se

Gaël Blondelle
Eclipse Foundation Europe GmbH
Zwingenberg, Germany
gael.blondelle@eclipse-foundation.org

Elena Alaña
GMV Aerospace and Defence
Madrid, Spain
ealana@gmv.com

Javier Herrero
GMV Aerospace and Defence
Madrid, Spain
jaherrero@gmv.com

Fredrik Warg
Department of Electronics
RISE Research Institutes of Sweden
Borås, Sweden
fredrik.warg@ri.se

Martin Skoglund
Department of Electronics
RISE Research Institutes of Sweden
Borås, Sweden
martin.skoglund@ri.se

Robert Bramberger
VIRTUAL VEHICLE Research Cente
Graz, Austria
robert.bramberger@v2c2.at

***Abstract*—Safety-critical systems are subject to rigorous assurance and certification processes to guarantee that they do not pose unreasonable risks to people, property, or the environment. The associated activities are usually complex and time-consuming, thus they need adequate support for their execution. The activities are further becoming more challenging as the systems are evolving towards open, interconnected systems with new features, e.g. Internet connectivity, and new assurance needs, e.g. compliance with several assurance standards for different dependability attributes. This requires the development of novel approaches for cost-effective assurance and certification. With the overall goal of lowering assurance and certification costs in face of rapidly changing features and market needs, the AMASS project has created and consolidated the de-facto European-wide open solution for assurance and certification of critical systems. This has been achieved by establishing a novel holistic and reuse-oriented approach for architecture-driven assurance, multi-concern assurance, and for seamless interoperability between assurance and engineering activities along with third-party activities. This paper introduces the main elements of the AMASS approach and how to use them and benefit from them.**

Keywords—AMASS; assurance; certification; safety-critical systems; cyber-physical systems; platform; ecosystem; community

I. INTRODUCTION

Safety-critical systems can be defined as computer-based systems that in case of an incident or misbehaviour can lead to an accident that will put people or the environment in danger, resulting in injuries or casualties [14]. This kind of systems must go through very intensive verification and validation (V&V) activities in order to assure the acceptable safety of the

systems and as a final result to assess or certify them, providing sufficient and relevant evidence [15]. Assurance can be defined as the set of planned and systematic actions necessary to provide adequate confidence and evidence that a system satisfies given requirements, e.g. for system safety, and certification can be defined as the legal recognition that a system complies with standards and regulations designed to ensure that the system can be depended upon to deliver its intended service [23].

Assurance and certification of safety-critical systems require the execution of complex and labour-intensive activities [10], e.g. the management of compliance with hundreds or thousands of criteria defined in standards, the management of a high number of evidence artefacts, or the provision of convincing and valid justifications that a system is dependable. Therefore, system manufacturers and component suppliers need approaches that facilitate these activities and ideally increase their efficiency. The challenges arising from system assurance and certification are further growing as a result of the evolution of safety-critical systems.

For example, embedded systems have significantly increased in number, technical complexity, and sophistication towards open, interconnected, networked systems such as "the connected car". This has brought a “cyber-physical” dimension with it, exacerbating the problem of ensuring safety, as well as other dependability concerns such as security, robustness, and reliability, in the presence of human, environmental, and technological risks. The rise of notions such as cyber-physical systems and their complexity are leading to the need for new approaches for system assurance and certification. In general, practitioners expect improvements in the available support for assurance and certification with new methods and tools [9],[16].

Within this context, the AMASS project [1] has worked on the improvement of the assurance and certification practices for the new generation of critical systems, and more concretely of critical systems that correspond cyber-physical ones. AMASS stands for Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems, and its consortium consists of 29 organizations from 8 countries, including large enterprises, SMEs, and research institutions, and covering the whole supply chain for assurance and certification (system manufacturers, component suppliers, tool vendors, assessors, certification authorities, and regulation developers). AMASS directly addresses the assurance and certification needs in aerospace, automotive, industrial automation, railway, and space.

The ultimate goal of AMASS is to lower certification costs for critical systems in face of rapidly changing features and market needs. To this end, the project has created and consolidated the de-facto European-wide open tool platform, ecosystem, and sustainable community for assurance and certification. The platform is the implementation of a novel holistic and reuse-oriented approach for Architecture-Driven Assurance (fully compatible with standards such as SysML), Multi-Concern Assurance (for co-analysis and co-assurance of e.g. security and safety aspects), and Seamless Interoperability between assurance and engineering activities along with third-party activities (e.g. external assessments and supplier assurance).

In addition, to ensure maintenance and sustainability of its results, AMASS has created an open source community and ecosystem around which its main results are publicly available. This way, different stakeholders can get involved in the community and contribute to the further use and development of the AMASS approach.

To present the approach, we use examples from two of the 11 industrial case studies that have been used in the AMASS project to validate and evaluate its results [2]: design and safety assessment of on-board software applications in space systems, and a telematics function for automated and connected vehicles.

The rest of the paper is organized as follows. Section II presents the overall process underlying the AMASS approach, whereas Section III presents the main tool features that support the approach. Section IV describes the open source ecosystem that has been developed around AMASS. Finally, Section V presents our main conclusions.

II. Overall Process of the AMASS Approach

From a process perspective, six main stages can be distinguished in the AMASS approach (Fig. 1). Not every stage and step should be performed for each assurance project. In particular, the first two stages (“Standards Compliance Definition” and “Process Reusability Definition”) are project-independent and only need to be performed once, so the outcome and data provided from these steps could be re-used for multiple projects.

Standards Compliance Definition is a project-independent phase focused on capturing, digitalizing, storing and retrieving the different standard compliance knowledge. It should be performed by an expert in the regulatory frameworks that will be part of the reference knowledge included in the platform.

Process Reusability Definition is conducted only once by a process expert. This expert will take care of tasks such as specifying reusable compliant processes and validating the process reusability.

For **Assurance Project Definition**, the assurance manager defines the scope of compliance for a project in the context of a certain regulation. The manager will follow the project compliance lifecycle and, when it is feasible, check the different reuse possibilities and compliance means.

The systems engineer performs **System Design Analysis and V&V** in collaboration with the safety and security engineers to define the system architecture, elicit system requirements, define component contracts, and conduct safety and security analyses. The validation of the components’ contracts and V&V of safety and security analyses is performed by the V&V engineer.

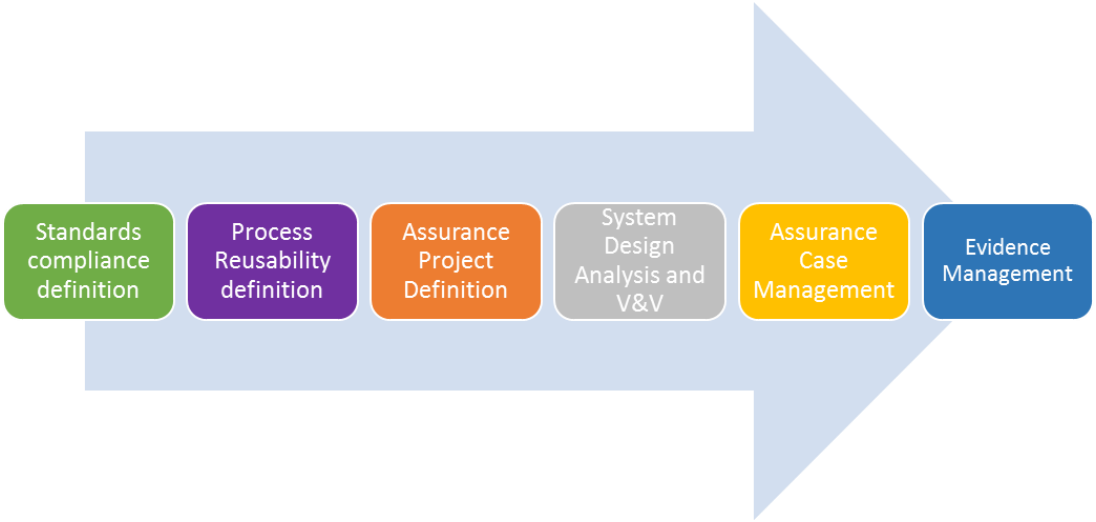


Fig. 1. Main stages of the AMASS approach [4].

Assurance Case Management deals with the definition of argumentation using compliance arguments and product arguments. The assurance manager will take care of resolving safety and security trade-offs and of linking the assurance case information to the system architecture.

During **Evidence Management**, the assurance manager will define the project artefacts that will be used as evidence and collect those artefacts. The manager will ensure artefact traceability, follow the progress of the process execution, and specify the compliance with standards and regulations.

More information about how to conduct the stages is presented in the next section, which introduces the tool support for the AMASS approach.

III. Main Tool Features of the AMASS Approach

From a tool perspective, the AMASS approach can be divided into different tool features or functional areas. An overview of this features is shown in Fig. 2. The AMASS Reference Tool Architecture [3] is the conceptual result of this set of features and their relationships.

The corresponding software tool to support assurance and

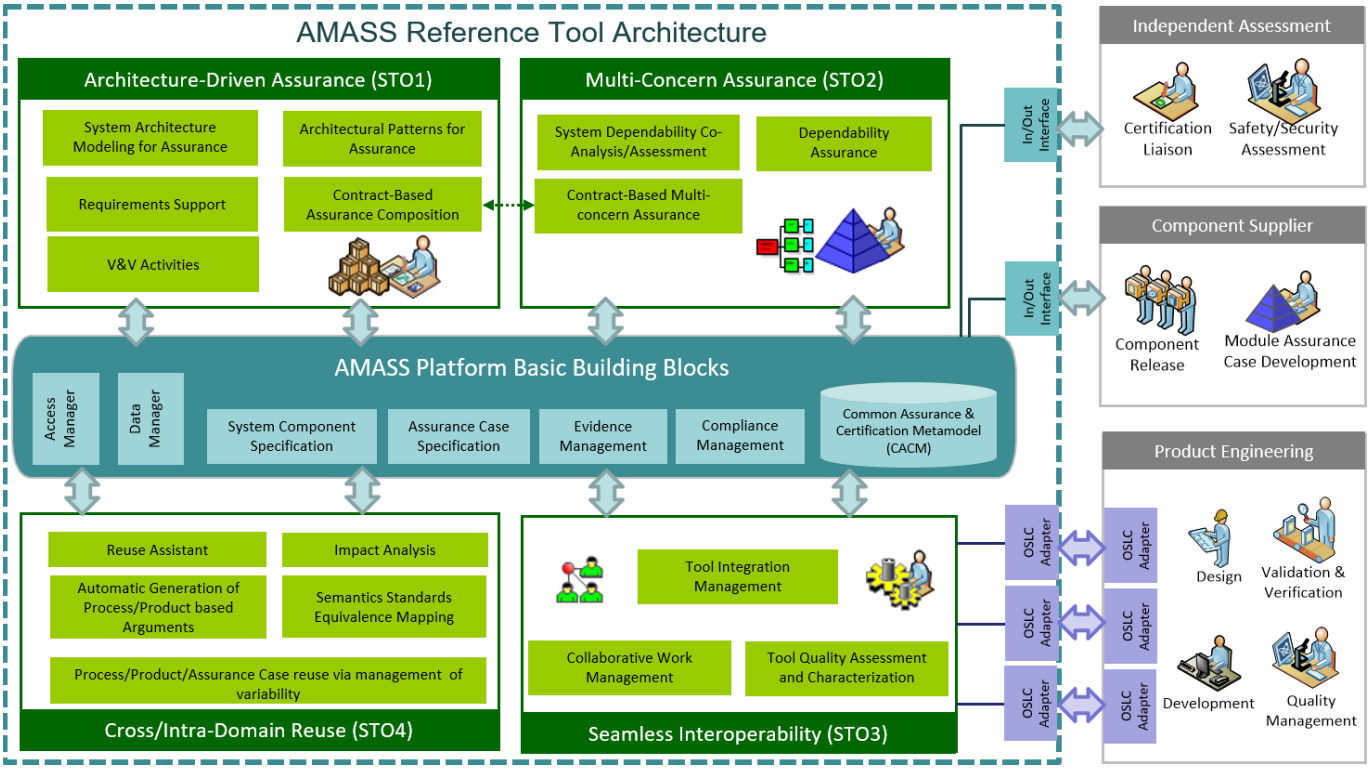


Fig. 2. Overview of the main tool features of the AMASS approach [3].

certification of critical systems, referred to as the AMASS Tool Platform, has been implemented by integrating and extending different existing tools, most notably:

- OpenCert [17] for assurance- and certification-specific activities, e.g. evidence management and assurance cases specification.
- The CHESS toolset [22], which is based on the Papyrus tool [19], for model-driven, component-based, and contract-based development of high-integrity systems.
- EPF Composer [12] for systems and software process engineering, as well as for compliance management.
- The BVR tool [5] for variability management, including feature modelling, resolution, and realization, and derivation of system asset families.

In addition, the AMASS Tool Platform is integrated with over 20 external tools with which the Platform exchanges data and that provide additional services for assurance and certification, e.g. for system artefact quality analysis [25].

The resulting tool support can be used in different specific usage scenarios, e.g. architecture refinement, safety and security co-assessment, process and product compliance and configuration, and toolchain for system specification and quality assessment [4].

The main tool features are presented in the next subsections, describing the blocks that are part of the different areas.

A. *Basic Building Blocks*

This area includes blocks for core support for an assurance and certification process, e.g. for compliance and evidence management. In addition, they provide support for access management and data management, and a Common Assurance and Certification Metamodel (CACM) as a reference data model for an assurance project.

System Component Specification provides features for system architecture modelling; in particular, to allow the definition of components as reusable entities, and then the assembly of the components themselves, at any level of the hierarchical architecture, to build or decompose a system.

Assurance Case Specification manages argumentation information in a modular fashion. It also includes mechanisms to support compositional assurance and assurance patterns management.

Evidence Management deals with the full lifecycle of evidence artefacts and evidence chains. This includes evidence traceability management and impact analysis.

Compliance Management addresses the management (edition, search, transfer, etc.) of process and standards' information as well as of any other information derived from them, such as interpretations about intents and mappings between processes and standards. This functional group maintains a knowledge database about standards and processes, which can be consulted by other AMASS features.

Examples of the use of these features are shown in Fig 3 and Fig. 4. Fig. 3 shows a SysML block diagram of a space software system, as an example of the use of System Component Specification. Fig. 4 shows an excerpt of the model of the safety standard applicable to the telematics function (ISO 26262), as an example of use of Compliance Management.

B. *Architecture-Driven Assurance*

This area supports the integration of assurance and certification activities with the development activities of critical systems, including specification and design. It provides support for system component composition in accordance with the domain best practices, guaranteeing that emerging behaviour does not interfere with the whole system assurance.

System Architecture Modelling for Assurance contains

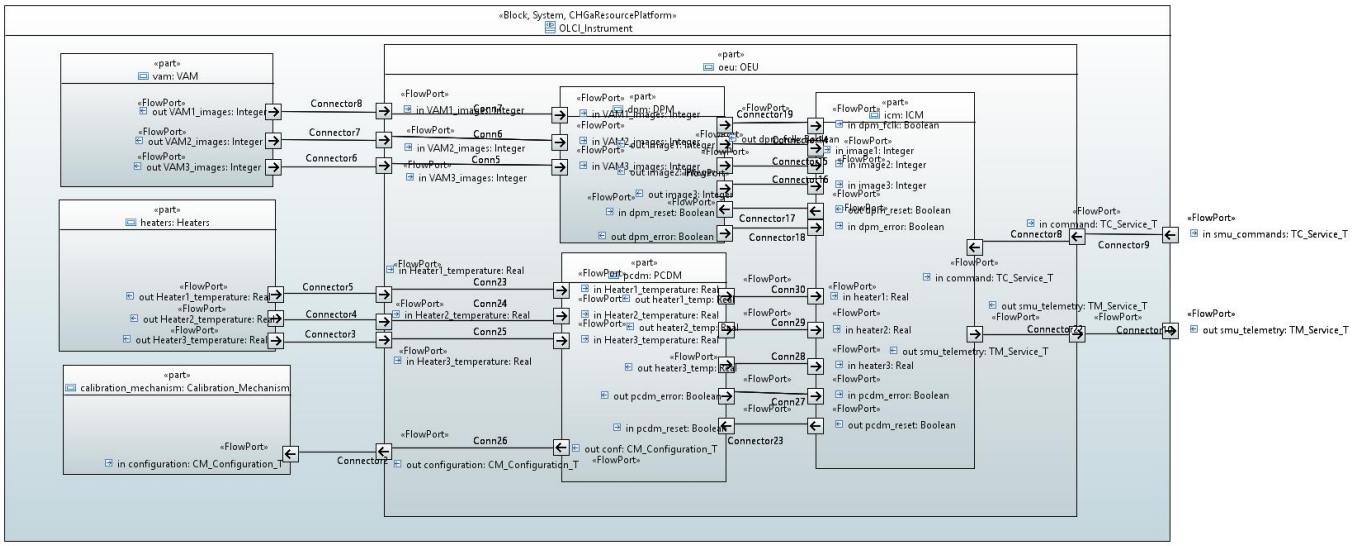


Fig. 3. Block diagram of a space software system.

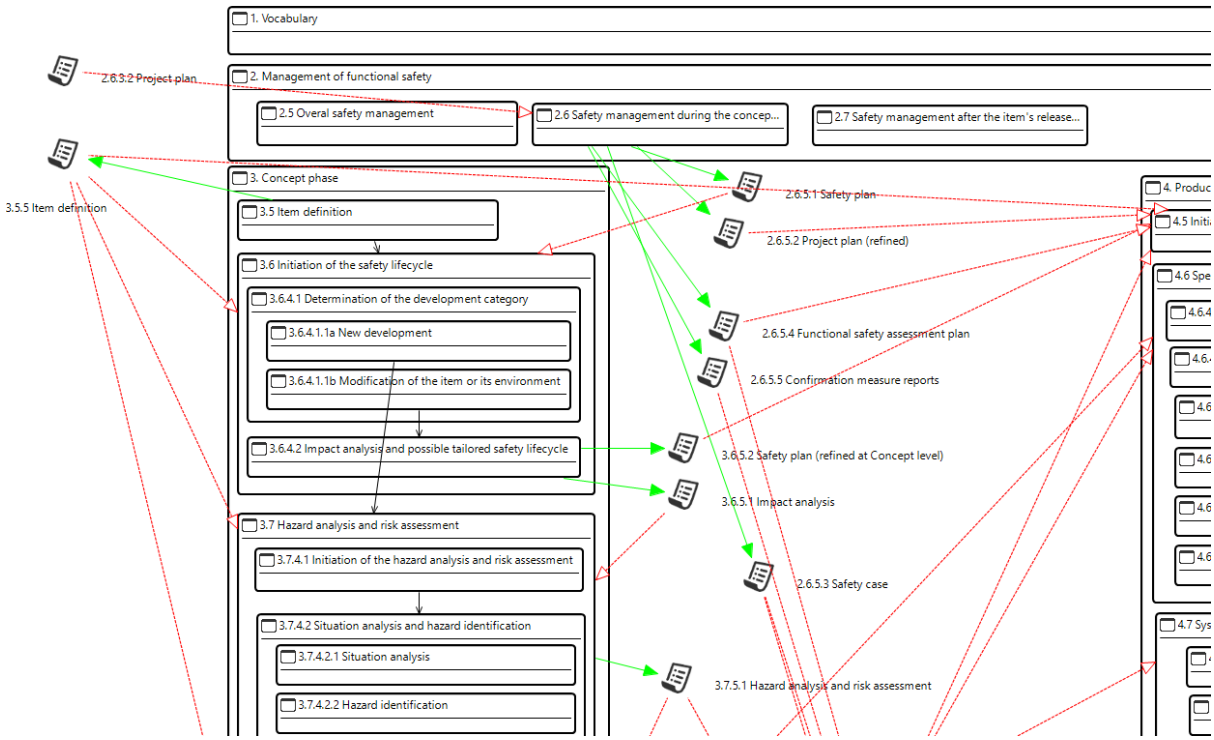


Fig. 4. Excerpt of an ISO 26262 model for an automotive telematics function.

the functionalities that are focused on the modelling of the system architecture to support the system assurance. This includes supporting the modelling of additional aspects (not already included in the system component specification) related to the system architecture, tracing the elements of the system architecture model to the assurance case, generating evidence for the assurance case, and functional refinement.

Architectural Patterns for Assurance helps designers and system architects when choosing suitable solutions for commonly recurring design problems while achieving component reuse. This block contains the functionality for management of a library of architectural patterns, automated application of specific architectural patterns, and generation of assurance arguments from architectural patterns application.

Contract-Based Assurance Composition provides the functionality that supports the contract-based design of the system architecture, which provides additional arguments and evidence for system assurance. The functionality includes contract specification, i.e. specification of components' assumptions and guarantees, contract-based reuse of components, i.e. a component reuse that is supported by checks on the contracts, and generation of assurance arguments from the contract specification and validation.

Requirements Support contains features focused on enriching the assurance case with advanced analysis to support the evidence of the assurance case. This includes requirements formalisation into temporal logics, analysis of requirements' semantics based on their formalisation into temporal logics, analysis of requirements based on quality metrics, and safety requirements derivation.

V&V Activities deals with enriching the assurance case concerning V&V information to support the evidence of the assurance case. The functionality supports contract-based verification, automated formal verification of requirements on the system design, model-based specification of fault-injection and analysis of faulty scenarios, other techniques for model-based safety analysis such as Component Fault Trees from SysML models, and document generation.

As a usage examples, Fig. 5 shows the specification of a contract for the space software system. The specification includes assumptions and guarantees.

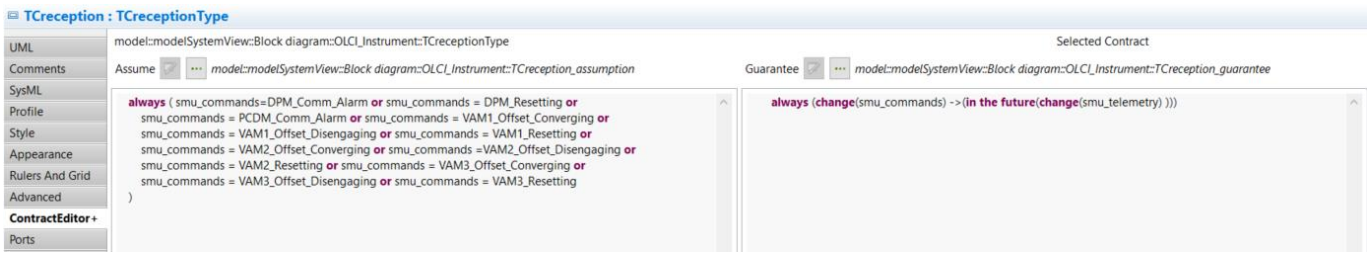


Fig. 5. Contract specification for a block of a space software system.

C. Multi-Concern Assurance

This area provides a tool-supported methodology for the development of assurance cases, co-assessment, and contract-based assurance. It addresses multiple system characteristics, mainly safety and security, but also other dependability attributes such as robustness and reliability.

Dependability Assurance contains the functionality to create and structure the multi-concern assurance case argumentation in an understandable and maintainable way. This includes argumentations targeting various dependability attributes with support of argumentation patterns.

System Dependability Co-Analysis/Co-Assessment provides functionalities to analyse different quality attributes while taking care of the inter-dependences between them. This is ideally realized by inherently combined co-analysis and co-assessment methods, which take care of the inter-dependencies within the method. On the other hand, Multi-Concern Assurance can be implemented by combining separate mono-concern assurance processes through interaction points to treat the mutual dependencies between the quality attributes.

Contract-Based Multi-Concern Assurance comprises functionality that contributes to assurance for multiple concerns via two kinds of contracts: component contracts, which target more than one quality attribute, and argument contracts, which provide a means to realise a link between related assurance cases.

Fig. 6 shows an example of Multi-Concern Assurance, and more concretely of Dependability Assurance, in the form of an assurance case for acceptable safety, acceptable security, and adequate measures to find and resolve conflicts between the quality attributes of the automotive telematics function.

D. Seamless Interoperability

This area enables an open and generically applicable approach to ensure the interoperability between the tools used in the modelling, analysis, and development of critical systems, among other possible engineering activities. Interoperability is particularly important from an assurance and certification-specific perspective because of all the different tools that are typically involved in the engineering process of a critical system, and as an enabler of collaborative work among the different stakeholders of the assurance and certification process.

Tool Integration Management enables the exchange of data between engineering and assurance tools, e.g. between the AMASS Tool Platform and other tools developed by the AMASS partners.

Collaborative Work Management allows different users to work at the same time with the same pieces of data, supporting the interaction of the different users.

Tool Quality Assessment and Characterisation supports the specification and management of tool quality needs for assurance and certification. It is currently supported by the Compliance Management functionality; i.e. tool qualification is managed as a specific case of compliance management, as it will be based on requirements from some assurance standard and their satisfaction will have to be declared.

As a usage example of Seamless Interoperability, Fig. 7 shows results of contract refinement verification imported into the AMASS Tool Platform for the space software system, by using Tool Integration Management.

E. Cross- and Intra-Domain Reuse

This area deals with the consistent assistance to reuse assurance information in a domain or across domains, based

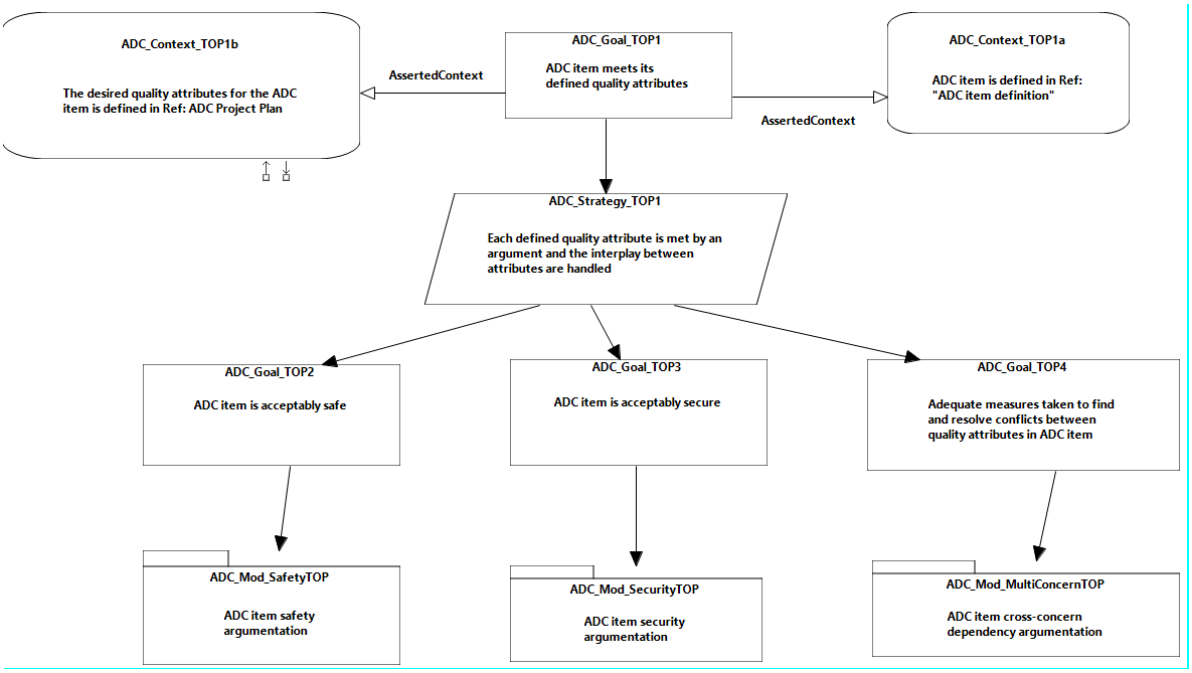
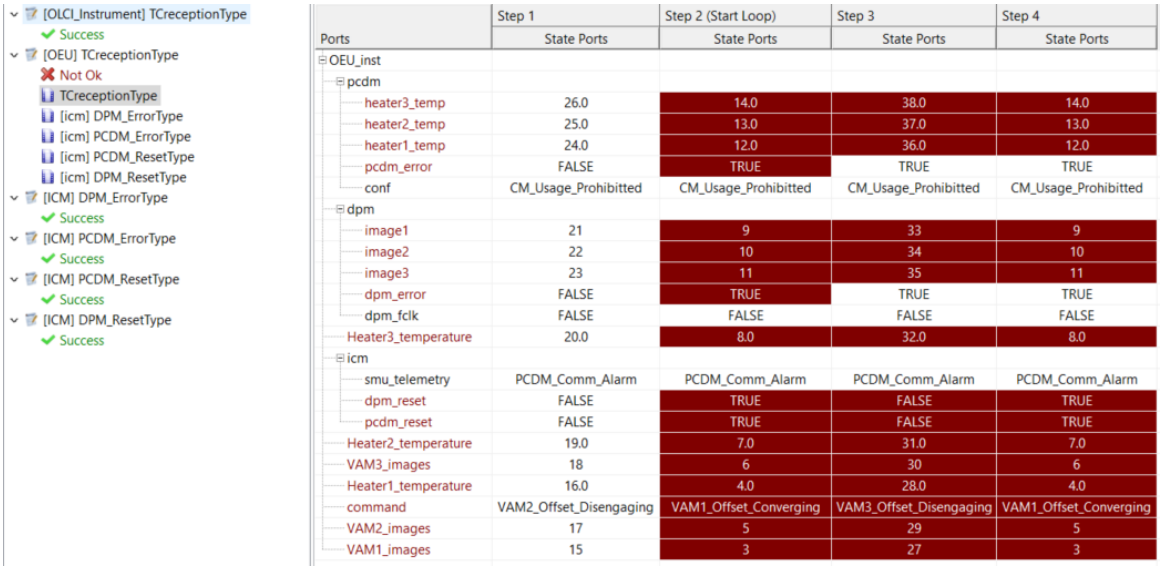


Fig. 6. Assurance case structure for Dependability Assurance of an automotive telematics function.



Ports	Step 1 State Ports	Step 2 (Start Loop) State Ports	Step 3 State Ports	Step 4 State Ports
OEU_inst				
pcdm				
heater3_temp	26.0	14.0	38.0	14.0
heater2_temp	25.0	13.0	37.0	13.0
heater1_temp	24.0	12.0	36.0	12.0
pcdm_error	FALSE	TRUE	TRUE	TRUE
conf	CM_Usage_Prohibitted	CM_Usage_Prohibitted	CM_Usage_Prohibitted	CM_Usage_Prohibitted
dpm				
image1	21	9	33	9
image2	22	10	34	10
image3	23	11	35	11
dpm_error	FALSE	TRUE	TRUE	TRUE
dpm_fclk	FALSE	FALSE	FALSE	FALSE
Heater3_temperature	20.0	8.0	32.0	8.0
icm				
smu_telemetry	PCDM_Comm_Alarm	PCDM_Comm_Alarm	PCDM_Comm_Alarm	PCDM_Comm_Alarm
dpm_reset	FALSE	TRUE	FALSE	TRUE
pcdm_reset	FALSE	TRUE	FALSE	TRUE
Heater2_temperature	19.0	7.0	31.0	7.0
VAM3_images	18	6	30	6
Heater1_temperature	16.0	4.0	28.0	4.0
command	VAM2_Offset_Disengaging	VAM1_Offset_Converging	VAM3_Offset_Disengaging	VAM1_Offset_Converging
VAM2_images	17	5	29	5
VAM1_images	15	3	27	3

Fig. 7. Results from contract refinement verification imported to the AMASS Tool Platform for a space software system.

on a conceptual framework to specify and manage assurance and certification assets.

Reuse Assistant concerns cross- and intra-domain reuse of assurance and certification assets. This module helps users to understand whether reuse of the assurance assets is reasonable or to determine what further assurance activities are required to justify compliance in the new scenario, e.g. engineering, V&V, or compliance activities.

Automatic Generation of Process- and Product-Based Arguments is related with the generation of structured arguments from process and product models; in the latter, from contract-based architectural specification. It supports the strengthening of the assurance case via arguments that are aimed at explaining why a process is compliant or at showing why the product is expected to behave dependably.

Impact Analysis tackles assurance asset changes, indicating how an assurance asset's change impacts related assurance assets and providing semi-automated support for change management.

Semantics Standards Equivalence Mapping addresses the correspondence between assurance standards according to their semantics, supporting an informed gap analysis of the standards and thus mitigating the risk of inappropriate reuse across standards.

Process-, Product, and Assurance Case-Related Reuse via Management of Variability deals with the management of variability at process, product, and assurance case levels. This functionality takes as input a process, a product (more specifically, an architectural specification), or an assurance case that needs to be tailored or reconfigured, and as outcome it generates a new valid re-configuration of the input asset.

As an example, Fig. 8 shows a feature resolution diagram for automotive that has been developed for Process-Related Reuse via Management of Variability in the scope of the ISO 26262 standard.

IV. The AMASS Open Source Ecosystem

The AMASS open source ecosystem corresponds to what Jansen et al. [13] describe as "a set of actors [research institutions, tool vendors, manufacturers, component suppliers, assessors, and tools] functioning as a unit and interacting with a shared market", for assurance and certification and for the development of an open source tool platform.

From the AMASS inception, the objective of the consortium was to create a large open ecosystem by merging results from prior research projects such as OPENCOSS [18], SafeCer [24], CHESS [6], and CRYSTAL [7]. The resulting AMASS Tool Platform not only leverages existing open source components, but also publishes new open source components to the community. This is performed in the scope of the OpenCert project [17], as a part of the PolarSys top-level project [21] of Eclipse [11] (Fig. 9).

In order to enable the creation of a larger ecosystem, the AMASS project partners created the AMASS Tool Platform as a package of several Eclipse open source projects (OpenCert, CHESS, EPF Composer, and BVR) that can be used as a joint platform for new products and services. These open source projects are hosted by Eclipse and PolarSys with different life cycles. The AMASS open source ecosystem relies on the processes already defined by Eclipse and on the services and tools provided. The AMASS Tool Platform bundles the development from the different projects into a package that provides an integrated and extensible solution for assurance and certification of critical systems.

As a whole, the AMASS open source ecosystem consists of two main parts:

- The AMASS open source tool platform, which provides core features for assurance and certification of critical systems.
- Other tools with additional functionality either as Eclipse plug-ins, e.g. for system dependability analysis with Papyrus [20], or as external tools, e.g.

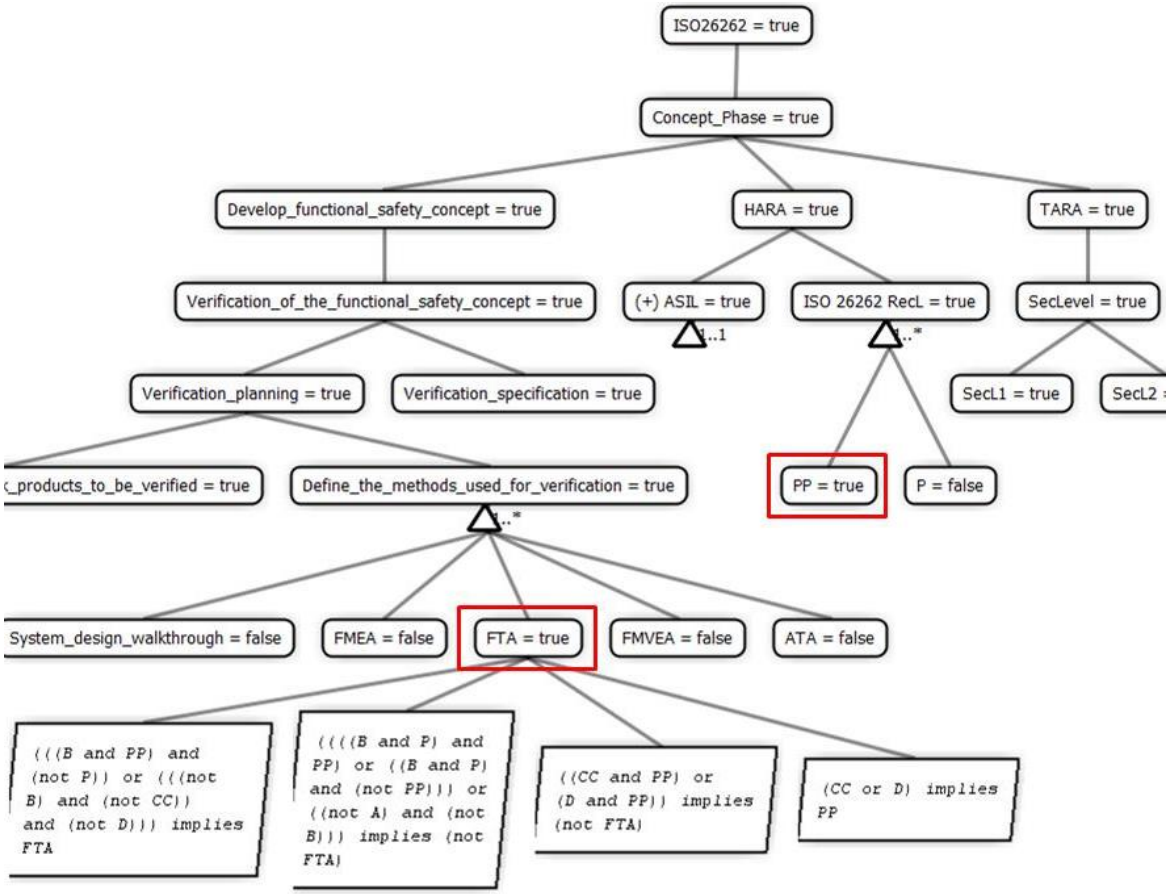


Fig. 8. Feature resolution diagram for an integrated safety and security process in automotive.

the Verification Studio tool [25] for analysis of system artefact quality. Interoperability with over 20 tools, most of which are commercial, have been ensured within the project [4].

The creation and management of a software ecosystem poses a series of challenges [8] related to the definition of open source contribution strategies, creation of partnering models, and creation of a developer community. For the AMASS open source ecosystem, we have relied on the overall strategies underlying the Eclipse approach for the creation and management of open source projects.

The technology and source code developed by the Eclipse community is made available royalty-free under the Eclipse Public License. This allows the development of commercial features on top of the open source solutions. The three main principles of open source projects in Eclipse and thus of the AMASS open source ecosystem are: transparency, so that all the decisions in a project are public; openness, so that participation is open to every individual without restriction, and; meritocracy, so that new committers are elected by their peers after publicly demonstrating their ability to contribute to the project.

V. CONCLUSION

The assurance and certification of safety-critical systems are complex activities that must be carefully performed. Acceptable system safety must be justified, supported by evidence, and addressed in compliance with applicable standards. These activities are further becoming more challenging as the systems are evolving, e.g. towards cyber-physical systems. and new assurance and certification needs arise, e.g. about security. Novel approaches are necessary so that assurance and certification of the new generation of critical systems is cost-effective.

In this paper we have presented the approach developed in the scope of the AMASS project for assurance and certification of critical systems. Thanks to the joint effort of 29 partners from 8 countries, we have been able to define an approach that can successfully tackle Architecture-Driven Assurance, Multi-Concern Assurance, Seamless Interoperability, and Cross- and Intra-Domain Assurance Reuse. The application of the approach is based on a specific process for Standards Compliance Definition, Process Reusability Definition, Assurance Project Definition, System Design Analysis and V&V, Assurance Case Management, and Evidence Management. The approach has also been developed as a part of an open source project, leading to the creation of the AMASS open source ecosystem. These elements have ultimately resulted in the de-facto European-wide open tool platform, ecosystem, and sustainable community for assurance and certification of critical systems in the largest industrial vertical markets including aerospace, automotive, industrial automation, railway, and space.

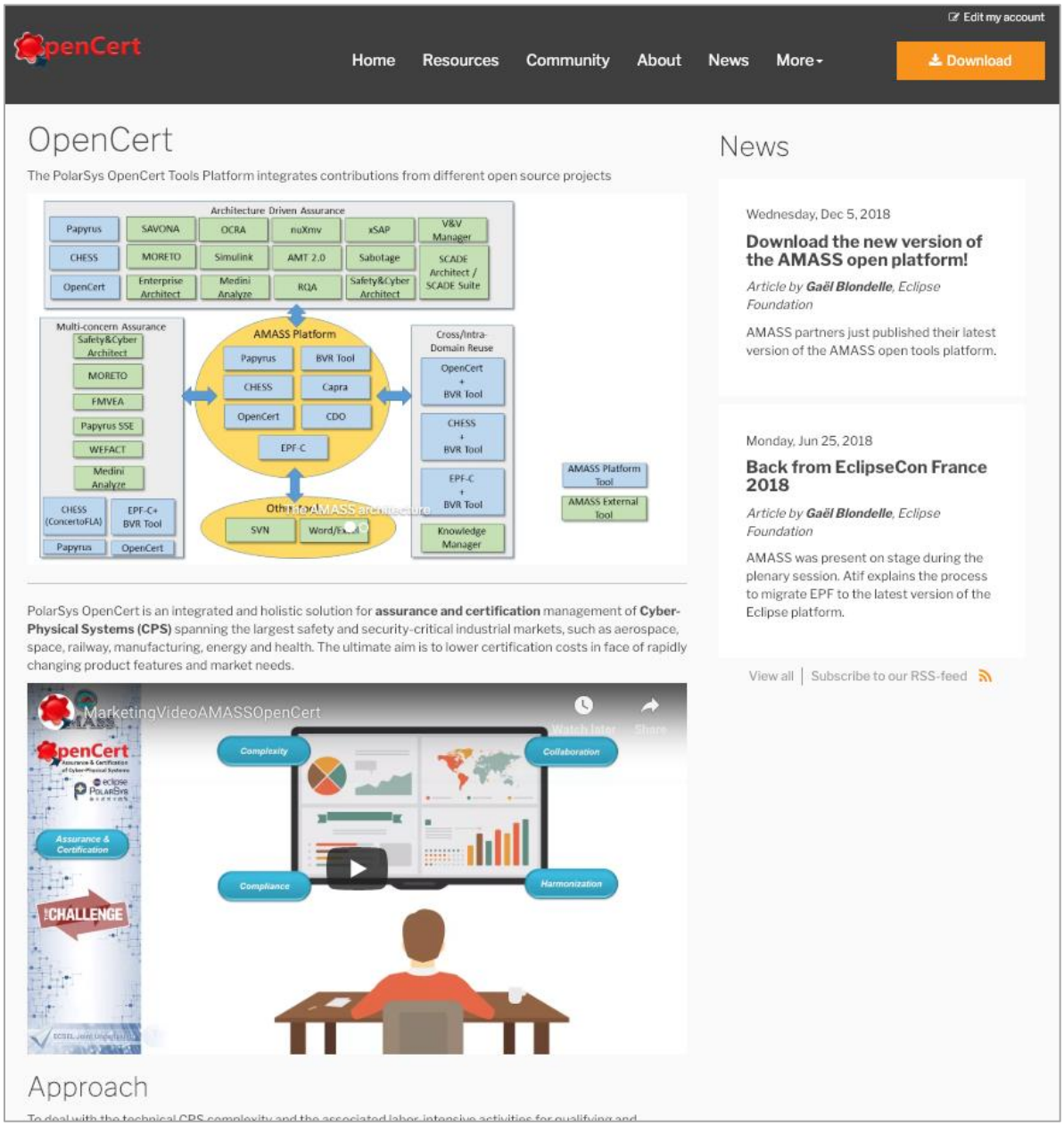


Fig. 9. OpenCert website home page.

The main benefits that can be obtained from the application of the AMASS approach include: (1) reduced initial and rework costs, thanks to guidance for compliance with standards and regulations, and for reuse of assurance and certification assets, helping engineers to more efficiently execute assurance projects; (2) better coping with risks, by deployment of safety and security analyses, and of cost-effective and transparent assurance and certification processes, improving risk management; (3) harmonized compliance, helping engineers to create a transparent view of process and product quality against a set of harmonized requirements derived from standards and regulations, and; (4) reduced compliance management and (re)certification costs, through the use of existing knowledge, quantitative methods, and modular reuse techniques.

In summary, the AMASS approach is an advanced solution for assurance and certification of critical systems, and more concretely of cyber-physical systems, that will allow different stakeholders (engineers, assessors, tool vendors, etc.) to more easily and better perform their work.

As future work, we will continue developing and applying the AMASS approach, as well as managing and maintaining the associated open source projects. At this moment we are evaluating the approach in new situations of the industrial case studies of the AMASS project. This will allow us to quantitatively assess the benefits that the approach can enable in practice. It is also expected that the approach evolves as a result of its use for different systems types (e.g. in robotics)

and of the corresponding identification of different assurance and certification needs to address.

ACKNOWLEDGMENT

The work leading to this paper has received funding from the AMASS project (H2020-ECSEL grant agreement no 692474; Spain's MINECO ref. PCIN-2015-262). We thank all the AMASS partners that have contributed to envisioning and to developing the AMASS approach, especially Huáscar Espinoza, Garazi Juez, and Cristina Martínez for the input for and feedback on the content presented in this paper.

REFERENCES

[1] AMASS Project, https://www.amass-ecsel.eu/ (accessed Jan 18, 2019)

[2] AMASS Project, "Deliverable D1.1 - Case studies description and business impact", v1.3, 2018, https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D1.1_Case-studies-description-and-business-impact_AMASS_Final.pdf (accessed Jan 18, 2019)

[3] AMASS Project, "Deliverable D2.4 - AMASS reference architecture (c)", 2018, https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D2.4_AMASS-reference-architecture-%28c%29_AMASS_Final.pdf (accessed Jan 18, 2019)

[4] AMASS Project, "Deliverable D2.5 - AMASS user guidance and methodological framework, v1.0", 2018, https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/D2.5_User-guidance-and-methodological-framework_AMASS_Final.pdf (accessed Jan 18, 2019)

[5] BVR, https://github.com/SINTEF-9012/bvr (accessed Jan 18, 2019)

[6] CHESS Project, http://www.chess-project.org/ (accessed Jan 18, 2019)

[7] CRYSTAL Project, http://www.crystal-artemis.eu/ (accessed Jan 18, 2019)

[8] M. Cusumano and A. Gawer, "The elements of platform leadership", MIT Sloan Management Review 43(3): 51-58, 2002

[9] J.L. de la Vara, M. Borg, K. Wnuk, and L. Moonen, "An Industrial Survey on Safety Evidence Change Impact Analysis Practice", IEEE Transactions on Software Engineering 42(12): 1095-1117, 2016

[10] J.L. de la Vara, A. Ruiz, K. Attwood, H. Espinoza, R.K. Panesar-Walawege, A. Lopez, I. del Rio, T. Kelly, "Model-Based Specification of Safety Compliance Needs: A Holistic Generic Metamodel", Information and Software Technology 72: 16-30, 2016

[11] Eclipse, https://www.eclipse.org/ (accessed Jan 18, 2019)

[12] Eclipse Process Framework Project, https://www.eclipse.org/epf/ (accessed Jan 18, 2019)

[13] S. Jansen, A. Finkelstein, and S. Brinkkemper, "A sense of community: A research agenda for software ecosystems", 31st International Conference on Software Engineering, ICSE 2009

[14] J.C. Knight, "Safety critical systems: challenges and directions", 24rd International Conference on Software Engineering, ICSE 2002

[15] S. Nair, J.L. de la Vara, M. Sabetzadeh, L. Briand, "An Extended Systematic Literature Review on Provision of Evidence for Safety Certification", Information and Software Technology 56(7): 689-717, 2014

[16] S. Nair, J.L. de la Vara, M. Sabetzadeh, D. Falessi, "Evidence Management for Compliance of Critical Systems with Safety Standards: A Survey on the State of Practice", Information and Software Technology 60: 1-15, 2015

[17] OpenCert: https://www.polarsys.org/opencert/ (accessed Jan 18, 2019)

[18] OPENCOSS Project: http://www.opencoss-project.eu/ (accessed Jan 18, 2019)

[19] Papyrus: https://www.eclipse.org/papyrus/ (accessed Jan 18, 2019)

[20] Papyrus IC: Product Management. https://wiki.polarsys.org/Papyrus_IC/Product_Management (accessed Jan 18, 2019)

[21] PolarSys: https://www.polarsys.org/ (accessed Jan 18, 2019)

[22] PolarSys CHESS: https://www.polarsys.org/projects/polarsys.chess (accessed Jan 18, 2019)

[23] RTCA: DO-178C: Software Considerations in Airborne Systems and Equipment Certification. 2012

[24] SafeCer Project: https://artemis-ia.eu/project/40-nsafecer.html (accessed Jan 18, 2019)

[25] The REUSE Company: Verification Studio, https://www.reusecompany.com/verification-studio (accessed Jan 18, 2019)