Assurance and Certification of Cyber-Physical Systems: The AMASS Open Source Ecosystem

Jose Luis de la Vara[1], Alejandra Ruiz[2], and Gaël Blondelle[3]

[1]University of Castilla-La Mancha, Spain
joseluis.delavara@uclm.es

[2]Tecnalia Research & Innovation, Spain
alejandra.ruiz@tecnalia.com

[3]Eclipse Foundation, Germany
gael.blondelle@eclipse-foundation.org

Abstract. Many cyber-physical systems (CPS) are subject to rigorous assurance and certification processes to provide confidence that undue risks are not posed and thus the systems are trustworthy. These processes are complex and time-consuming and tool support can greatly aid in their execution. In line with other trends for systems and software engineering, the need for and interest in open source tools for assurance and certification is growing and different initiatives have been launched. As a concrete example, we report on our experience in developing the AMASS open source ecosystem. This ecosystem includes (1) an open source tool platform that supports the main CPS assurance and certification activities, (2) external tools with added-value features, and (3) an open community of developers and users. The platform integrates existing solutions for system modelling, process engineering, and compliance and argumentation management. We also present the application of the AMASS tool platform in 11 industrial case studies from five different application domains. The results show that the platform is a feasible means for CPS assurance and certification and that practitioners find benefits in assurance-oriented system modelling and in integrated system assurance information, among other areas. Nonetheless, improvement opportunities also exist, most notably regarding tool interoperability and usability.

Keywords: AMASS, open source, ecosystem, assurance, certification, cyber-physical system.

1. 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 [51]. This kind of system is required to go through very intensive verification and validation (V&V) activities in order to assure the safety of the systems and as a final result to certify them, providing valid evidence [63]. 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 [81].

Assurance and certification of safety-critical systems require the execution of complex and labour-intensive activities [34,60,63] such as the management of compliance with numerous criteria defined in safety standards, the management of a large volume of evidence artefacts and of trace links throughout a system's lifecycle to demonstrate compliance, or the provision of convincing and valid justifications that a system is dependable. Therefore, companies developing safety-critical systems or components need tool support that facilitates these activities and ideally increases their efficiency. The challenges arising from system assurance and certification are further growing as a result of the evolution of safety-critical systems. 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, availability, robustness, and reliability, in the presence of human, environmental, and technological risks. The rise of notions such as cyber-physical systems (CPS) and their complexity are leading to the need for new approaches for system assurance and certification. In general, practitioners expect improvements in the available tool support for assurance and certification [33,64].

Another trend in CPS engineering and assurance is the use of open source tool support. Among the reasons that have contributed to this trend [20,21,22], we highlight two. Firstly, the lifecycle of a CPS can span decades in application domains such as aerospace, energy, and railway, e.g. the lifecycle of aircrafts, power plant systems, and trains. This is a characteristic of how systems are developed and used in these domains. The enterprises developing the systems and their components need to ensure that the tools and their associated support services will be available during the whole lifecycle, as this is required for system maintenance and certification. Secondly, companies in the safety-critical domains have experienced issues with vendor lock-in, lack of tool maintenance, and tool and service acquisition by third companies. Tool vendors can change the conditions under which a tool is commercialised and maintained, and changes in the market strategy can impact the tools that they develop and thus their customers.

In summary, many companies involved in CPS engineering are starting to prefer to control the development of their engineering and assurance tools to avoid possible risks from commercial tools by tool vendors. The envisioned solution has been to contribute to the development of open source solutions. Different companies can work together towards the

development of a base, common platform. They can later build on this platform to develop their own customised solutions or to provide added-value features and services. Large companies such as Airbus, Ericsson, Saab, and Thales are supporting this vision of open source tools for CPS engineering and assurance. Two clear examples are the Papyrus [73] and Capella [25] tools for model-based systems engineering. Furthermore, the need for and interest in open source solutions that support system assurance and certification is growing and different initiatives have been launched in previous years. For example, NASA, the Japan Science and Technology Agency, and several industry-academia consortia in Europe have been working on the development of open source tools for system assurance and certification [28,36,82]. Open standards have also been developed as a reference for assurance and certification activities and for assurance data exchange [66,93,97].

This paper presents the practical experience in the creation of an open source ecosystem for the AMASS tool platform, in the context of the AMASS large-scale European project [5]. AMASS stands for Architecture-driven, Multi-concern, and Seamless Assurance and Certification of CPS. In AMASS, CPS are regarded as the new generation of embedded systems with interfaces to users, the physical world, and the cyberspace. When compared to other embedded systems, a key characteristic of CPS is that they are connected by means of communication networks, such as sensors or the Internet. Examples of CPS include airplanes, cars, and trains. A CPS may consist of sub-CPS, in which case, the CPS can be regarded as a CPS of systems, e.g. a smart city.

The community of the project collaborates on the development of different methods and tools for CPS assurance and certification. The AMASS open source ecosystem corresponds to what Jansen et al. [49] 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 CPS assurance and certification and for the development of an open source tool platform. Based on its initial community, one of the intentions of the AMASS project was to create an ecosystem that is sustainable even after the project ends in order to support the evolutions of the project outcomes in the future. It must be noted that the AMASS ecosystem and the tool platform are different entities, and that the ecosystem is not a software system, but software systems are part of the ecosystem. The AMASS ecosystem also encompasses the community that supports it and the community that develops the tool platform, for example.

The creation and management of a software ecosystem poses a series of challenges [29] related to the definition of open source contribution strategies, creation of partnering models, and creation of a developer community. We report on the strategies followed and the decisions made for the AMASS open source ecosystem in order to enact the underlying ecosystem development process and to try to ensure its success, combining input and contributions from different initiatives and stakeholders. The process has been performed in the scope of the Eclipse environment [39], which imposes some requirements such as transparency, openness, and meritocracy. The process has also allowed us to learn several lessons. We share the main ones in the paper. Among the gaps identified for the development and adoption of open source solutions, it is important to define a strategy for growth and sustainability, that the solutions can be integrated with current practices and commercial tools, and that the parties involved in the development of the solution plan how and when to follow open source principles.

In addition, we present the application of the AMASS tool platform in 11 industrial case studies from aerospace, automotive, avionics, industrial automation, and railway. The utilisation of the platform features in different usage scenarios has aimed to demonstrate that the tool platform is a feasible means for CPS assurance and certification and to identify benefits and limitations that practitioners find in the platform.

The main contributions of the paper are (1) the description of a practical experience to develop a sustainable open source ecosystem for CPS assurance and certification, (2) the presentation of the main strategies followed, decisions made, and lessons learned, and (3) the demonstration of how the AMASS tool platform can support CPS assurance and certification in specific industrial situations. This is valuable for both researchers and practitioners interested in the development or use of open source solutions for the engineering and assurance of critical systems, as the challenges are shared among system lifecycle phases and critical application domains. The information is further helpful for other open source initiatives because most of the challenges faced are pervasive, e.g. the definition of a business model and of strategies for partnering.

Prior publications on the AMASS project [16] have presented aspects such as the motivation for the project [83] and the general process to apply the underlying approach [35] in more depth. Publication on specific results can also be found, e.g. on system artefact quality analysis [75]. This paper complements prior ones by presenting the details of the AMASS open source ecosystem and of the application of the AMASS tool platform in industrial case studies. More concretely, new information is provided in this paper about the management of the AMASS ecosystem community, the lessons learned, and the design, results, and discussion of the application of the platform.

The rest of the paper is organised as follows. Section 2 reviews related work. Section 3 describes the AMASS open source ecosystem, including its main features and the strategies followed for its development and management. Section 4 reports on the application of the AMASS tool platform. Finally, Section 5 summarises our conclusions.

2. Related Work

We have divided related work into two main areas: open source ecosystems and tools for system assurance and certification.

2.1 Open Source Ecosystems

The development and management of software ecosystems, in particular open source ecosystems, has been an important research area for nearly two decades. This is shown by the number of secondary studies on the topic, for example. The sub-areas studied include the business models [90], governance mechanisms and health [4], and quality assurance [18]. Franco-Bedoya et al. [45] conducted a systematic mapping focused on open source software ecosystems and concluded that the research on several topics related to this type of ecosystems is still scarce and that further investigation is needed on how organizations and open source communities actually understand open source software ecosystems. Literature reviews on software ecosystems in general [58,59] have indicated that little research is done in the context of real-world ecosystems and that further ecosystems need

to be studied. We consider that the experience reported in this paper contributes to filling some of the main gaps identified in the above secondary studies.

The specific aspects of open source ecosystems that have been studied in primary studies include sustainability [50], meritocracy [38], quality models [44], the health of the ecosystems [48], the importance of socio-technical resources [53], the forms of power [95], and the strategies to manage power [96]. An area whose interest in is growing and that is related to open source ecosystems is open innovation [62], including how to motivate contributors [54]. Foundations play a major role in open source projects and ecosystems, but they do not remove the need for developing project-specific governance, contribution, and development policies [24]. All these publications have contributed to the elaboration of theories about the development and management of open source ecosystems, thus to a better understanding of them. We contribute to the further progress of the state of the art by focusing on a specific and real case, presenting our experience with the AMASS open source ecosystem.

In the scope of systems and software engineering, Stol and Ali Babar [91] studied the challenges in using open source software in product development and found that having a community, support, and maintenance strongly impact the use in industry, as well as the integration with other components and a clear business model. For safety-critical systems, open source components have been identified in the literature [92] and their use has been analysed for defence [86] and aerospace [88]. Some publications have studied practitioners' perceptions on open source software for critical systems [55,76] and have found, on the one hand, barriers related to the lack of responsible third-party engagement, and to the complexity of open source ecosystems, among other issues; and on the other hand, advantages such as control over the software, and easy long-term maintenance. In summary, open source solutions are used for critical systems, but industry expects mature solutions, associated services such as support, and clear benefits over commercial solutions. These aspects are among the main factors that the AMASS open source ecosystem is addressing to ensure its adoption.

There exist publications that have presented the work on the development of specific open source solutions for systems and software engineering. We can find insights into initiatives and tools such as Capella [22], CHESS [61], Open-DO [84], Papyrus [21], and Polarsys [20]. This paper complements these insights by presenting new and specific information about the development of the AMASS open source ecosystem.

2.2 Tools for System Assurance and Certification

We have divided the review of tool support for assurance and certification according to the main high-level features of the AMASS tool platform (see Section 3.1): system modelling and analysis, compliance management, assurance case management, and evidence management. The main weakness of the existing tools is that most often they support only one high-level feature and are usually not integrated with tools for other CPS assurance and certification activities. For those that support several features or are integrated, the integration is only partial and only deals with pairs of activities, e.g. system modelling and argumentation [89]. Without a larger integration, it is easier for inconsistency to appear between pieces of assurance information, and that other issues such as incompleteness are not found. Some weaknesses recently identified in the assurance and certification activities

of the Boeing 737 MAX are clear examples of this situation. Information was managed with several different means and its loss between artefacts and stages was unnoticed.

System modelling and analysis is arguably the high-level feature for which tool support can be found more easily and for which more mature tools exist. Well-known tools such as IBM Rhapsody [47] and Magic Draw [65] support model-based systems engineering and are used by many companies. Open source solutions also exist, e.g. Capella [25] and Papyrus [73]. However, they are general purpose tools that are not tailored to most of the concrete needs for CPS assurance and certification. More specific modelling approaches have been proposed by researchers considering compliance with standards, either by proposing new modelling languages [72] or by extending existing ones [19]. However, these approaches are standard-specific and not applicable to other CPS assurance and certification contexts.

Compliance management is also a high-level feature that has been addressed by industrial tools by companies such as LDRA [52] and PTC [79]. These tools focus on compliance with single standards and considering single dependability concerns, and their tailoring possibilities to specific companies, CPS products, or projects are limited. This issue also exists with most compliance management tools from academia, e.g. [46]. Some researchers have proposed standard-independent tools for compliance management [43], but the tools do not consider all the types of compliance needs for CPS [34], namely the requirements from the standards to comply with, the artefacts to manage as evidence, the process to execute, the applicability of the different elements, and the configuration of a project according them.

Prior research has paid great attention to assurance case management [63]. Maksimov et al. [56,57] have performed recently detailed analyses of the tool support of this high-level feature. Most of the tools are open source and their maturity needs to be improved. The main commercial tool is arguably ASCE [1], which strongly focuses on assurance case specification and provides limited automated support. Advanced assurance case management mechanisms such as argument patterns and argument structure generation are barely supported by existing tools, as well as means for developing assurance cases that address several dependability concerns.

Evidence management is probably the high-level feature with the smallest amount of specific tool support. Some compliance management tools integrate it but the support is very limited. Companies usually integrate evidence management in general tools such as document and version control management systems, even in spreadsheets [33,64]. Tools that can be considered to partially support evidence management such as tools for quality information management [94] do not focus on specific CPS assurance and certification needs such as explicit management of compliance with several standards, the link between evidence and the arguments in assurance cases, and management of the lifecycle of evidence artefacts, considering steps such as evidence definition, information collection, evaluation, traceability specification, and change impact analysis. These aspects are essential for assurance and certification.

In synthesis, there exist tools that support the high-level features of the AMASS tool platform, but (1) the support is limited as some important needs for CPS assurance and certification are not addressed, (2) the support is not flexible enough to be applied in a sufficiently wide range of scenarios for CPS assurance and certification, and (3) most

importantly, the support is not integrated into a single environment for CPS assurance and certification. We think that these weaknesses hinder a wider application of the existing tools.

3. The AMASS Open Source Ecosystem

This section presents how we have created and developed the AMASS open source ecosystem and how we are managing it as an Eclipse project.

3.1 Features and Components of the AMASS Ecosystem

The ultimate goal of the AMASS project, and thus of the AMASS open source ecosystem, is to lower assurance and certification costs for complex CPS in face of rapidly changing features and market needs. To enable cost reduction, AMASS worked on how to increase design efficiency of complex CPS, how to increase the reuse of assurance results, how to reduce assurance and certification risks, and how to increase the harmonisation and interoperability of assurance and certification technologies. Showing directly and accurately a cost reduction on assurance and certification can be difficult in practice [32] because of e.g. the lack of precise data. Nonetheless, it can be estimated or judged based on the achievement of sub-goals. Showing a potential increase in design efficiency, increase in reuse, reduction of risks, and increase in harmonisation indirectly shows that costs could be reduced.

To the above ends, the AMASS project established 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 security and safety aspects), and for seamless interoperability between assurance and engineering activities along with third-party activities such as external assessments and supplier assurance. Aspects specific to CPS include the consideration of the new characteristics of their architectures, of the need for addressing several dependability concerns, of a wider range of interoperability aspects among tools and stakeholders, and of reuse in different situations such as those in which both safety and security are essential.

The AMASS open source ecosystem supports the main activities needed for CPS assurance and certification [9]. More concretely, the high-level features of the AMASS tool platform are:

- Assurance project management, to define the scope of compliance for an assurance project, project compliance lifecycle, reuse possibilities, and compliance means.
- Compliance needs specification, to capture, digitalise, store, and retrieve knowledge about how to comply with assurance standards.
- System modelling, mainly targeted at system requirements specification, system architecture design, and system component V&V.
- System dependability analysis, to determine the system properties and needs regarding safety as well as other dependability concerns such as security.
- Assurance case management, with which a user can justify system dependability using compliance arguments and product arguments, resolve safety-security trade-off, and link this information to system architecture.
- Evidence management, to characterise the project artefacts that are used as assurance evidence, handle artefact traceability, represent the execution of assurance processes, and specify how the artefacts contribute to compliance.

In more specific terms, the features of the AMASS tool platform have been grouped into several architectural areas (Figure 1):

- Basic building blocks, which provide the core support for system component specification, assurance case specification, evidence management, and compliance management. Its base data model is referred to as Common Assurance & Certification Metamodel. General support for access management and data management, which is not assurance and certification-specific, is also considered.
- Architecture-driven assurance, which enables system architecture modelling for assurance, architectural patterns for assurance, requirements support, contract-based assurance composition, and V&V activities.
- Multi-concern assurance, which tackles system dependability co-analysis and co-assessment, dependability assurance, and contract-based multi-concern assurance.
- Seamless Interoperability, for tool integration management, collaborative work management, and tool quality assessment and characterisation.
- Cross- and intra-domain reuse, which exploits reuse assistant, impact analysis, automatic generation of process- and product-based arguments, semantic standard equivalence mapping, and process, product, and assurance case reuse via management of variability.

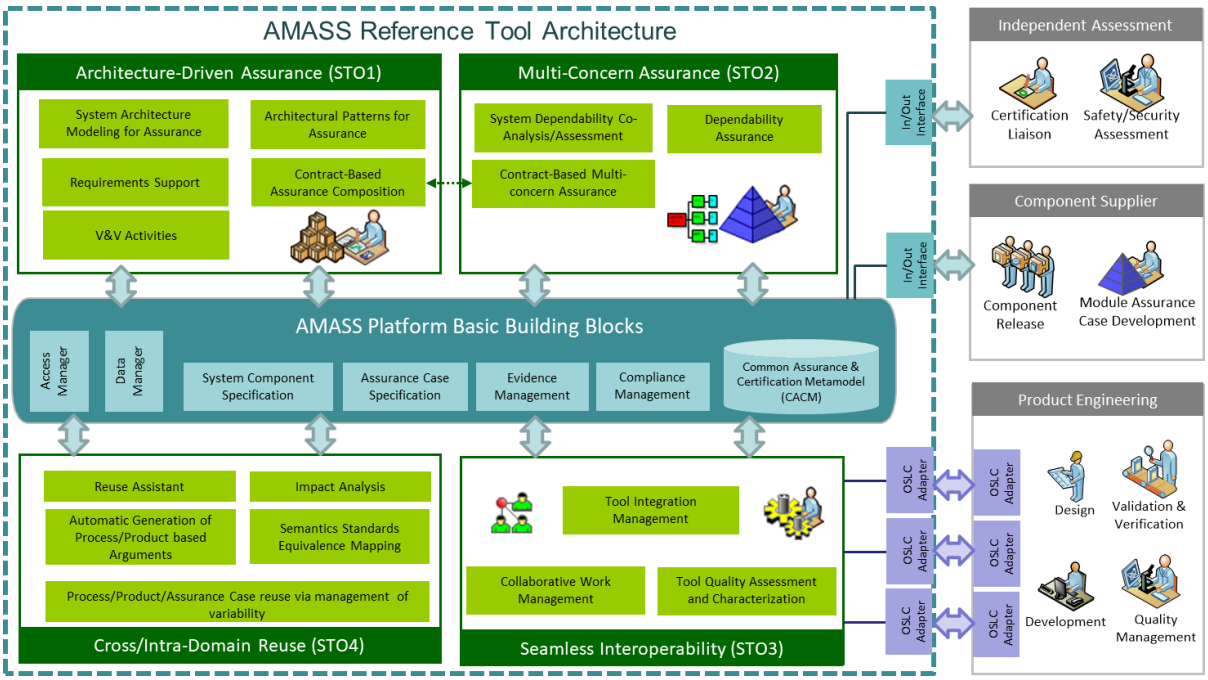

Figure 1. AMASS reference tool architecture

These features have been selected according to: (1) the input from prior projects whose results and insights have been used as basis, e.g. solutions for modular assurance cases [70]; (2) the decisions made by the AMASS consortium upon how to best enable a new approach for CPS assurance and certification, and; (3) the feedback received from other parties such as the project's advisory board and practitioners that have attended presentations on the AMASS tool platform. To a large extent, the AMASS project aimed to fill gaps in the current means for assurance and certification that prior projects did not, and to build on their results.

More details about the blocks can be found in AMASS deliverables [8]. The blocks allow different stakeholders to perform tasks for CPS assurance and certification: Managers, such as Project, Assurance, and IT Managers; Engineers, such as Development Engineers (including Process Engineers) and Assurance Engineers (including Safety and Security Engineers), and; Assessors, such as Assurance Assessors (including Independent and Internal Assessors). The blocks consider both CPS-developer-internal aspects, e.g. for product engineering, and external ones, e.g. regarding component suppliers and independent assessors. Therefore, the AMASS tool platform offers interfaces to a wide range of possible users and help them in contributing to CPS assurance and certification. The usage of the platform can be customised, both in terms of the selection of the features to exploit and in terms of how to enact a CPS assurance and engineering approach for a project, e.g. according to the applicable elements of a standards.

The stakeholders above are also among the main ones that have been involved in the development of the AMASS tool platform, either providing input for its design or specifying, implementing, and validating the platform. Nonetheless, others have largely contributed as well, most notably those leading the effort to build the AMASS open source ecosystem in the scope of the Eclipse environment. Staff at the Eclipse Foundation, as one of the partners of the AMASS project, have played a major role.

The AMASS tool platform can also deal with risk-based strategies, not only with compliance-based assurance. Risk goals can be explicitly shown in assurance cases and traced to specific elements of the architecture. The rationale about how the risks are managed can be described in the assurance cases and connections to the evidence that supports the corresponding claims can be established. If needed, such evidence can be collected from external tools connected to the platform, which can also support specific risk-oriented analyses such as detailed fault tree analysis.

Since its inception, one of the goals of the AMASS project was to create a large ecosystem by merging open results from prior research projects such as OPENCOSS [70], SafeCer [87], CHESS [30], and CRYSTAL [31]. Joining these communities was the first step to gaining the necessary visibility for the ecosystem and a good start to bootstrap an open source tool platform. In order to enable the creation of a larger ecosystem, the AMASS project partners created the AMASS tool platform, an open source package of several Eclipse open source projects that can be used as a joint platform for new products and services. The platform includes:

- OpenCert [67] (Figure 2) for assurance- and certification-specific activities such as assurance case specification, evidence management, and compliance management.
- The CHESS toolset [78] (Figure 3), which is based on the Papyrus tool [73], for model-driven, component-based, and contract-based development of high-integrity systems.
- EPF Composer [42] (Figure 4) for systems and software process engineering.

The screenshots in Figures 2, 3, and 4 show the kind of user interface of these tools and their different elements, e.g. a canvas for graphical modelling and validation information.

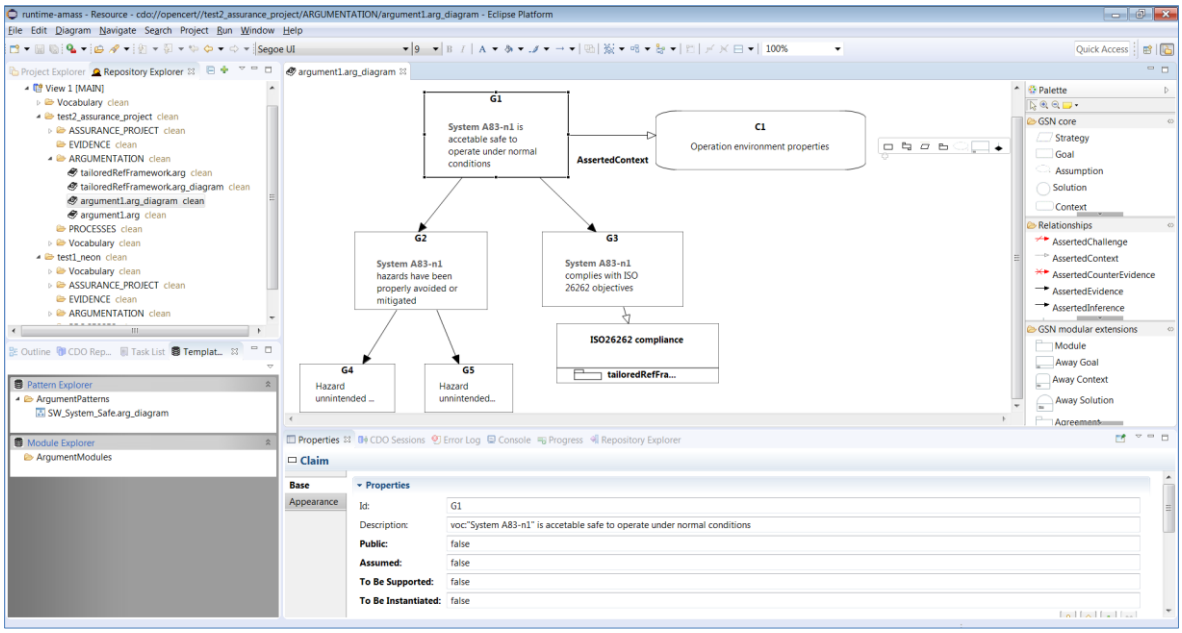

Figure 2. OpenCert screenshot for assurance case specification

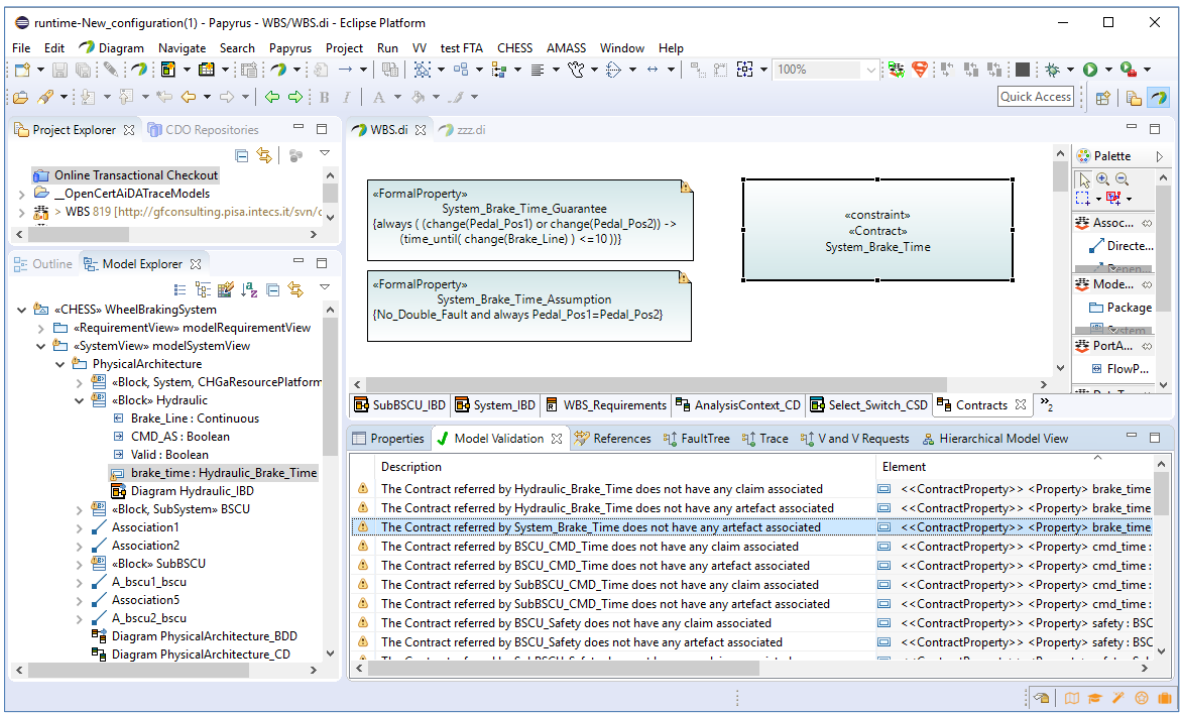

Figure 3. CHESS screenshot for contract specification

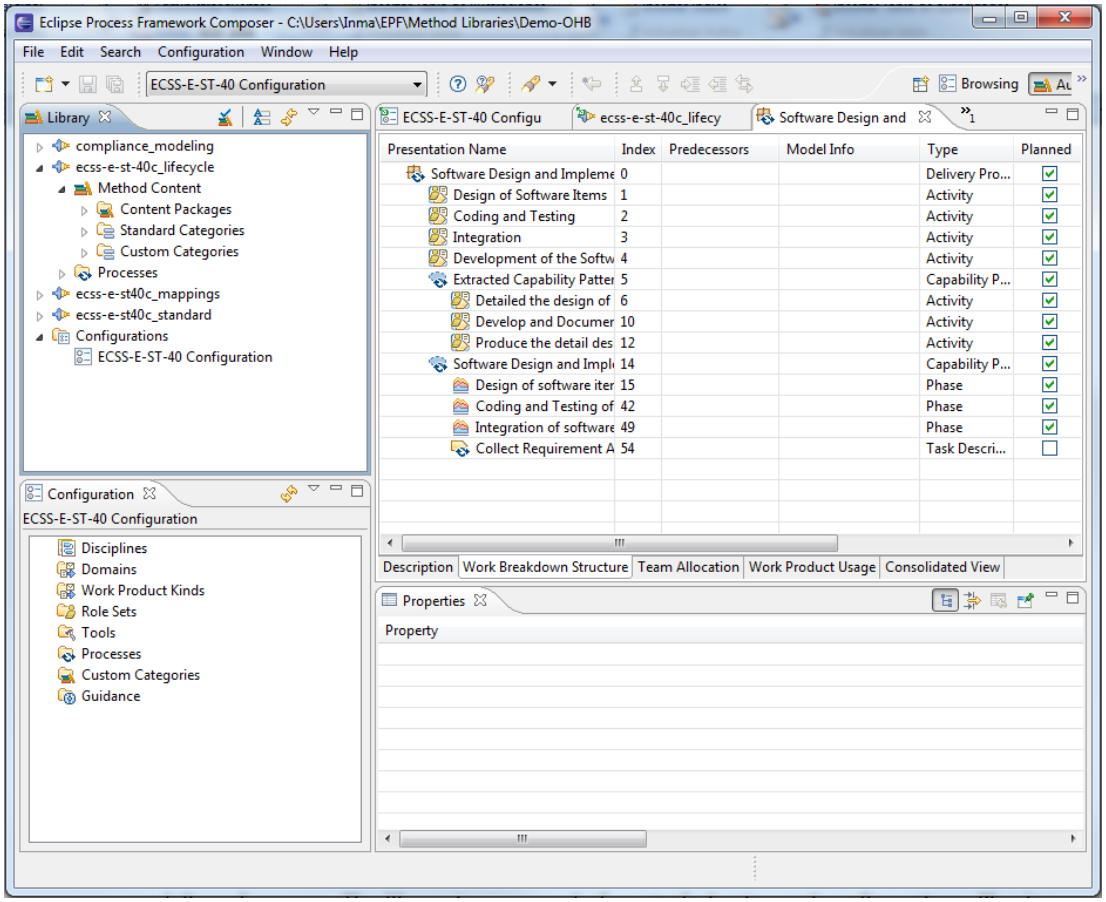

Figure 4. EPF Composer screenshot for process specification

Further tools and technologies have been used for the development of the AMASS tool platform, most notably BVR [23] for variability management, Capra [26] for traceability, OSLC [71] for tool interoperability, and CDO [27] for data storage.

The open source projects are hosted by Eclipse with different lifecycles. The AMASS tool platform bundles them into a package that provides an integrated and extensible solution for CPS assurance and certification. The first step was to integrate all solutions and to make them work together seamlessly for the user. Then new features were developed, e.g. for integration with external tools.

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

- The AMASS tool platform, which provides open-source base functionality for compliance and argumentation management, system dependability analysis, and system modelling, based on the integration of the above tools.
- Other tools that provide additional features either as Eclipse plug-ins, e.g. for system dependability analysis with Papyrus [74], or as external tools that exchange data with the platform, e.g. the RQA - Quality Studio tool [94] for analysis of system artefact quality. Currently there are over 20 tools for this part [9]. These tools are typically commercial ones.

The integration with external tools can be addressed in two different ways: as an ad-hoc, tool-specific solution that is tailored to the data format and services of a tool, or as a generic, tool-independent solution using the OSLC-KM approach [3], which provides a base common

format and infrastructure. Both approaches have been successfully applied and have advantages and disadvantages. The former is typically better for very tool-specific aspects, e.g. to take advantage of concrete services of a tool such as Rhapsody. Whereas the latter facilitates cross-tool integration between similar technologies, thus reducing the effort in connector development. Ad-hoc tool integration usually poses more challenges related to having to deal with the specific interoperability means, which might be difficult to use or have limitations. A usual example of this kind of issues is the exploitation of the DXL language for integration with DOORS.

Although the AMASS tool platform offers a large and comprehensive set of features for CPS assurance and certification, its functionality could be more extensive and thus can be complemented with other tools for specialised aspects. In addition, it is possible that a company is already using another tool with a similar purpose, e.g. for system modelling, and wants to keep its use because the engineers are experienced with the tool, for example. In this case the usage of the AMASS tool platform could be tailored.

More information about the implementation of the tools of the AMASS ecosystems can be found in the deliverables of the AMASS project [11,12,13,14].

The stakeholders that are part of the AMASS ecosystem include:

- System manufacturers, e.g. Alstom for railway, Schneider Electric for energy, and Thales for avionics.
- Component suppliers, e.g. GMV for space, Honeywell for aerospace, and Infineon for automotive.
- Certification organisations and assessors, e.g. Alten for aerospace, Intecs for automotive, and RINA for railway.
- Tool vendors, e.g. ANSYS medini for automotive, Rapita Systems for avionics, and The REUSE Company for aerospace.
- Research and technology organizations, e.g. AIT, FBK, RISE, and Tecnalia.
- Universities, e.g. Carlos III University of Madrid, Mälardalen University, and Masaryk University.

In total, over 50 organisations are directly or indirectly involved in the AMASS open source ecosystem, not only organisations from the AMASS project (partners and organisations in the advisory board) but also from its antecessor projects. The AMASS tool platform is further being used in other projects, e.g. AQUAS [17] and RobMoSys [80], thus the set of involved parties is growing.

During the presentation of the AMASS tool platform to different audiences, people external to the AMASS project were able to learn about the platform and to use it. Making the platform open into the Eclipse framework has also made different stakeholders confident that there is a community behind it, that the community will respond to specific tool support needs, and that there are companies that can be hired if specific developments and adaptations are needed. It has been usual to receive direct emails from users when trying to use the AMASS tool platform for the first time and they have received response. The public resources have also been updated according to their feedback to provide answers to future users. In most of the cases, less than two email exchanges have been needed to provide answers to the specific questions posed. Some AMASS partners are already providing specific services on the AMASS tool platform.

Regarding standardisation, this was a key activity of the AMASS project. Most of the project partners were involved in standardisation efforts, such as about new system and assurance modelling approaches or about co-engineering processes for safety and security. This involvement has allowed the partners to contribute to the update of standards or the development of new ones according to the insights gained during the project. The specific contributions have been reported in AMASS deliverables [15].

3.2 Management of the AMASS Ecosystem Community

The overall management of the AMASS open source ecosystems can be divided into three main areas: the integration of the ecosystem in Eclipse, the governance and sub-projects, and the advantages of the principles followed. Although the aspects presented are linked to the AMASS ecosystem, we strongly believe that they can be applicable to other situations, such as the development of other open source platforms and communities. Our prior experience in such situations with other Eclipse projects support this, as we also had to deal with e.g. how to follow Eclipse principles in an adequate way.

3.2.1 Integration in Eclipse

The AMASS open source ecosystem relies on the values, processes, and services of the Eclipse Foundation. Many challenges are automatically tackled in pursuing compliance with the Eclipse Foundation's functioning model. Competitive and cooperative endeavours are strategically balanced to ensure that the resulting synergy contributes to making the platform take the necessary steps towards maturity.

The Eclipse Foundation was created in 2004 as an independent not-for-profit organization to act as the steward of the growing community around the Eclipse Integrated Development Environment. The Foundation fosters a large business ecosystem with more than 300 open source projects in various domains. The technology and source code developed by the Eclipse community is made available royalty-free under the Eclipse Public License (EPL). As a "business friendly" license, the EPL simultaneously fosters cooperation on a platform and competition on products. Figure 5 depicts how this is done, building products on top of a platform. On one hand, developers who modify code licensed under EPL must publish their code if they redistribute it; on the other hand, the binaries can be integrated into any kind of product whether it is open or proprietary. Eclipse is also a platform that facilitates the development of rich modelling tools in open source. The ecosystem represents a "mille-feuille" of dozens of complementary projects and has enabled tool vendors to create hundreds of open and proprietary modelling products.

Over the last decade, the Eclipse ecosystem has proven to be a good place to foster collaboration in open source. This is rooted not only in the characteristics of the EPL but also in the core values of the Eclipse ecosystem and the services and processes offered by the Foundation. These values, services, and processes are used in the AMASS open source ecosystem. The core values are complementary and not negotiable. They are the DNA of the Eclipse ecosystem, their application ensures vendor neutrality, and they support the partnering model of Eclipse.

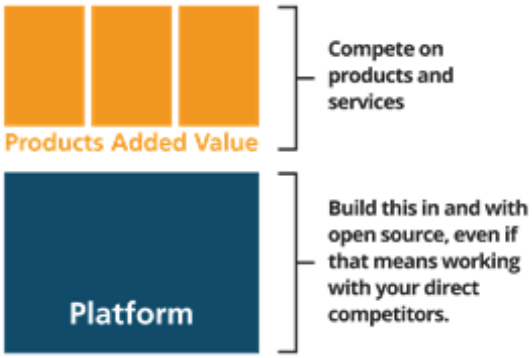

Figure 5. A business-friendly ecosystem based on extensible platforms

Through “transparency”, all the decisions in a project must be public, including the conversations leading to these decisions that provide the context. Outsiders can follow what is happening and eventually step into the discussion. For example, anyone can access the forum of the AMASS tool platform to check what has been and is being discussed. “Openness” means that participation in an Eclipse project is open to every individual without restriction in terms of affiliation, gender, etc. It explicitly fosters collaboration between organizations that otherwise compete on the market. Anyone interested can join the community of the AMASS ecosystem and contribute to the development of the tool platform. “Meritocracy” dictates that new committers, the individuals with write access to the source code, are elected by their peers, the project committers, only after the new committers publicly demonstrate their ability to contribute to the project. Being hired by a company participating in a project does not grant the right to become a committer. Staff at partners of the AMASS consortium had to show their suitability before being allowed to commit code for the AMASS tool platform.

It should be noted that the values of project development in Eclipse might not match the practices used in assurance and certification processes. For example, these processes usually are not transparent. The values for the development of the AMASS tool platform do not need to be same as the values of the processes that use the platform.

The ecosystem also needs to be supported by infrastructure, services and processes. The Eclipse Foundation manages the IT infrastructure for Eclipse working groups, including code repositories, bug trackers, continuous integration, mailing lists, and websites. The Eclipse community also shares best practices through the Eclipse Development Process [41], adapted for large-scale distributed development that involves different organizations, and the Eclipse IP Process [40] that manages the Intellectual Property of all the code published at Eclipse, including license compatibilities.

In 2012, a group of major industry players, including Airbus, Ericsson, and Thales, decided to create PolarSys [77], an Eclipse working group on open source solutions for model-based systems engineering. The ground rules of open source in PolarSys are a guarantee that users can get together to co-fund specific features and support the maintenance of the tools as long as they are needed for systems that can be in use for potentially 30 to 40 years. Among its goals, Polarsys aimed to provide means of collaboration between end user companies, organize sustainable commercial services and ecosystems around open source components, foster exchanges between academics and industry partners, and manage the quality and maturity of tools and components from early research prototypes through to

obsolescence. The AMASS open source ecosystem was initially created in the scope of Polarsys but its management has been transferred to the Eclipse Foundation.

As a strategic decision, the AMASS open source ecosystem is based on the above environments so that (1) it enables the development of both collaborative and competitive features, (2) it takes advantage of model-based technology for platform development, and (3) it builds on the experience and competence at Eclipse and PolarSys and on their commitment towards solution sustainability. In addition, the AMASS community had to work with two overall constraints. On one hand, the community had to manage the development within a research project with its predefined deadlines. On the other hand, the community had to put the first steps to create a global and open community which had to survive the end of the research project.

3.2.2 Sub-Projects and Overall Governance

As mentioned above, the AMASS tool platform was initially formed by the joint effort of three open source projects. As of October 2019, three versions of the AMASS tool platform have been released. A new version is under preparation.

OpenCert is the core of the AMASS tool platform and hosts the released integrated bundle of the platform [68]. OpenCert was created by the members of the OPENCOSS project. At the end of the project, a core OPENCOSS project partner (Tecnalia, in its role of OPENCOSS coordinator) started to work on the creation of an open source project for the project results and in November 2015 submitted the open source licensable results as a PolarSys project. After the recruitment of two mentors, the approval of the name after an analysis of potential trademarks issues, and a period of community review, the proposal was approved in December 2015. TECNALIA started the initial contribution for OpenCert in March 2016. After one month, with the help of the project mentors, the Eclipse IP team authorized the project committers to start working, according to the parallel IP process. OpenCert was considered as an incubation project at the beginning of the AMASS project. It later evolved to become a consolidated project, increasing the number of committers.

The **CHESS** toolset was created by the CHESS project and continued by SafeCer. This toolset leverages another important Eclipse project, the Papyrus platform for UML design and profiles. Intecs posted the CHESS project proposal in October 2013 and played a similar role for CHESS as the one that TECNALIA played for OpenCert. The project was accepted in February 2014 and the development activities started in the following months. As CHESS started earlier than OpenCert, CHESS has been able to go through the full IP due diligence process during the whole AMASS project, and several releases have been made. The release naming (0.9, 0.10...) indicates that the project team keeps the right to change the APIs before releasing a 1.0 version. The convention is that every time a project team breaks external APIs, they have to increase the major version number.

The Eclipse Process Framework, which manages **EPF Composer**, is one of the historic projects hosted by the Eclipse Foundation, created by IBM several years ago. As it is a very mature project, there is not so much activity on the project. One of the main actions done in order to include this project within the AMASS tool platform was the migration of EPF to a more recent version of Eclipse so that the project could be seamlessly integrated in the tool chain.

Regarding the **relation with external parties**, as any open source project, the AMASS ecosystem is open to new stakeholders. By design, the Eclipse license and the Eclipse community allow newcomers to become active in the project and to implement such changes. Open source projects are meritocratic. In order to become a committer, a new contributor must be elected by the existing committers. This election must take into account previous contributions that demonstrate the capability of the contributor to understand the project architecture and to contribute useful code to the project. Since the beginning of the AMASS project, over 10 new developers have been granted rights to commit to the public code repositories through the processes of meritocracy. The experience with Eclipse and development in the open were new for the developers or their organizations in some cases. This impacted the way in which the companies see the open source and its business.

New users are highly welcome and there are different means to learn about the platform use, such as user manuals [9] or training activities that are recorded and available in a dedicated YouTube channel [69]. Tool vendors that are interested in integrating any of the results of the AMASS tool platform can do so by following the Eclipse licence. This has already been done by AMASS project partners, e.g. The REUSE Company for the RQA - Quality Studio tool [94] and AIT for the WEFACT tool [2]. Inquiries about the tool platform have also taken place from organizations outside the AMASS project that have learned about the ecosystem through community activities.

3.2.3 Advantages of the Principles Followed

The major advantages resulting from the application of the principles and structure of the AMASS open source ecosystem are as follows.

No vendor lock-in. The AMASS tool platform does not depend on the status of a single vendor that has built it. The ecosystem can welcome small or large organizations that use the platform as is or extend it, as well as vendors that use it as a basis for their products.

Mutualisation. The community gathered around the AMASS tool platform brings a much larger capacity to maintain the platform assets. This capacity is also strengthened with each new user or extender.

Sustainability. The AMASS tool platform is resilient to the departure of authors and to the end of the AMASS European project. The source code will remain accessible and the community can continue platform development.

Standardization. AMASS is developing a standard for the management of information about CPS assurance and certification that is available in the platform for reuse by everybody. Users of the standard can access not only the specification but also an open source reference implementation.

Interoperability. Tools based on the AMASS tool platform will share the code and will be able to easily exchange data.

Adoption and innovation. Users can begin using the platform without asking for permission - before they decide upon adoption. The platform is suitable for new users who want to better understand the technology, as well as vendors who want to provide their own innovations on top of the platform.

3.3 Lessons Learned

This section summarises and synthesises the main lessons that we have learned as a result of the development of the AMASS open source ecosystem. The description of the lessons includes recommendations on how to proceed in similar situations. Both the lessons and the recommendations are useful for other researchers and practitioners when dealing with other open source projects and show the aspects that need to be considered, and how they can be addressed. We have aimed to formulate and present the lessons learned in a way that can be useful for people and projects involved in similar endeavours.

Integration of existing solutions is a good option for an open source ecosystem and it is feasible.

Contributing to pre-existing open source components is not always easy but it is practically always possible. The advantages of this approach, instead of developing new components, include the integration with an existing community, thus sharing component management and maintenance, as well as reducing development effort and cost.

The most representative example of this approach in the AMASS open source ecosystem is the usage of EPF Composer. The AMASS partners had to step in and migrate the tool from a 11 years old version of the Eclipse framework to a more recent one. The developers had to get in touch with the EPF team, submitted a plan for migration, started to collaborate with the team, and finally could fix migration issues and submit a patch. The EPF team then integrated the patch and did additional testing before creating a new release of EPF Composer, and before inviting one of the contributors to officially join as a committer of EPF. This also demonstrates in practice the openness and the meritocracy, and the sustainability of the open source projects.

Challenges must nonetheless be taken into account. In addition to the one about the versions of the Eclipse framework mentioned above, for the AMASS tool platform we had to deal with the integration of the different metamodels underlying the tools and of the different storage technologies that they used (files vs. a database). The integration was possible but not always straightforward. For integration with the AMASS tool platform we have also implemented an approach based on the OSLC standard. Implementations of this integration mechanism are publicly available for Java and .Net technologies.

A strategy for growth and sustainability is essential.

When envisioning the creation of the AMASS tool platform, one of the first questions was how to guarantee that the platform (1) would grow as needed (i.e. that new features would be developed and could be effectively added) and (2) would exist and be maintained once the AMASS project finished.

First, we needed to foster an active and productive collaboration among AMASS partners on the development of the platform. Next, the platform needed to be easily accessible and extensible to attract adopters. Finally, the immediate growth of the ecosystem depends on both the community of Eclipse and PolarSys members that share the same center of interest, and tool vendors who build products on top of the platform for vertical markets.

Engaging software development companies in the implementation of the AMASS tool platform and in the ecosystem has been essential for growth and sustainability. The companies are building their own solutions on top of the platform and integrating the platform with their tools, providing added-value products and services for CPS assurance and certification.

Open source solutions for systems and software engineering need to be integrated with current practices and commercial tools for success.

Tools that support systems and software engineering at a company are not isolated, but are part of toolchains consisting of different tools for different purposes: requirements specification, system design, simulation, testing, etc. The different tools work together towards the common objective of engineering a product. This includes open source tools and the tools for system assurance and certification. Therefore, for effective adoption, these tools must be integrated with the current practices and commercial tools at a company.

The above need has been a premise for the AMASS open source ecosystem since its conception, as a precondition for success. On the one hand, the AMASS tool platform will have to interact with other tools to exchange data. These tools can perform additional functions for e.g. V&V, and can manage data that must be managed in an assurance project, e.g. evidence artefacts in the form of textual requirements. On the other hand, it is possible that a company is already using some tool that is equivalent to a part of the AMASS tool platform. The most common case that we have found is that a company is already using some commercial tool for system modelling such as Rhapsody. In these situations, it must be possible for a company to continue using the available tools and to only select those features of the AMASS tool platform that complement or enhance its current practices.

To ensure effective and easy integration, providing means that enable it is a must for a tool such as the AMASS tool platform. We have been working both on the provision of generic approaches for tool integration that could be tailored to different tools, based on the OSLC KM approach [3], and on the development of ad-hoc connectors to further exploit certain services and capabilities of external tools.

Transfer and early adoption must be considered in advance.

One of the main issues when developing a new open source ecosystem is to guarantee that it will be used in practice. Usually this is not a major challenge when the companies that correspond to tool users are involved in the development, but the situation is different when a group of developers is working on an ecosystem for third parties, as it has most often happened with the AMASS open source ecosystem.

Means to promote transfer and early adoption are necessary, including the demonstration of the effectiveness and of the benefits of the ecosystem. Several means have been used for the AMASS open source ecosystem. Firstly, we have had the advantage that the ecosystem has been developed in the scope of a research project between industry and academia, including potential users of the AMASS tool platform. This has allowed us to use and deploy the platform in 11 industrial case studies from aerospace, automotive, avionics, industrial automation, and railway (see Section 4). The companies providing the case studies are the first possible early adopters, and the results from the case studies allow us to provide evidence of how the platform can be used in practice and of the advantages that it can

enable. Secondly, great effort has been and is being spent in the dissemination of the ecosystem at different industry-targeted events to increase the awareness about it. In this case, we are not only focusing on user companies as a target, but also on open source communities that could be interested in contributing to the AMASS tool platform.

It is also important that adoption considerations take tool qualification aspects into account. These aspects, which are about the fact that the quality and suitability of the outcomes of an engineering or assurance tool must be guaranteed, have been regularly referred to by industrial stakeholders throughout the AMASS project. We have addressed them by explaining how the documented engineering activities for the platform (requirements specification, architecture design, testing results…) contribute to the tool qualification requirements and the associated work-products required in different application domains [10]. For example, 151 high-level requirements and 73 use cases were specified for the AMASS tool platform. The execution of 141 test cases confirmed the valid implementation of 93% of the high-level requirements. Some of these requirements were finally not implemented or only partially. Tool qualification levels for the different sub-tools of the AMASS tool platform according to several standards have been initially established. For example, the level for CHESS is higher than for OpenCert because CHESS performs certain verification actions.

Following all the principles of an open source approach can take some time, thus it needs some planning.

One of the main challenges that the AMASS ecosystem has faced is to fully work in a transparent and open source manner, including the management of the intellectual property. Even in a research project, where open collaboration between partners is the rule, moving from a private environment to a fully public one is not easy, especially with the requirements to also enable the development of proprietary extensions to the AMASS tool platform. For example, only parts of the communication related to the platform have moved to the public mailing lists since the very beginning of the AMASS project, mainly for the contributions to CHESS and EPF Composer.

Being conformant to the intellectual property process is also a big hurdle. Extenders need to be sure that they are allowed to incorporate open source software into their products. The two key steps are that each source file must have a correct copyright and license header, and that each library referenced must use a license compatible with the rest of the AMASS tool platform. This kind of processes is usually implemented by software vendors or integrators, instead of by researchers. Doing it early facilitates technology transfer.

Finally, for companies and developers that do not have experience in contributing to open source projects, the availability of documented guidance is very important. This has been addressed by AMASS partners [9]. Otherwise, they can become frustrated because of not knowing how to effectively contribute or can end up not contributing in an adequate way. This can lead to re-work and loss of motivation.

4. Application of the AMASS Tool Platform

This section presents the work conducted for the application of the AMASS tool platform in industrial case studies. Case study research aims to investigate contemporary phenomena

within their real-life context, especially when the boundary between the phenomena and the context cannot be clearly specified [85]. Case study research is typically exploratory and flexible, and uses qualitative data as primary source.

The goal of the industrial case studies was to evaluate the effectiveness of the use of the AMASS tool platform for CPS assurance and certification. Three research questions were formulated:

- RQ1: Is the AMASS tool platform a feasible means for CPS assurance and certification?
- RQ2: What benefits do practitioners find in the usage of the AMASS tool platform?
- RQ3: What improvement opportunities do practitioners identify in the AMASS tool platform?

The next sections present the design, the results, and a discussion of the application of the AMASS tool platform.

4.1 Design

The design used to answer the research questions consisted in three elements: (1) selection of relevant and representative industrial case studies on CPS assurance and certification and of concrete usage scenarios; (2) utilisation of the features of the AMASS tool platform in the industrial case studies and usage scenarios; (3) collection of feedback on feature utilisation regarding the benefits and improvement opportunities found.

The industrial case studies used for the application of the AMASS tool platform and the usage scenarios enacted are as follows:

- ICS1 - Industrial and automation control
  Owner: Schneider Electric (ES)
  - US1-ICS1 - Management of compliance with IEC 61508 and IEC 62443
  - US2-ICS1 - Safety and security co-assessment
- ICS2 - Advanced driver assistance function with electric vehicle sub-system
  Owner: Infineon (DE)
  - US1-ICS2 - Reuse of safety artefacts within a product family
- ICS3 - Collaborative automated fleet of vehicles
  Owner: Assystem (DE)
  - US1-ICS3 - Safety assessment for collaborative automated vehicle functions by model-based safety analysis and contracts
  - US2-ICS3 - Process for development of collaborative automated vehicle functions, which considers functional safety, cybersecurity and reuse aspects
  - US3-ICS3 - DC drive (powertrain) validation
- ICS4 - Design and safety assessment of on-board software applications in space systems
  Owner: GMV Aerospace and Defence (ES)
  - US1-ICS4 - Architectural design of on-board software
- ICS5 - Railway platform screen-doors controller
  Owner: CLEARSY (FR)

    - US1-ICS5 - Generation of Frama-C asserted C code from B models
    - US2-ICS5 - Support for system-level modelling, including safety and security aspects

- ICS6 - Automatic train control formal verification
  Owner: Alstom Transport (FR)
    - US1-ICS6 - Assurance project management
    - US2-ICS6 - System design, V&V, and dependability assessment
    - US3-ICS6 - Evidence Management
    - US4-ICS6 - Management of compliance with EN 50128 and EN 50129

- ICS7 - Safety assessment of multi-modal interactions in cockpits
  Owner: Honeywell (CZ)
    - US1-ICS7 - Application of aerospace industrial standards for safety assessments
    - US2-ICS7 - Automation of verification objectives
    - US3-ICS7 - Reuse of assurance artefacts from automotive technology into the avionics domains

- ICS8 - Automotive telematics function
  Owner: RISE (SE)
    - US1-ICS8 - Multi-concern assurance case for safety and security
    - US2-ICS8 - Multi-concern assessment
    - US3-ICS8 - Multi-concern specification, analysis, and assurance

- ICS9 - Safety-critical software lifecycle of a monitoring system for navigational aid
  Owner: Thales (IT)
    - US1-ICS9 - System and software design and safety analysis
    - US2-ICS9 - Assurance case development

- ICS10 - Certification basis to boost the usage of multiprocessor system-on-chip architectures in the space market
  Owner: Thales Alenia Space (ES)
    - US1-ICS10 - System modelling and Reconfigurable FPGA architectures

- ICS11 - Design and efficiency assessment of model-based attitude and orbit control software
  Owner: OHB (SE)
    - US1-ICS11 - Compliance management and generation of process-based arguments
    - US2-ICS11 - Reuse via variability management

Some industrial case studies correspond to past projects at the owners, whereas others correspond to new situations in the companies. More information about the industrial case studies and their usage scenarios can be found in AMASS deliverables [6,7].

Based on the characteristics and needs of the usage scenarios and of the functionality provided by the AMASS tool platform, those involved in the industrial case studies selected the features to use. The users mostly corresponded to practitioners (engineers, system assurance managers, assessors, and certifiers) that did not contribute to the implementation of the platform, including case study owners. Nonetheless, researchers and tool support

providers were also involved, e.g. for training and feature demonstration purposes. Training sessions were arranged [69] and methodological guidance and a user manual were prepared [9] so that the users had sufficient and consistent knowledge about the platform. The users could also ask the developers at any moment about the features of the AMASS tool platforms and how to use them. All the participants were experienced engineers and managers that had not worked with the new features developed for the AMASS tool platform but could have some knowledge about the basic building blocks or the underlying techniques. For example, they could have already dealt with model-based systems engineering or assurance case development, but in a different way to how the new features of the AMASS tool platform enable them.

After utilising the AMASS tool platform in the usage scenarios, the practitioners involved were asked to fill a table in which they had to specify the main concern areas of the corresponding industrial case study, the main benefits found, and the main improvement opportunities identified. Researchers did not participate in this step.

4.2 Results

Table 1 shows the results regarding the features of the AMASS tool platform used in each industrial case studies. This information is the basis to answer RQ1 (feasibility of CPS assurance and certification). The features that were used in a highest number of industrial case studies are System Component Specification and System Architecture Modelling for Assurance (10 industrial case studies; 91%), followed by Assurance Case Specification, Evidence Management, Compliance Management, Requirements Support, and V&V activities (eight industrial case studies; 73%). The least frequently used features are Impact Analysis and Tool Quality Assessment and Characterisation (one industrial case study; 9%). The industrial case study with the widest feature usage was ICS3 (19 features out of 22; 86%), and the case study with the narrowest feature usage was ICS2 (3 features; 14%).

Table 2 shows the data related to RQ2 (benefits) for each industrial case study. The data provided by the practitioners has been synthesised and generalised for homogeneity among the industrial case studies. This process was led by the first author, following an open-coding approach and employing several iterations. The second author validated the outcome, comparing the results with the raw data. Possible divergences were discussed and agreements were reached when needed. In total, the practitioners referred to 31 individual benefits 69 times. The benefit most frequently reported was Assurance-oriented system modelling (6 industrial case studies; 55%), followed by Integrated system assurance information and Evidence information generation (5 industrial case studies; 45%). Out of 31 benefits, only one has been reported in most of the industrial case studies. Almost half of the benefits (13) have been identified in only one industrial case study. The industrial case study with the highest number of found benefits was ICS10 (10 benefits; 32% of the total number of benefits), and the case studies with the lowest number were ICS2, ICS5, and CIS11 (four; 10%). The average number of found benefits is 6.3 and the median is 6.

Table 3 includes the improvement opportunities (RQ3) identified by the practitioners involved in the industrial case studies. The data provided by the practitioners has been synthesised and generalised for homogeneity among the industrial case studies, in the same way as for Table 2. In total, the practitioners referred to 22 individual improvement opportunities 51

times. Among them, Enhanced usability and user interface and Further tool interoperability possibilities were the improvement opportunities referred to in the highest number of industrial case studies (seven industrial case studies; 64%), followed by Advanced system modelling features (5 industrial case studies; 45%) and by Better tool performance and Support for workflow configuration (4 industrial case studies; 36%). The industrial case studies with the highest number of identified improvement opportunities were ICS6, ICS7, and ICS10 (seven improvement opportunities; 32% of the total number of improvement opportunities), and the case study with the lowest number was ICS5 (two; 10%). The average number of improvement opportunities is 4.8 and the median is 5.

AMASS deliverables present more details about the results from applying the AMASS tool platform [7], such as the concrete tools used among those of which the AMASS ecosystem consists and the concrete responses regarding the benefits and improvement opportunities. As examples, we are summarising in the following subsections the results from the application of the platform in three usage scenarios from three different application domains.

Table 1. AMASS feature coverage

	Industrial Case Study										
AMASS feature	**ICS1**	**ICS2**	**ICS3**	**ICS4**	**ICS5**	**ICS6**	**ICS7**	**ICS8**	**ICS9**	**ICS10**	**ICS11**
Basic Building Blocks											
System Component Specification	X		X	X	X	X	X	X	X	X	X
Assurance Case Specification	X		X	X	X	X	X	X		X	
Evidence Management	X	X	X			X	X	X	X	X	
Compliance Management	X		X	X		X	X		X	X	X
Architecture-Driven Assurance											
System Architecture Modelling for Assur.	X	X	X	X	X		X	X	X	X	X
Architectural Patterns for Assurance	X			X	X						
Requirements Support			X	X	X	X	X	X	X	X	
Contract-Based Assurance Composition			X	X	X	X	X		X	X	
V&V Activities			X	X	X	X	X		X	X	X
Multi-Concern Assurance											
System Dependability Co-Analysis/Assess.	X		X	X			X	X		X	X
Dependability Assurance	X		X		X			X		X	
Contract-Based Multi-Concern Assurance			X							X	
Seamless Interoperability											
Tool Integration Management			X	X	X		X			X	
Collaborative Work Management			X					X		X	
Tool Quality Assessment and Characterisation			X								
Cross- and Intra-Domain Reuse											
Reuse Assistant	X		X								X
Impact Analysis											X
Automatic Generation of Arguments	X		X			X	X	X			X
Semantic Standards Equivalence Mapping			X				X				
Reuse via Management of Variability		X	X				X				X

Table 2. Main benefits found in the application of the AMASS tool platform for each industrial case study

	Industrial Case Study										
Benefit	**ICS1**	**ICS2**	**ICS3**	**ICS4**	**ICS5**	**ICS6**	**ICS7**	**ICS8**	**ICS9**	**ICS10**	**ICS11**
Argument structure generation								X			X
Assurance case-driven system assessment								X			
Assurance-oriented system modelling	X			X	X	X			X	X	
Cross-concern variability man. and co-eng. for process reuse			X								
Early V&V							X		X	X	
Eased compliance management	X		X					X			
Eased model-based safety engineering		X	X								
Easy assurance project tailoring				X							
Easy exchange of data between tools							X				
Enhanced requirements management			X	X			X			X	
Evidence information generation	X			X		X	X			X	
Evidence reuse support	X										
Explicit assurance-targeted artefact description		X									
Explicit spec. of assurance standards and compliance means									X	X	X
Gap analysis											X
Integrated contract-based approach for system analysis and modelling		X	X							X	
Integrated evidence and assurance case management									X		
Integrated mapping of assurance reqs. and system architecture			X				X				
Integrated system assurance information				X		X	X		X	X	
Integrated system modelling and argumentation						X					
Integrated system modelling and V&V for assurance				X	X	X					
Link between system analysis results and other artefacts									X		
Multi-concern argumentation support								X			
Pattern-based system modelling	X	X		X							
Possibility of modelling both safety and security standards	X							X			
Provision of metrics and charts of an assurance project	X										X
Safety and security co-analysis	X									X	
Selection of applicable standard’s elements based on product criticality and element applicability	X										
System analysis automation							X				
Useful demonstration and training videos					X	X				X	
Useful user manual					X	X				X	

Table 3. Main improvement opportunities identified in the application of the AMASS tool platform for each industrial case study

	Industrial Case Study										
Improvement Opportunity	**ICS1**	**ICS2**	**ICS3**	**ICS4**	**ICS5**	**ICS6**	**ICS7**	**ICS8**	**ICS9**	**ICS10**	**ICS11**
Additional charts for gap analysis	X										
Additional features for assurance case management						X		X			
Additional methodological guidance									X	X	
Additional traceability capabilities			X							X	
Advanced support for system re-assessment		X					X				
Advanced system modelling features				X	X				X	X	
Better tool performance	X					X	X		X		
Enhanced usability and user interface	X			X		X	X	X	X		X
Extended functionality for system dependability analysis			X								
Extended integration between contract-based modelling and requirements specification		X									
Extended requirements specification support							X			X	
Extended V&V support							X				
Further assurance information generation						X		X			X
Further document generation				X							
Further tool interoperability possibilities	X	X	X	X				X		X	X
Higher level of automation							X				
More detailed analysis for co-assurance and co-assessment								X		X	
Other data storage alternatives						X					X
Project scheduling capabilities	X										
Support for workflow configuration					X	X	X			X	
Wizards to guide users in the different assurance and certification tasks	X					X			X		

4.2.1 Management of compliance with IEC 61508 and IEC 62443

The industrial case study ICS1 - Industrial and automation control worked on systems that control and monitor electrical infrastructures, such as primary and secondary substations. It focused on remote terminal unit devices, which are among the main elements in the control systems because they execute the commands received by the control centre, acting directly over the devices placed in the field site. Security and safety aspects are one of the primary concerns for the manufacturers and end users of remote terminal units. We are presenting the usage scenario US1-ICS1 - Management of compliance with IEC 61508 and IEC 62443.

The main activities performed for the usage scenario are:

- Creation of models of the IEC 61508 and IEC 62443 standards.
- Selection of the elements of each standard that apply to the corresponding assurance project (Figure 6), e.g. the Concept phase of IEC 61508.
- Generation of the structures of compliance arguments, e.g. to justify that Security-related activities are planned, documented, and executed.
- Definition and collection of evidence information for the assurance project, e.g. regarding the Validation plan, the Product specification, and the Design functional verification report.
- Reuse of evidence information between assurance projects, e.g. the Product design and the Validation report
- Declaration and management of compliance with the standards (Figure 7), e.g. for the evidence artefacts.

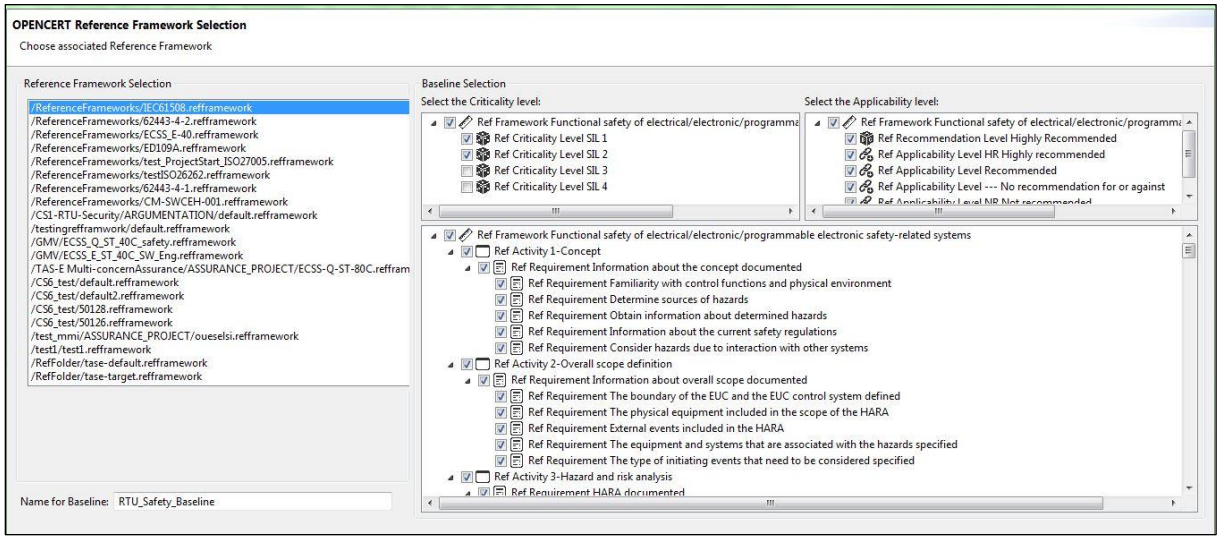

Figure 6. Selection of elements of IEC 61508 for an assurance project of an automation device

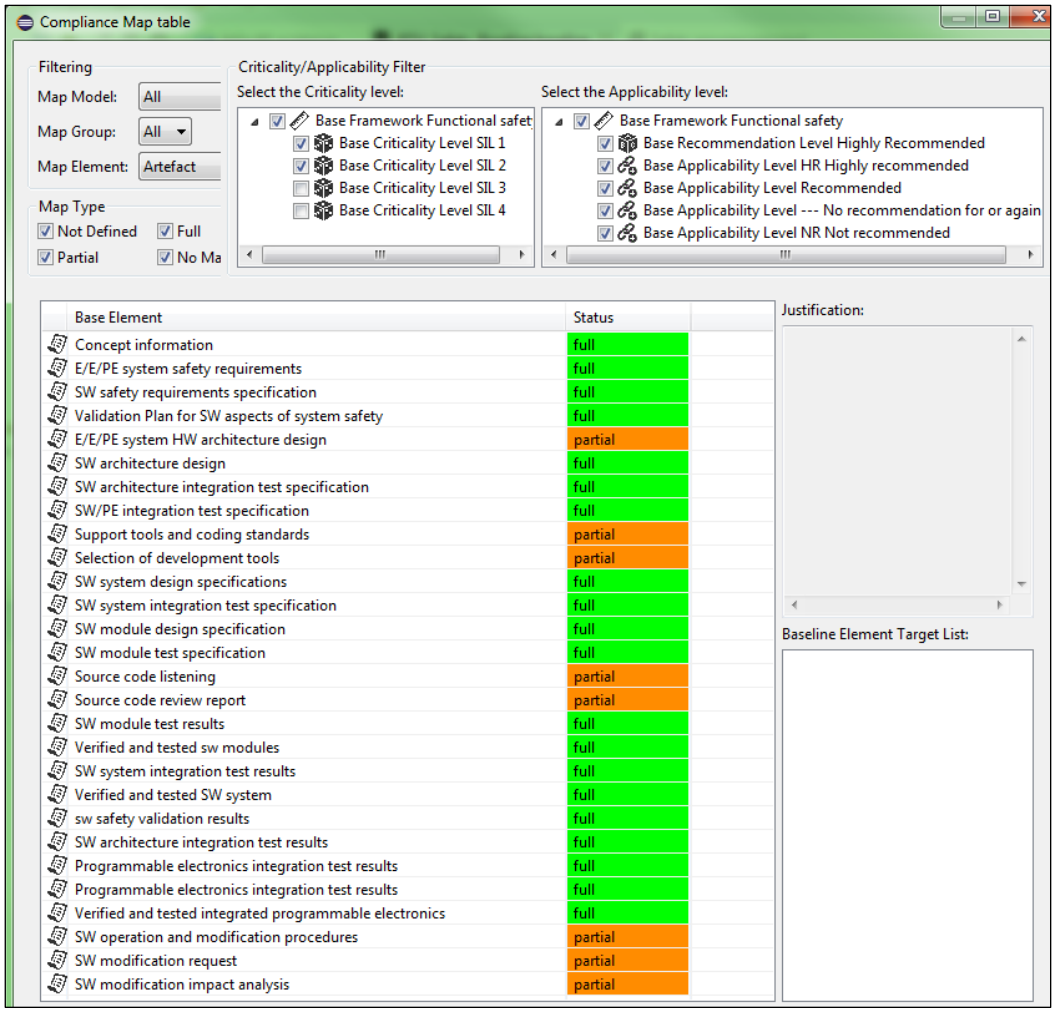

Figure 7. Compliance status for an automation device

4.2.2 Architectural Design of On-Board Software

The industrial case study ICS4 - Design and safety assessment of on-board software applications in space systems dealt with the Sentinel-3 satellite. It is an ocean and land mission to measure sea-surface topography, sea- and land-surface temperature, and ocean colour and land colour with high-end accuracy and reliability. The mission supports ocean forecasting systems, as well as environmental and climate monitoring. The first satellite of the constellation (Sentinel-3A) was launched on February 16th, 2016, whereas the second launch (Sentinel-3B) was on April 25th, 2018. We are presenting the usage scenario US1-ICS4 - Architectural design of on-board software.

The main activities performed for the usage scenario are:

- System modelling (Figure 8), including requirements and pattern-based architecture specification (Figure 9), e.g. about temperatures aspects.
- Early verification of requirements, e.g. for time properties.
- Functional refinement to define the internal structure of the components.

- Component behaviour specification by means of state machines.
- Formal verification of component behaviour with external tools, e.g. OCRA.
- Derivation of data for safety analysis information, e.g. fault tree generation.
- Specification of evidence information, e.g. regarding verification results.
- Development of assurance cases, e.g. to link them with architecture information and with verification results.
- Report generation in the form of HTML files.

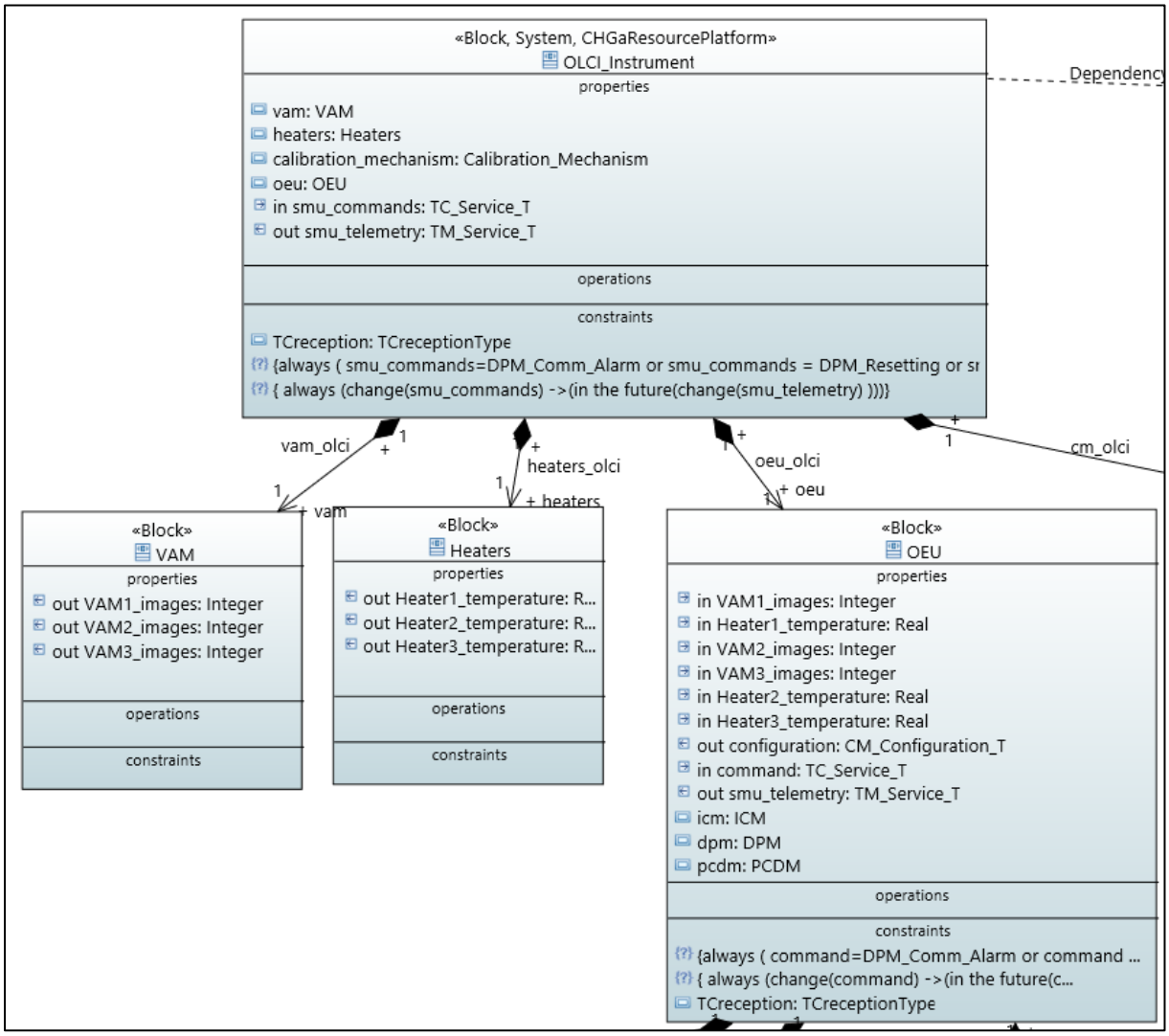

Figure 8. Fragment of a system model for satellite software

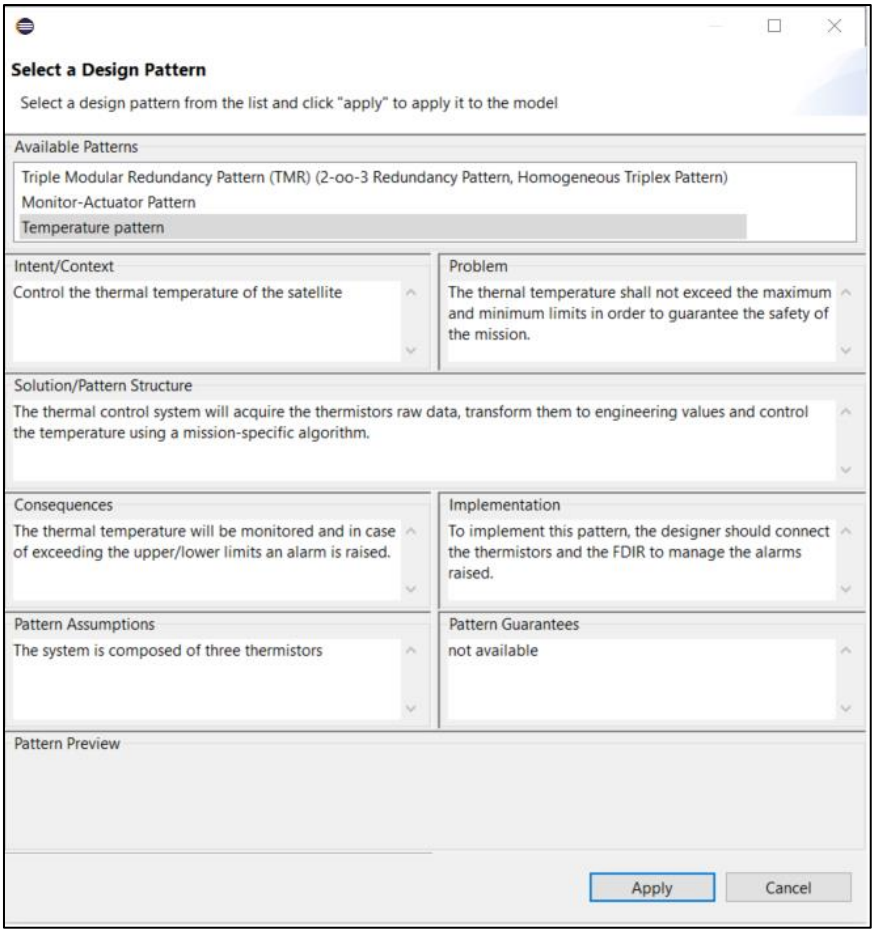

Figure 9. Selection of architectural patterns for satellite software

4.2.3 Multi-Concern Assurance Case for Safety and Security

The industrial case study ICS8 - Automotive telematics function focused on multi-concern assurance, analysis and assessment of an automotive component (element-out-of-context). The intended vehicle-level function is an automated driving one that gives a vehicle the functionality for driverless operation on controlled-access motorways. One of the components used to build the function is a positioning component which uses satellite positioning augmented with odometry to provide geographical positioning with sufficient performance while fulfilling safety and cybersecurity requirements, including compliance with the ISO 26262 and ISO/SAE 21434 standards for safety and cybersecurity, respectively. We are presenting the usage scenario US1-ICS8 - Multi-concern assurance case for safety and security.

The main activities performed for the usage scenario are:

- Creation of models of the ISO 26262 and ISO/SAE 21434 standards (Figure 10).
- Modelling of the assurance process in compliance with the standards.
- Development of assurance cases considering both safety and security aspects (Figure 11), e.g. to justify that an element is acceptably safe and secure.

- Definition and specification of evidence information, e.g. a Security risk assessment report, a Functional safety and cybersecurity communication and co-analysis plan, and a Review of safety/security cross-concern analysis.
- Declaration and management of compliance with the standards, e.g. for the evidence artefacts.
- Report generation in the form of documents.

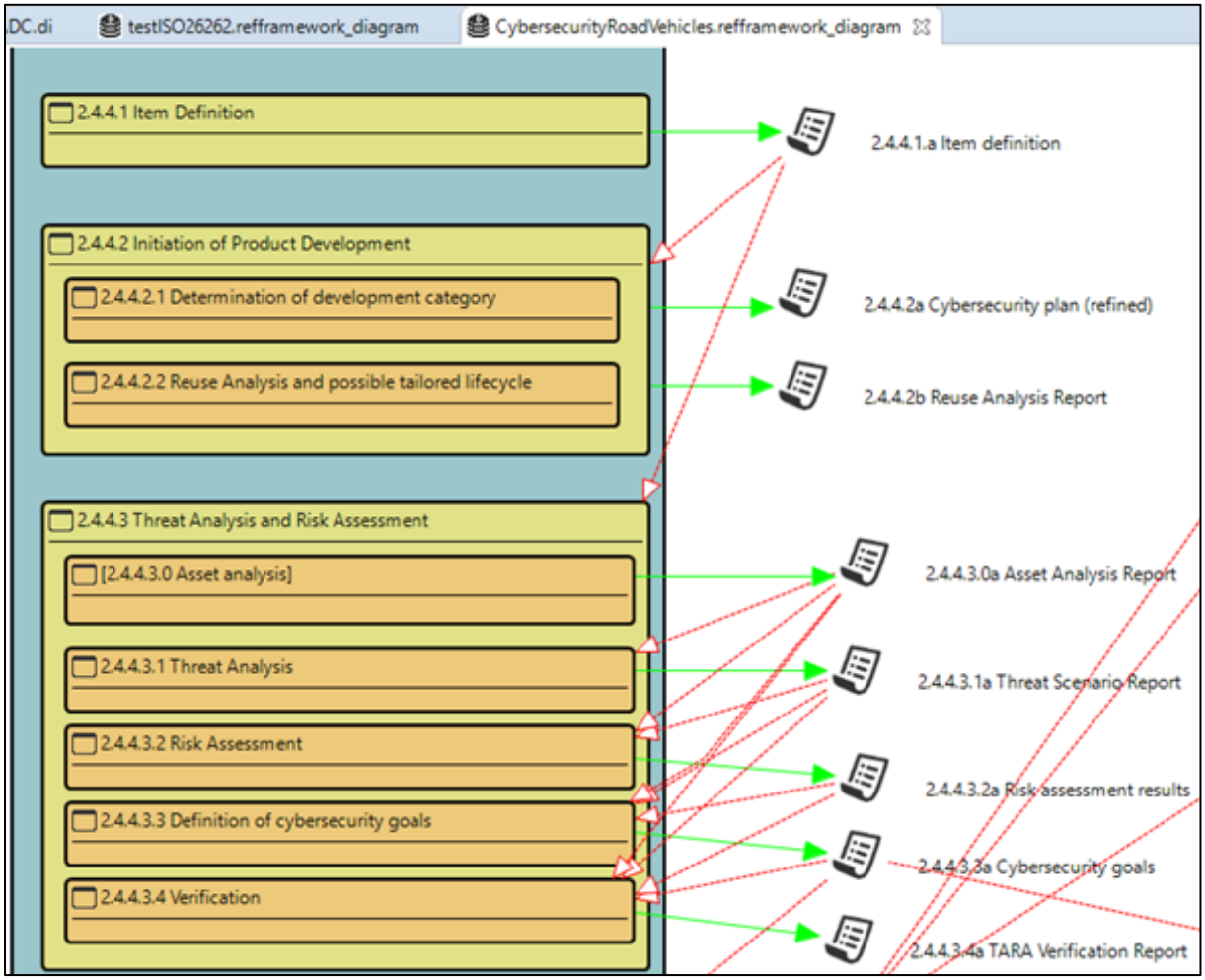

Figure 10. Fragment of a model of ISO/SAE 21434

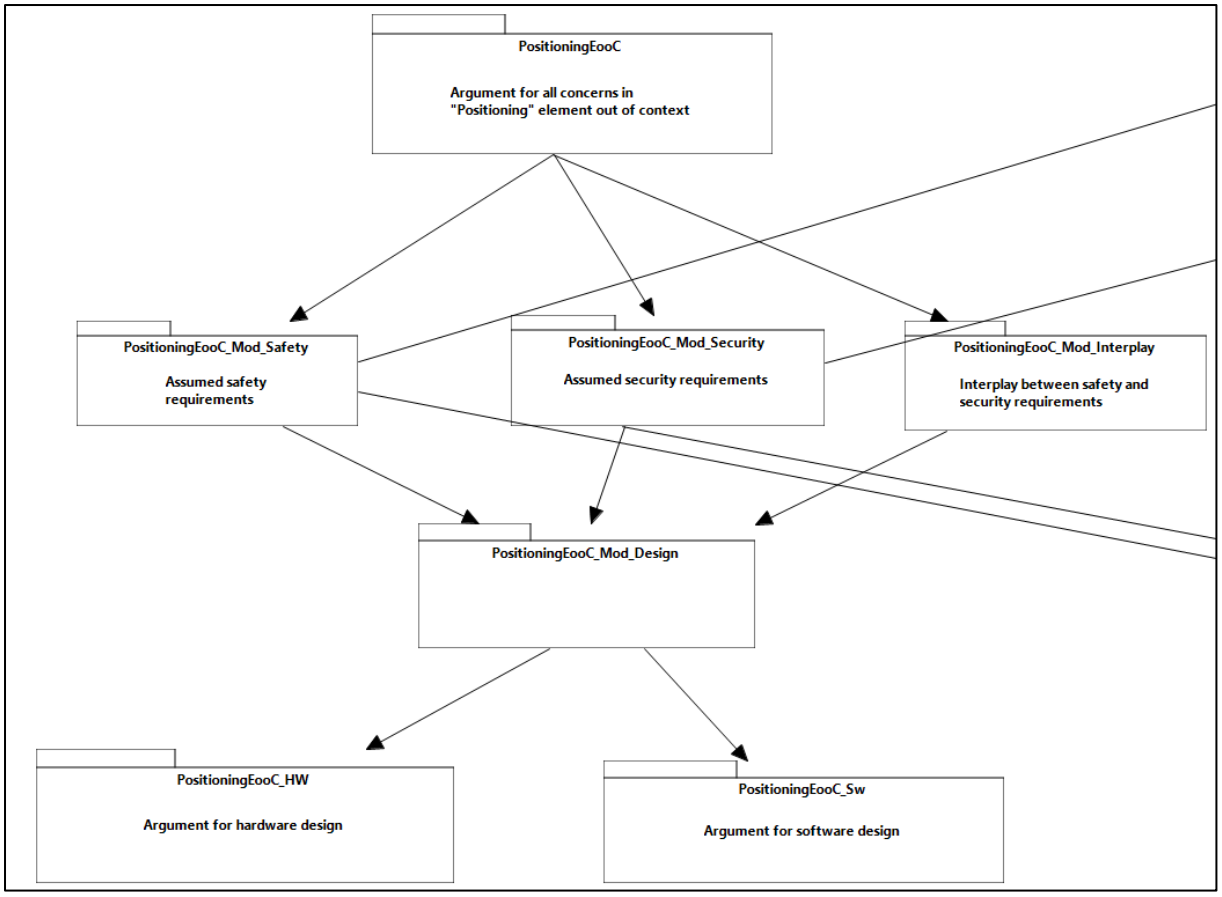

Figure 10. Fragment of argument structure for a telematics function

4.3 Discussion

This section discusses the answer to each research question formulated for the application of the AMASS tool platform. It also discusses the validity of the application.

4.3.1 Feasibility of CPS Assurance and Certification (RQ1)

We argue that the results from the application of the AMASS tool platform allows us to claim that it is a feasible means for CPS assurance and certification. The features of the platform were successfully used in 24 usage scenarios of 11 industrial case studies from aerospace, automotive, avionics, industrial automation, and railway. Each feature was demonstrated in at least one industrial case study. The broad range of systems and application domains covered also contributes to the feasibility of the approach. Although someone might argue that the existence of improvement opportunities and the possibility of addressing them negatively impacts feasibility, it is a matter of fact that practitioners were able to effectively utilise the AMASS tool platform in real CPS assurance and certification situations, i.e. they were able to successfully perform the assurance and certification activities of the industrial case studies. We do not claim that the platform is perfect or could not be enhanced, but that as currently developed it is a suitable and valid means for CPS assurance and certification.

It must be noted that the AMASS tool platform is a large tool that supports many tasks for CPS assurance and certification and its utilisation will typically have to be tailored to the

specific assurance project. Not all the systems and all the projects might need all the features, e.g. specification of structured assurance cases. It is also possible that a company decides to use a different tool for some tasks, e.g. for system modelling because the corresponding tool is already used at the company. In these cases, the tool integration capabilities of the AMASS tool platform play a major role for its adoption.

It is also important to note that depending on how the AMASS tool platform is used and the constraints on tool usage in each application domain, tool qualification aspects might need to be considered when using the platform. Tool qualification considerations have already been analysed and discussed in AMASS deliverables [10].

4.3.2 Benefits on CPS Assurance and Certification (RQ2)

We can claim that practitioners do find benefits in using the AMASS tool platform for CPS assurance and certification. A wide range of benefits were reported (31 benefits) and at least four benefits were found in each industrial case study. Benefits have been indicated for all the high-level features of the AMASS tool platform (Assurance project management, Compliance needs specification, System modelling, System dependability analysis, Assurance case management, and Evidence management). Therefore, we argue that the AMASS tool platform, and the AMASS open source ecosystem in general, can improve the state of the practice on CPS assurance and certification. In addition, the number of different benefits, the number of times that the benefits were referred to, and the average and median found benefits per industrial case study were higher than the same figures for the improvement opportunities.

System modelling appears to be the high-level feature of the platform from which practitioners find the highest benefit. This makes us wonder about the limitations of current commercial tools for CPS assurance and certification. Another possible explanation is that the practitioners were new to model-based systems engineering and recognised its potential. On the other hand, Advanced system modelling features was among the most frequently identified improvement opportunities, which might be regarded as contradictory. Assurance-oriented system modelling as a benefit and Advanced system modelling as an improvement opportunity were both referred to in four industrial case studies. These aspects could be investigated in more depth.

Some results of the benefits on CPS assurance and certification deserve a deeper analysis. We find especially interesting that six benefits refer to the integration of different types of assurance information. This is important because this integration has been one of the main objectives of the AMASS tool platform and it shows that its development has suitably advanced towards the achievement of this objective. Five benefits relate to system modelling, which has played a more central role in the AMASS project when compared to prior projects on system assurance and certification such as OPENCOSS and SafeCer. This is also rewarding as it suggests that the AMASS tool platform can fill one of the main gaps that the AMASS project had identified in the state of the art.

On the other hand, someone could have expected a larger reference of benefits on multi-concern assurance, as it was one of the main high-level areas in the AMASS project. Even more surprisingly, only two benefits referred to assurance reuse. In our opinion, these are major areas for CPS assurance and certification both nowadays and in the future. We

wonder if the AMASS tool platform either does not support yet some industrial needs, because the current support is too basic, or provides support that is too beyond current needs and practices. For the first case, a counterargument is that improvement opportunities have barely referred to multi-concern assurance and assurance reuse. For the second, certain reuse means such as variability management are not widely established for safety-critical systems yet, practitioners might not know them in depth, and thus it might be difficult for them to distinguish their benefits. Last but not least, it must be noted that the final scope of the industrial case studies could have impacted these aspects.

4.3.3 Improvement Opportunities for CPS Assurance and Certification (RQ3)

Practitioners have identified improvement opportunities on the AMASS tool platform. This is important to us because it shows that the platform can still continue being developed towards an enhanced support for CPS assurance and certification, complementing the benefits that practitioners already find. It would have been worrying that practitioners had not reported improvement opportunities, as it would have made us wonder if they had invested sufficient effort in using the AMASS tool platform. In other words, the identification of improvement opportunities shows the practitioners' interest in the availability of tool support for CPS assurance and certification that better addresses some aspects of these processes.

Most of the improvement opportunities refer to technological aspects of the AMASS tool platform, such as the inclusion of new features or the enhancement of current ones. Tool interoperability is a major aspect for the engineering and assurance of complex CPS, as many different tools and with different purposes can be used. Although the AMASS tool platform provides support for it, there exist tools with which the platform has not been integrated yet. This is not a major concern to us, as we consider that the important point is that the platform provides solutions for easy integration with further tools. Usability and user experience are very relevant and might have been impacted by the automatic application generation features that Eclipse provides. Manual user interface fine-tuning will address this issue in the future. Regarding performance, this quality aspect was constrained by the data storage approach used during the AMASS project. It was a distributed approach so that different people could work on the same industrial case study data jointly, at the same time, and at different locations. This affected the platform but can be addressed easily with other data storage configurations.

There are also improvement opportunities that relate to methodological aspects of the AMASS tool platform. Two that we find particularly relevant are Support for workflow configuration and Wizards to guide users in the different assurance and certification tasks. The AMASS tool platform is a solution that supports many needs for CPS assurance and certification, from those related to system analysis and modelling to those related to assurance cases. The needs for a concrete CPS product vary among application domains, standards, companies, and projects, thus the usage of the tool platform must be tailored to specific cases. For example, a CPS product developer might not need to develop structured assurance cases. Therefore, support within the AMASS tool platform that helps users to configure a tailored application is important to better fits industrial practices and expectations. This support is currently part of the methodological guidance and user manual [9] of the platform, but it would arguably be more suitable if embedded in the platform itself.

Another aspect that is worth analysing is that practically all the improvement opportunities can be regarded as aspects that affect users. Nonetheless, some might also affect developers, including contributors to the AMASS tool platform. These improvement opportunities include Better tool performance, Enhanced usability and user interface, Further tool interoperability possibilities, and Other data storage.

Finally, it is also important to comment that the Technology Readiness Level (TRL) [37] of the AMASS tool platform is TRL 5 (technology validated in relevant environment). Work is going on to increase the maturity level and to be able to compete more suitably against commercial tools.

4.3.4 Validity

We discuss validity according to the aspects proposed by Runeson et al. for case study research [85].

We consider that **construct validity** is largely ensured because of the adequacy of the industrial case studies on which the AMASS tool platform has been applied. They correspond to real, relevant, and representative situations for CPS assurance and certification in practice. The wide range of industrial case studies used also contributes to avoiding mono-operation bias. Nonetheless, there is the threat of mono-method bias. This could be addressed in the future by conducting experiments on CPS assurance and certification with the AMASS tool platform. Asking about both positive aspects (benefits) and negative ones (improvement opportunities) of the platform reduced the bias of those involved in the case studies towards only thinking of the advantages.

An aspect that affects **internal validity** is the lack of control during the industrial case studies. The practitioners involved in the industrial cases studies and usage scenarios performed different actions and at different moments. This heterogeneity is a threat. The specific environment and circumstances in which the practitioners used the AMASS tool platform and provided their feedback could also have impacted the results, as well as their expectations about the platform. Another threat is that the practitioners had access to several versions of the AMASS tool platform. Their final perspective on the platform might be influenced by the previous experience. The effect could be negative or positive. For example, a negative prejudice might exist for features that are improved later, or the practitioners' opinion might largely improve after realising the improvements made and the new features added. We consider that the effect is hard to analyse or predict. On the other hand, it is positive that the practitioners used the AMASS tool platform several times to avoid threats from a single use. It could even be argued that this way their opinion had a stronger basis. There is also threat in the fact some practitioners had participated in the antecessor projects, thus their expectations might correspond to what they thought that could or should be achieved from the results of these projects. More specifically, two industrial case study owners participated in the CHESS, CRYSTAL, and SafeCer projects, one in CRYSTAL, one in OPENCOSS, and one in SafeCer.

In general, case study research does not aim to broadly generalise its findings, which impacts **external validity**. However, our findings are expected to be applicable to CPS assurance and certification in situations similar to those in the industrial case studies and their usage scenarios; e.g. for CPS with the same characteristics and the same assurance

and certification needs. We also argue that the broad scope of the application of the AMASS tool platform in different industrial case studies, usage scenarios, and application domains, for different systems and standards, and by different practitioners and roles contributes to external validity. It is very uncommon to find a paper that reports the validation of some technology in 11 case studies.

Regarding **reliability**, our involvement in the industrial case studies was very limited and we did not provide feedback on the benefits and the improvement opportunities. This minimises threats from fishing. The lack of a more structured approach for feedback collection, e.g. through a predefined questionnaire, affects reliability. Higher priority was assigned in the AMASS project to ease feedback collection and provision by practitioners, and to ease their participation in the industrial case studies, with less formal means.

5. Conclusion

Assurance and certification of safety-critical systems is a requirement in most application domains, and new means are necessary to support the underlying activities. On the one hand, safety-critical systems and embedded systems are evolving towards larger, more complex, interconnected systems. New assurance approaches are necessary to suitably deal with cyber-physical aspects of the systems. On the other hand, there is a trend and an increasing interest in industry towards the adoption of open source solutions for safety-critical systems engineering, assurance, and certification.

Within this context, we have presented our experience in the development of the AMASS open source ecosystem for assurance and certification of cyber-physical systems (CPS). The ecosystem was built through the joint effort of different organizations in the scope of a research project with industry. The open source tool platform integrates and extends three main existing open source projects (OpenCert, CHESS, and EPF), leveraging Eclipse means for project development and management. As main features, the integration supports assurance project management, compliance needs specification, system modelling, system dependability analysis, assurance case management, and evidence management. The platform is further integrated with external tools. The advantages of the approach followed include no vendor-lock-in, mutualisation, sustainability, standardisation, interoperability, and adoption and innovation. These characteristics distinguish the AMASS open source ecosystem and have allowed us to set a strong basis towards a mature and sustainable platform.

Regarding the lessons learned, we strongly recommend others engaged in similar efforts to pay special attention to the possibility of integrating existing solutions, defining a strategy for growth and sustainability, addressing the integration of open source solutions for systems and software engineering with current practices and commercial tools, working on transfer and early adoption in advance, and planning the work towards following all the principles of an open source approach. We regard these areas as essential to ensure the current and future success of the AMASS open source ecosystem.

The application of the AMASS tool platform in 11 industrial case studies has allowed us to evaluate its effectiveness for CPS assurance and certification. Based on the application results, we argue that the tool platform is a feasible means for these activities. Practitioners

have also reported benefits, thus the AMASS tool platform could improve current approaches and support for CPS assurance and certification. As improvement opportunities have also been identified, the amount of advantages from using the tool platform can still grow in the future.

In synthesis, our experience with the AMASS open source ecosystem corresponds to a concrete case of the work necessary to build a successful open source project; for AMASS, combining different prior solutions towards a common objective and in the scope of CPS engineering and assurance. The decisions made and the approaches followed can guide and inspire others when having to address similar situations, as different alternatives can be chosen at a given moment of an open source project or ecosystem. This includes alternative initiatives and communities to be part of, alternative business models, and alternative strategies for growth and sustainability.

We will continue the development of the AMASS open source ecosystem and of its underlying approach for CPS assurance and certification in the future. This includes several work areas such as the use of the solutions in different projects and application domains (e.g. healthcare), the implementation of new features for specific needs (e.g. the analysis of the text of assurance standards for compliance needs determination), and a stronger link with certain engineering phases (e.g. testing). We will also continue presenting the AMASS open source ecosystem at different industrial events to reach new users and to attract new communities of developers and researchers. Last but not least, it might valuable to propose a methodology for open source ecosystem development based on our experience with the AMASS ecosystem and other initiatives.

Acknowledgement. The work leading to this paper has received funding from the AMASS (H2020-ECSEL grant agreement no 692474; Spain's MINECO ref. PCIN-2015-262), iRel4.0 (H2020-ECSEL grant agreement no 876659), VALU3S (H2020-ECSEL grant agreement no 876852), and Treasure (JCCM ref. SBPLY/19/180501/000270; EC's European Regional Development Fund) projects, and from the Ramon y Cajal Program (Spain's MICINN ref. RYC-2017-22836; EC's European Social Fund). We thank all the AMASS partners that have contributed to envisioning and to developing the AMASS open source ecosystem, especially Huascar Espinoza, Barbara Gallina, Philippe Krief, and Silvia Mazzini, who have provided feedback on the scope and content of this paper, and those involved in the industrial case studies.

References

1. Adelard: ASCE Software. https://www.adelard.com/asce/ (accessed Apr 2, 2020)
2. AIT: WEFACT, https://www.ait.ac.at/en/research-fields/dependable-systems-engineering/safety-security/wefact/ (accessed Apr 2, 2020)
3. Álvarez-Rodríguez, J.M., Mendieta, R., de la Vara, J.L., Fraga, A., Llorens, J.: Enabling system artefact exchange and selection through a Linked Data layer. Journal of Universal Computer Science 24(11): 1536-1560. 2018
4. Alves, C., Oliveira, J., Jansen, S.: Understanding Governance Mechanisms and Health in Software Ecosystems: A Systematic Literature Review. In: S. Hammoudi et al. (Eds.): ICEIS 2017, LNBIP 321, pp. 517–542. 2018.
5. AMASS Project: https://www.amass-ecsel.eu/ (accessed Apr 2, 2020)

6. 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 Apr 2, 2020)
7. AMASS Project: Deliverable 1.6 - AMASS demonstrators (c), v1.1, 2019. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D1.6_AMASS-demonstrators-%28c%29_AMASS_Final_0.pdf (accessed Apr 2, 2020)
8. AMASS Project: Deliverable D2.4 - AMASS reference architecture (c), v1.0. 2018. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D2.4_AMASS-reference-architecture-%28c%29_AMASS_Final.pdf (accessed Apr 2, 2020)
9. 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 Apr 2, 2020)
10. AMASS Project: Deliverable D2.9 - AMASS platform validation, v1.1, 2019. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/D2.9_AMASS-platform-validation_AMASS_Final.pdf (accessed Apr 2, 2020)
11. AMASS Project: Deliverable D3.6 - Prototype for architecture-driven assurance (c), v1.0. 2018. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D3.6_Prototype-for-architecture-driven-assurance-%28c%29_AMASS_Final.pdf (accessed Apr 2, 2020)
12. AMASS Project: Deliverable D4.6 - Prototype for multiconcern assurance (c) , v1.0. 2018. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D4.6_Prototype-for-multiconcern-assurance-%28c%29_AMASS_Final.pdf (accessed Apr 2, 2020)
13. AMASS Project: Deliverable D5.6 - Prototype for seamless interoperability (c), v1.0. 2018. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D5.6_Prototype-for-seamless-interoperability-%28c%29_AMASS_Final.pdf (accessed Apr 2, 2020)
14. AMASS Project: Deliverable D6.6 - Prototype for cross/intra-domain reuse (c), v1.0. 2018. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D6.6_Prototype-for-cross-intra-domain-reuse-%28c%29_AMASS_Final.pdf (accessed Apr 2, 2020)
15. AMASS Project: Deliverable D8.11 - Standardization Roadmap and Report, v1.0. 2019. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D8.11_Standardization-Roadmap-and-Report_AMASS_Final.pdf (accessed Apr 2, 2020)
16. AMASS Project: Publications. https://www.amass-ecsel.eu/content/publications (accessed Apr 2, 2020)
17. AQUAS project: http://aquas-project.eu/ (accessed Apr 2, 2020)
18. Axelsson, J., Skoglund, M.: Quality assurance in software ecosystems: A systematic literature mapping and research agenda. Journal of Systems and Software 114: 69-81. 2016

19. Biggs, G., Sakamoto, T., Kotoku, T.: A profile and tool for modelling safety information with design information in SysML. Software and Systems Modeling 15(1): 147-178. 2016
20. Blondelle, G., Bordeleau, F., Exertier, D.: PolarSys: A New Collaborative Ecosystem for Open Source Solutions for Systems Engineering Driven by Major Industry Players. Insight 18(2): 35-38. 2015
21. Bordeleau, F.: Model-Based Engineering: A New Era Based on Papyrus and Open Source Tooling. First Workshop on Open Source Software for Model Driven Engineering, OSS4MDE 2014
22. Boudjennah, C., Combemale, B., Exertier, D., Lacrampe, S., Peraldi-Frati, M.A: CLARITY: Open-Sourcing the Model-Based Systems Engineering Solution Capella. Second Workshop on Open Source Software for Model Driven Engineering, OSS4MDE 2015
23. BVR: https://github.com/SINTEF-9012/bvr (accessed Apr 2, 2020)
24. Cánovas-Izquierdo, J.L., Cabot, J.: The Role of Foundations in Open Source Projects. 40th International Conference on Software Engineering: Software Engineering in Society, ICSE 2018
25. Capella: http://www.polarsys.org/capella/ (accessed Apr 2, 2020)
26. Capra: https://projects.eclipse.org/projects/modeling.capra (accessed Apr 2, 2020)
27. CDO: https://www.eclipse.org/cdo/ (accessed Apr 2, 2020)
28. CertWare: https://nasa.github.io/CertWare/ (accessed Apr 2, 2020)
29. Cusumano, M. A., Gawer, A.: The elements of platform leadership. MIT Sloan Management Review 43(3) 51-58. 2002
30. CHESS Project: http://www.chess-project.org/ (accessed Apr 2, 2020)
31. CRYSTAL Project: http://www.crystal-artemis.eu/ (accessed Apr 2, 2020)
32. de la Vara, J.L., Falessi, D., Verhuslt, E.: Specifying a Framework for Evaluating Requirements Engineering Technology: Challenges and Lessons Learned. IEEE 3rd International Workshop on Empirical Requirements Engineering, EmpiRE 2013
33. de la Vara, J.L., Borg, M., Wnuk, K., Moonen, L.: An Industrial Survey on Safety Evidence Change Impact Analysis Practice. IEEE Transactions on Software Engineering 42(12): 1095 – 1117. 2016
34. de la Vara, J.L., Ruiz, A., Attwood, K., Espinoza, H., Panesar-Walawege, R.K., Lopez, A., del Rio, I., Kelly, T.: Model-Based Specification of Safety Compliance Needs: A Holistic Generic Metamodel. Information and Software Technology 72: 16-30. 2016
35. de la Vara, J.L, Ruiz, A., Gallina, B., Blondelle, G., Alaña, E., Herrero, J., Warg, F., Skoglund, M., Bramberger, R.: The AMASS approach for assurance and certification of critical systems. embedded world Conference 2019
36. DEOS: D-Case Editor - A Typed Assurance Case Editor. http://www.jst.go.jp/crest/crest-os/tech/D-CaseEditor/index-e.html (accessed Apr 2, 2020)
37. EC: Technology readiness levels (TRL). https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf (accessed Apr 2, 2020)
38. Eckhardt, E., Kaats, E., Jansen, S., Alves, C.: The Merits of a Meritocracy in Open Source Software Ecosystems. ECSA Workshops 2014
39. Eclipse: https://www.eclipse.org/ (accessed Apr 2, 2020)

40. Eclipse Foundation: Development Resources/HOWTO/Parallel IP Process. http://wiki.eclipse.org/Development_Resources/HOWTO/Parallel_IP_Process (accessed Apr 2, 2020)
41. Eclipse: Eclipse Development Process 2015. https://www.eclipse.org/projects/dev_process/development_process.php (accessed Apr 2, 2020)
42. Eclipse Process Framework Project: https://www.eclipse.org/epf/ (accessed Apr 2, 2020)
43. Falessi, D., Sabetzadeh, M., Briand, L., Turella, E., Coq, T., Panesar-Walawege, R.K.: Planning for Safety Standards Compliance: A Model-Based Tool-Supported Approach. IEEE Software 29(3): 64-70. 2012
44. Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X.: QuESo - A Quality Model for Open Source Software Ecosystems. 9th International Conference on Software Engineering and Applications, ICSOFT 2014
45. Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X.: Open source software ecosystems: A Systematic mapping. Information and Software Technology 91:160–185. 2017
46. Gallina, B., Gómez-Martínez, E., Benac-Earle, C.: Promoting MBA in the rail sector by deriving process-related evidence via MDSafeCer. Computer Standards & Interfaces 54: 119-128. 2017
47. IBM: IBM Engineering Systems Design Rhapsody. https://www.ibm.com/us-en/marketplace/systems-design-rhapsody (accessed Apr 2, 2020)
48. Jansen, S.: Measuring the health of open source software ecosystems: Beyond the scope of project health. Information and Software Technology 56: 1508–1519. 2014
49. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: A research agenda for software ecosystems. 31st International Conference on Software Engineering, ICSE 2009
50. Kilamo, T., Hammouda, I., Mikkonen, T., Aaltonen, T.: From proprietary to open source - Growing an open source ecosystem. Journal of Systems and Software 85: 1467–1478. 2012
51. Knight, J. C.: Safety critical systems: challenges and directions. 24rd International Conference on Software Engineering, ICSE 2002
52. LDRA: LDRA Compliance Management System. https://ldra.com/aerospace-defence/products/ldra-compliance-management-system-lcms/ (accessed Apr 2, 2020)
53. Lima, T., Pereira dos Santos R., Oliveira, J., Werner, C.: The importance of socio-technical resources for software ecosystems management. Journal of Innovation in Digital Ecosystems 3(2): 98-113. 2016
54. Linåker, J., Munir, H., Wnuk, K., Mols, C.E.: Motivating the contributions: An Open Innovation perspective on what to share as Open Source Software. Journal of Systems and Software 135: 17–36. 2018
55. Lundell, B., Lings, B., Syberfeldt, A.: Practitioner perceptions of Open Source software in the embedded systems area. Journal of Systems and Software 84: 1540–1549. 2011
56. Maksimov. M., Fung, N.L.S., Kokaly,S., Chechik, M.: Two Decades of Assurance Case Tools: A Survey. SAFECOMP Workshops 2018
57. Maksimov, M., Kokaly,S., Chechik, M.: A Survey of Tool-supported Assurance Case Assessment Techniques. ACM Computing Surveys 52(5): 101. 2019

58. Manikas, K., Hansen, K.M: Software ecosystems – A systematic literature review. Journal of Systems and Software 86: 1294–1306. 2013
59. Manikas, K.: Revisiting software ecosystems Research: A longitudinal literature study. Journal of Systems and Software 117: 84–103. 2016
60. Maro, S., Steghöfer, J.P., Staron, M.: Software traceability in the automotive domain: Challenges and solutions. Journal of Systems and Software 141: 85-110. 2018
61. Mazzini, S., Favaro, J., Puri, S., Baracchi, L.: CHESS: an open source methodology and toolset for the development of critical systems. 3rd International Workshop on Open Source Software for Model Driven Engineering, OSS4MDE 2016
62. Munir, H., Runeson, P., Wnuk, K.: A theory of openness for software engineering tools in software organizations. Information and Software Technology 97: 26–45. 2018
63. Nair, S., de la Vara, J.L., Sabetzadeh, M., Briand, L.: An Extended Systematic Literature Review on Provision of Evidence for Safety Certification. Information and Software Technology 56(7): 689-717. 2014
64. Nair, S., de la Vara, J.L., Sabetzadeh, M., Falessi, D.: 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
65. No Magic: Magic Draw (accessed Apr 2, 2020)
66. OMG: Structured Assurance Case Metamodel, version 2.0. 2018
67. OpenCert: https://www.polarsys.org/opencert/ (accessed Apr 2, 2020)
68. OpenCert: Downloads. https://polarsys.org/opencert/downloads/ (accessed Apr 2, 2020)
69. OpenCert: Resources. https://polarsys.org/opencert/resources/ (accessed Apr 2, 2020)
70. OPENCOSS Project: http://www.opencoss-project.eu/ (accessed Apr 2, 2020)
71. OSLC: https://open-services.net/ (accessed Apr 2, 2020)
72. Panesar-Walawege, R.K., Sabetzadeh, M., Briand, L.: Supporting the verification of compliance to safety standards via model-driven engineering: Approach, tool-support and empirical validation. Information & Software Technology 55(5): 836-864. 2013
73. Papyrus: https://www.eclipse.org/papyrus/ (accessed Apr 2, 2020)
74. Papyrus IC: Product Management. https://wiki.polarsys.org/Papyrus_IC/Product_Management (accessed Apr 2, 2020)
75. Parra, E., Alonso, L., Mendieta, R., de la Vara, J.L.: Advances in Artefact Quality Analysis for Safety-Critical Systems. 30th International Symposium on Software Reliability Engineering, ISSRE 2019
76. Poba-Nzaou, P., Uwizeyemungu, S.: Barriers to Mission-Critical Open Source Software Adoption by Organizations: A Provider Perspective. 19th Americas Conference on Information Systems, AMCIS 2013
77. PolarSys: https://www.polarsys.org/ (accessed Apr 2, 2020)
78. PolarSys CHESS: https://www.polarsys.org/projects/polarsys.chess (accessed Apr 2, 2020)
79. PTC: Windchill Process Director. https://www.ptc.com/en/products/plm/plm-products/windchill/process-director (accessed Apr 2, 2020)
80. RobMoSys project: https://robmosys.eu/ (accessed Apr 2, 2020)
81. RTCA: DO-178C: Software Considerations in Airborne Systems and Equipment Certification. 2012

82. Ruiz, A., Juez, G., Espinoza, H., de la Vara, J.L., Larrucea, X.: Reuse of safety certification artefacts across standards and domains: A systematic approach. Reliability Engineering and System Safety 158: 153-171. 2017
83. Ruiz, A., Gallina, B., de la Vara, J.L., Mazzini, S., Espinoza, H: Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems. SAFECOMP Workshops 2016
84. Ruiz, J.F., Comar, C.: Open-DO: Open Framework for Critical Systems. 3rd International Workshop on Foundations and Techniques for Open Source Software Certification, OpenCert 2009
85. Runeson, P., Höst, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering - Guidelines and Examples. Wiley, 2012
86. Russo, D.: Benefits of Open Source Software in Defense Environments. 4th International Conference in Software Engineering for Defence Applications, SEDA 2015
87. SafeCer Project: https://artemis-ia.eu/project/40-nsafecer.html (accessed Apr 2, 2020)
88. Schreiber, A., Haupt, C.: Raising Awareness about Open Source Licensing at the German Aerospace Center. 2018 IEEE Aerospace Conference
89. SEAM: Systems Engineering and Assurance Modeling. https://modelbasedassurance.org/ (accessed Apr 2, 2020)
90. Shahrivar, S., Elahi, S., Hassanzadeh, A., Montazer, G.: A business model for commercial open source software: A systematic literature review. Information and Software Technology 103: 202-214. 2018
91. Stol, K.J., Ali Babar, M.: Challenges in Using Open Source Software in Product Development: A Review of the Literature. 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, FLOSS 2010
92. Sulaman, S.M., Orucevic-Alagic, A., Borg, M., Wnuk, K., Höst, M., de la Vara, J.L.: Development of Safety-Critical Software Systems Using Open Source Software - A Systematic Map. 40th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2104
93. The Open Group: Dependability through Assuredness (O-DA) Framework. 2013
94. The REUSE Company: RQA - Quality Studio, https://www.reusecompany.com/rqa-quality-studio (accessed Apr 2, 2020)
95. Valença, G., Alves, C.: A theory of power in emerging software ecosystems formed by small-to-medium enterprises. Journal of Systems and Software 134: 76–104. 2017
96. Valença, G., Alves, C., Jansen, S.: Strategies for managing power relationships in software ecosystems. Journal of Systems & Software 144: 478–500. 2018
97. Wei, R., Kelly, T., Dai, X., Zhao, S., Hawkins, R.: Model based system assurance using the structured assurance case metamodel. Journal of Systems and Software 154: 211-233. 2019