ECSEL Research and Innovation actions (RIA)

AMASS

Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems

Baseline and requirements for multi-concern assurance
D4.1

Work Package:	WP4: Multi-Concern Assurance
Dissemination level:	PU
Status:	Final
Date:	30 March 2018
Responsible partner:	Frédérique Vallée and Mohamed Bakkali (A4T)
Contact information:	mohamed.bakkali@all4tec.net
Document reference:	AMASS_D4.1_WP4_A4T_V1.1

This deliverable is part of a project that has received funding from the ECSEL JU under grant agreement No 692474. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and from Spain, Czech Republic, Germany, Sweden, Italy, United Kingdom and France

Contributors

Names	Organisation
Frédérique Vallée,Mohamed Bakkali and Marc Sango	A4T
Fredrik Warg	SPS
Irfan Sljivo and Barbara Gallina	MDH
Jose Luis de la Vara, Victor Sacristán	Universidad Carlos III de Madrid
Christoph Schmittner, Thomas Gruber, Zhendong Ma and Petr Böhm	AIT
Morayo Adedjouma	CEA
Stefano Puri	INT
Garazi Juez, Alejandra Ruiz, Huascar Espinoza	TECNALIA
Helmut Martin	VIF
Benito Caracuel	TLV

Document History

Version	Date	Status	Author (Partner)	Remarks
V1.0	2016-09-30	Final version	G. Juez (TEC), H. Espinoza (TEC), M. Bakkali (A4T), T. Gruber (AIT), Christoph Schmittner (AIT), Zhendong Ma (AIT)	Final
V1.1	2018-03-30	Final version	Mark Sango (A4T)	Revised to take into account comments from EC reviewers.

TABLE OF CONTENTS

List of Figures

List of Tables

Executive Summary

The deliverable D4.1 (Baseline and requirements for multi-concern assurance) is the output of the task T4.1, which falls within the scope of the Scientific and Technical Objective 2 (STO2) of AMASS, which focuses on Multi-concern Assurance. In the AMASS project, we aim to exploit the existing OPENCOSS and SafeCer approaches and extend them to provide a tool-supported methodology for the development of assurance cases which address multiple system characteristics (mainly safety and security, but also other dependability aspects such as availability, robustness and reliability). Therefore, this document exposes some work results in the state of the art and the state of the practice of multi-concern assurance.

This deliverable presents the concepts and main challenges (Section2) when facing with multi-concern assurance during the development of cyber-physical systems (CPS). We identified three main challenges:

- *Dependability Assurance Modelling*. This is related with the needs to enrich the concept of "Assurance Case" with multi-concern aspects: dependencies, overlapping, contradictory arguments, and the like. We must find the mechanisms, means and guidelines to model dependencies, overlapping, contradictory goals/claims, etc.
- *Contract-Based Multi-concern Assurance*. The challenge is to extend the AMASS "contract-based" compositional solution (related to WP3 in the project) so that we make it versatile enough in order to support various kind of properties (safety, security, reliability, etc.) in assumptions/guarantees.
- *System Dependability Co-Analysis/Assessment*. This topic relates to understanding the interplay between security and safety (and other concerns) while evaluating and designing CPS architectures.

We report the state of the art in these areas by looking first to all dependability attributes (Section3) and the related work of multi-concern design and assurance in the different phases of the CPS lifecycle. We then look to the specific cases of safety and security co-engineering as developed in the literature. This case is highly relevant to AMASS since the increasing amount of cyber-attacks around the world demonstrates that safety-critical systems are not that safe as the safety engineering community pretend, if those critical systems are not enough secure.

We then look at the state of the practice (Section4) in multi-concern assurance. In this section, domain-specific standards with dependability attributes including safety and security are collected and analysed. The aim of this review is to gain an overview of the perspectives and approaches to dependability in different domains, in order to facilitate the development of multi-concern assurance concept and toolchain in the rest of the project.

We finally summarize the main findings and way forward (Section5) in this WP. The AMASS project requirements related to multi-concern assurance are being collected in deliverable D2.1, as part of the whole project requirements.

This deliverable in the input of the task 4.2 (Conceptual approach for Multi-concern Assurance) and is related to the D4.2.

1. Introduction

The aim of this report is to describe existing approaches for multi-concern assurance, particularly paying attention to the integration of multiple concerns within a model-based assurance framework. This deliverable presents the results of the state-of-the-art survey of multi-concern assurance. It focuses on the identification of relevant assurance concerns for inclusion in the AMASS framework, and on the identification of areas where the “mapping” technology developed in OPENCOSS can most effectively be deployed in AMASS.

This introductory chapter is aimed at recalling the context of the AMASS project as well as the objectives and expected results that pertain to this document.

Embedded systems have significantly increased in number, technical complexity, and sophistication, moving towards open, interconnected, networked systems (such as "the connected car" and the cloud), integrating the physical and digital world, thus justifying the term “Cyber-Physical Systems” (CPS). This “Cyber-Physical” dimension is exacerbating the problem of ensuring safety, security, availability, robustness and reliability in the presence of human, environmental and technological risks. Furthermore, the products into which these Cyber-Physical Systems (CPS) are integrated (e.g. aircrafts) need to respect applicable standards for assurance and in some areas, they even need certification.

Unlike practices in electrical and mechanical equipment engineering, CPS do not have a set of standardized and harmonized practices for assurance and certification that ensure safe, secure and reliable operation with typical software and hardware architectures. As a result, the CPS community often finds it difficult to apply existing certification guidance. Ultimately, **the pace of assurance and certification will be determined by the ability of both industry and certification/assessment authorities to overcome technical, regulatory, and operational challenges.** A key regulatory-related challenge has to be faced when trying to reuse CPS products from one application domain in another because they are constrained by different standards and the full assurance and certification process must be applied as if it were a totally new product, thus reducing the return on investment of such reuse decisions. Similarly, reuse is hindered often even within the same domain, when trying to reuse CPS products from one project to another, where assumptions change together with the criticality level.

To face all these challenges, the AMASS approach focuses on the development and consolidation of an open and holistic assurance and certification framework for CPS, which constitutes the evolution of the OPENCOSS and SafeCer approaches towards an architecture-driven, multi-concern assurance, and seamlessly interoperable tool platform.

The AMASS tangible expected results are:

a) The **AMASS Reference Tool Architecture,** which will **extend the OPENCOSS and SafeCer conceptual, modelling and methodological frameworks** for architecture-driven and multi-concern assurance, as well as **for further cross-domain and intra-domain reuse capabilities** and seamless interoperability mechanisms (based on OSLC specifications).

b) The **AMASS Open Tool Platform**, which will correspond to a collaborative tool environment supporting CPS assurance and certification. **This platform represents a concrete implementation of the AMASS Reference Tool Architecture, with a capability for evolution and adaptation, which will be released as an open technological solution by the AMASS project. AMASS openness is based on both standard OSLC APIs with external tools (e.g. engineering tools including V&V tools) and on open-source release of the AMASS building blocks.**

c) The **Open AMASS Community**, which will **manage the project outcomes, for maintenance, evolution and industrialization**. The Open Community will be supported by a governance board, and by rules, policies, and quality models. This includes support for **AMASS base tools** (tool infrastructure for database and access management, among others) and **extension tools** (enriching

AMASS functionality). As Eclipse Foundation is part of the AMASS consortium, the **Polarsys/Eclipse community** (www.polarsys.org) is a strong candidate to host AMASS.

To achieve the AMASS results, as depicted in Figure 1, the multiple challenges and corresponding project scientific and technical objectives are addressed by different work-packages.

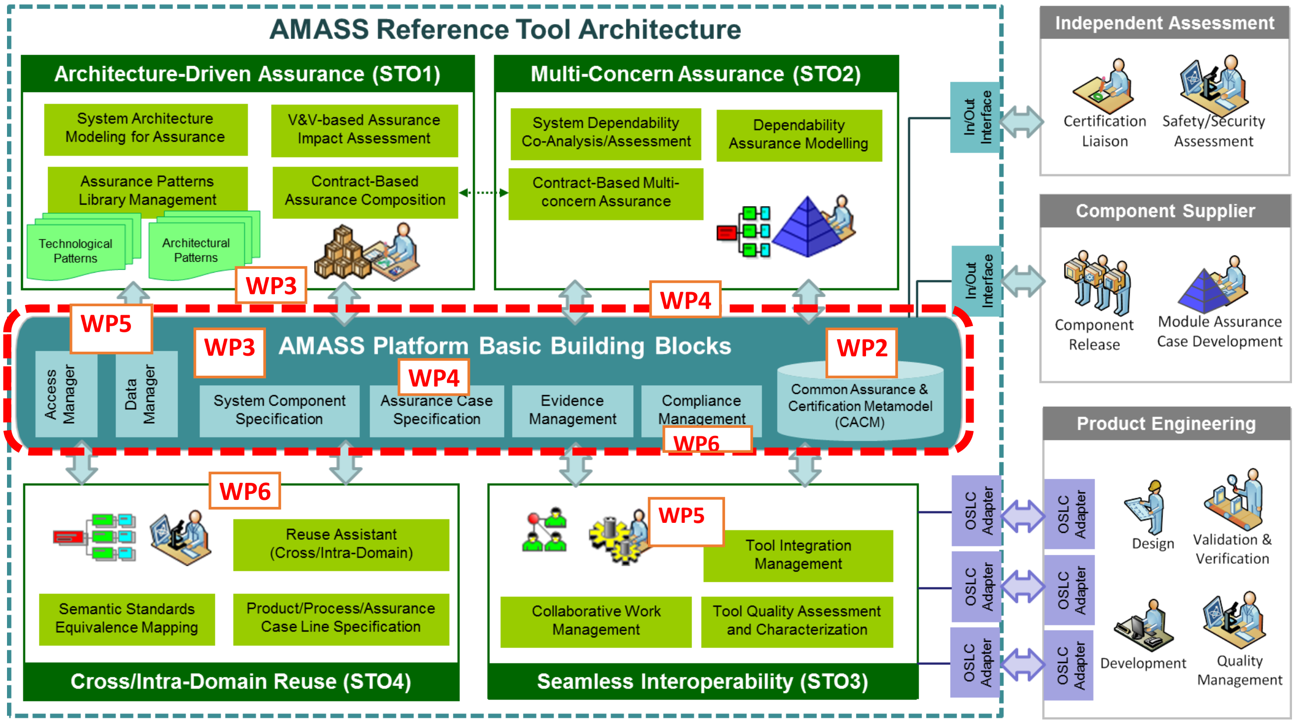

AMASS Reference Tool Architecture
Architecture-Driven Assurance (STO1)
System Architecture Modeling for Assurance
V&V-based Assurance Impact Assessment
Assurance Patterns Library Management
Contract-Based Assurance Composition
Technological Patterns
Architectural Patterns
WP3
Multi-Concern Assurance (STO2)
System Dependability Co-Analysis/Assessment
Dependability Assurance Modelling
Contract-Based Multi-concern Assurance
WP4
In/Out Interface
In/Out Interface
Independent Assessment
Certification Liaison
Safety/Security Assessment
Component Supplier
Component Release
Module Assurance Case Development
WP5
WP3
AMASS Platform Basic Building Blocks
WP2
Access Manager
Data Manager
System Component Specification
WP4
Assurance Case Specification
Evidence Management
Compliance Management
WP6
Common Assurance & Certification Metamodel (CACM)
WP6
Reuse Assistant (Cross/Intra-Domain)
Semantic Standards Equivalence Mapping
Product/Process/Assurance Case Line Specification
Cross/Intra-Domain Reuse (STO4)
WP5
Tool Integration Management
Collaborative Work Management
Tool Quality Assessment and Characterization
Seamless Interoperability (STO3)
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
OSLC Adapter
Product Engineering
Design
Validation & Verification
Development
Quality Management

Figure 1: AMASS Building blocks

WP4 aims at addressing multi-concern assurance. More specifically, with respect to the AMASS goals, this deliverable presents the background in terms of problem and solution space related to: Goal 3 (G3), the corresponding project objective O2, and to the project scientific and technical objective (STO) 2. G3, O2 and STO2 are recalled here to make the deliverable self-contained.

G3: to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification risks of new CPS products.

O2: define a multi-component assurance approach to ensure not only safety and security, but also other dependability aspects such as availability, robustness and reliability.

STO2, which focuses on multi-concern assurance, is constituted of three sub-objectives:

- Dependability Assurance Modelling.
- Contract-Based Multi-concern Assurance.
- System Dependability Co-Analysis/Assessment.

In addition, WP4 is responsible for consolidating the previous works on single-concern assurance as well as multi-concern assurance in order to design and implement the basic building block called "Assurance Case Specification" (Figure 1). Assurance Case Specification builds upon the industrial standards. That is: (a) what the standards define and (b) how assurance cases can be structured as well as how single-concern cases can interplay.

To achieve STO2, WP4 is structured into four tasks. The purpose of this deliverable is to document the work conducted during Task T4.1 (Consolidation of the Current Approaches Multi-Concern Assurance). More specifically, the purpose of the deliverable is multi-fold:

1) to analyse the problem related to multi-concerns in order to understand its multifaceted nature;
2) to present a corresponding state of the art;
3) to present the current state of practice; and finally, based on these findings;
4) to present a consolidation of existing results and profit from ongoing and past projects as well as available technology in the market are proposed.

More specifically, based on the investigated state of the art and state of practice approaches, their gaps are identified to come up with a way forward enabling the formulation of requirements to achieve the multi-concern-oriented vision of AMASS covering crucial concerns as well their trade-offs. This activity will serve to ensure both the innovation of the project and future feasibility of exploitation of results.

2. Problem Statement and Concepts

2.1 Dependability Background

Jean-Claude Laprie introduced in [1] a set of basic definitions of dependability covering various system attributes (cf. Figure 2). The author states that dependability is defined as the trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers. The service delivered by a system is its behaviour, as it is perceived by its user; a user is another type of system (human of physical) which interacts with the former.

Depending on the application intended for the system, different emphasis may be put on different facets of dependability, i.e. dependability may be viewed according to different, but complementary, properties, which enable the attributes of dependability to be defined:

- With respect to readiness for usage, dependable means available;
- With respect to continuity of service, dependable means reliable;
- With respect to avoidance of catastrophic consequences on the environment, dependable means safe;
- With respect to the prevention of unauthorized access and/or handling of information, dependable means secure.

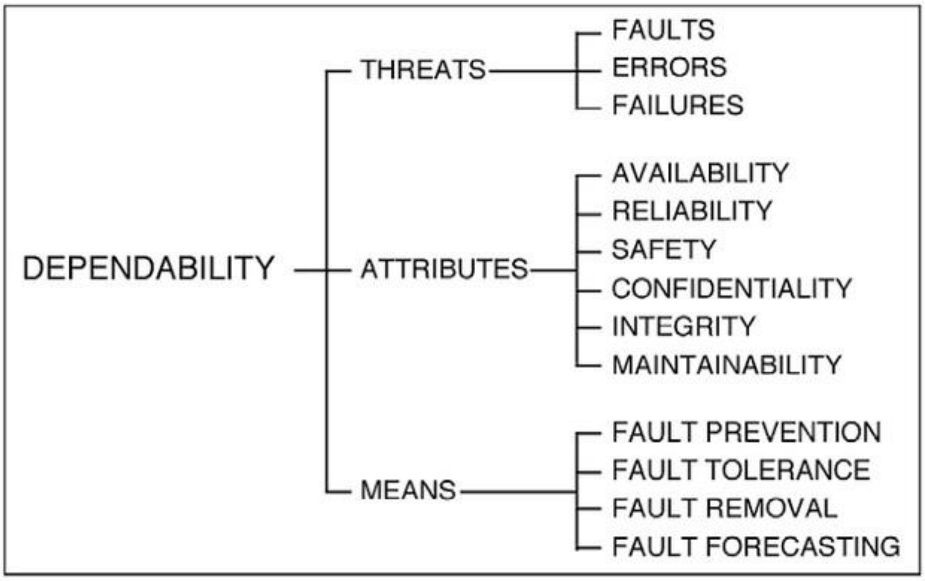


Figure 2: Dependability – Basic Concepts and Terminology [1]

In the same line of that definition where security is considered as an attribute of dependability, the work presented in [2] provides a novel approach to security, intended to facilitate and improve this integration. This is accomplished by taking a dependability viewpoint on traditional security and interpreting it in terms of system behaviour and fault prevention (cf. Figure 3). The author defines a modified security concept, comprising only fault prevention characteristics and a new behaviouristic concept, privacy. He also claims that the outcomes of this interpretation will influence the integration of the other three dependability attributes.

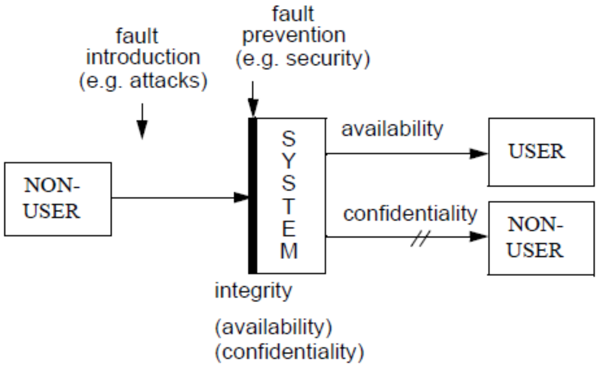


Figure 3: Understanding security in dependability terms [2]

On the other hand, in the work presented in [3] the author does not consider security as a sub-attribute of dependability. He begins by giving the main definitions relating to dependability, a generic concept including such attributes as reliability, availability, safety, integrity, maintainability, etc. and considers dependability as an integrating concept that encompasses the following attributes:

- Availability: readiness for correct service;
- Reliability: continuity of correct service;
- Safety: absence of catastrophic consequences on the user and the environment;
- Integrity: absence of improper system alterations;
- Maintainability: ability to undergo modifications and repairs;
- Security: brings in concerns for confidentiality, in addition to availability and integrity.

The author claims that when addressing security, an additional attribute has great prominence, confidentiality, i.e. the absence of unauthorized disclosure of information. Security is a composite of the attributes of confidentiality, integrity, and availability requiring the concurrent existence of 1) availability for authorized actions only, 2) confidentiality, and 3) integrity with "improper" meaning "unauthorized". Figure 4 summarizes the relationship between dependability and security in terms of their principal attributes.

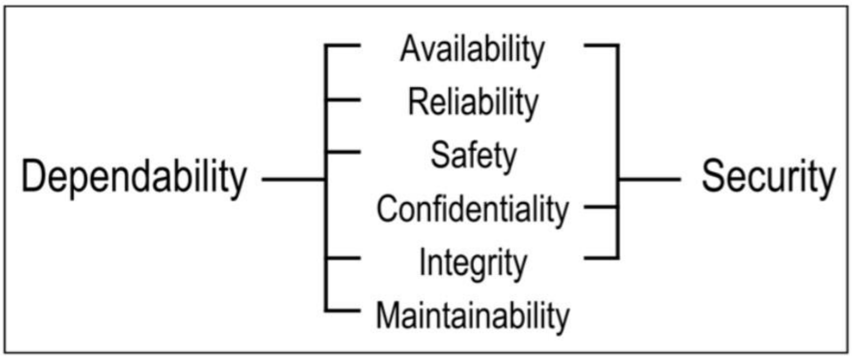


Figure 4: Dependability and security attributes [3]

It is commonly accepted that security and dependability largely represent two different aspects of an overall meta-concept that reflects the trust that we put in a computer system. There exist a large number of models of security and dependability with various definitions and terminology. This position presented in [4] suggests a high-level conceptual model that is aimed to give a novel approach to the area. The model defines security and dependability characteristics (Figure 5) in terms of a system's interaction with its environment via the system boundaries and attempts to clarify the relation between malicious environmental influence, e.g. attacks, and the service delivered by the system. The model is intended to help reasoning about security and dependability and to provide an overall means for finding and applying fundamental defence mechanisms. Since the model is high-level and conceptual, it must be interpreted into each specific sub-area of security/dependability to be practically useful.

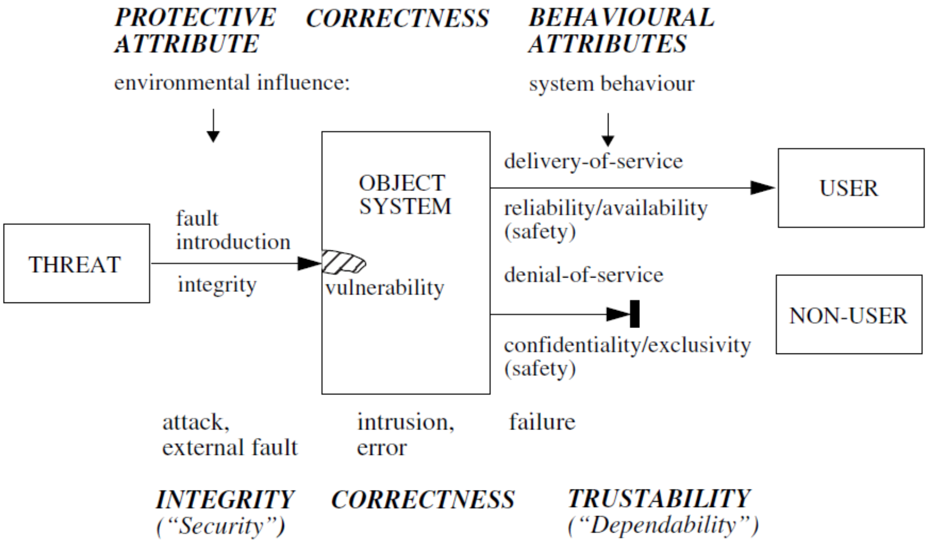


Figure 5: An integrated model of security and dependability [4]

2.2 Safety and Security Co-Engineering Background

As the AMASS project will focus initially on extending the OPENCOSS and SafeCer approaches to address those aspects of security which impact on safety issues for-dependability-critical cyber-physical systems, it is important to focus on integration of safety and security.

Among other dependability attributes, it is essential to underline the synergies between safety and security concerns, especially those aspects of security, which affect safety issues.

In the literature, several work [15][16][17][18][19][20][21] have been proposed to tackle the synergies between safety and security. Whereas functional safety is part of the overall safety (freedom from unacceptable risk of harm) that depends on a system or equipment operating correctly in response to its inputs, security is concerned with the protection of assets from threats, where these are categorized as "the potential for abuse of protected assets". Thus, safety requires protection from (unintentional) malfunctions while security does it from (deliberate) attacks. In fact, information security for safety related system has become a real issue and they seem to be in a kind of Ying Yang relationship. A remarkable example would be the one related to the cockpit cabin: from a security point of view, it should be locked whereas from the safety point of view it should open in case of emergencies. This concept is addressed by the SEMA paradigm [16] where the six boxes give the sub-notions for the domain organized according to the system - environment, and the malicious - accidental dimensions: Fail-safe behaviour is important from a safety perspective but conflicts with the security requirement of availability. Together, they constitute the SEMA framework, which can be used to clarify terms and ambiguities between safety and security, in order to "to analyse the consequences of their co-existence when dealing with the notions of security and safety in a multi-domain, cross-cultural environment".

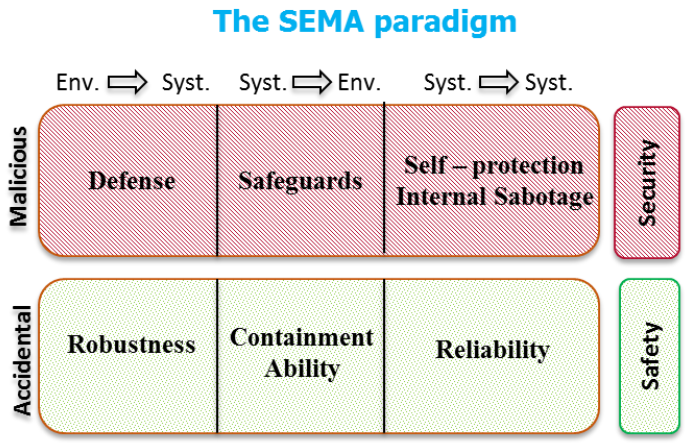


Figure 6: The SEMA paradigm [16]

In fact, safety and security share fundamental important concepts, which can be inherited from dependability. Since it is important to understand the potential synergies between safety and security, the goals of both dependability attributes are analysed in the following table:

Table 1: Safety and Security Goals

Goals	Safety	Security
Integrity	Demands the correct operation of the system under all defined circumstances with in a fixed period of time. Divided into stochastic (hardware) integrity and systematic integrity.	Unauthorized entity must not be able to change data without being detected
Authentication	Demands that message comes from the correct source A common approach is source based addressing	Allowing to determine the sender/creator of a message
Availability	Not necessarily a direct safety goal since a non-available system can find a simple fail-safe state by going to no-operation	Mandating that data is on-hand when it is needed
Authorization	Implemented implicitly by allowing authenticated operation. Additionally, a check for maximum plausibility is sometimes applied, for example to check timing values.	Defining access rights
Confidentiality	--	Only authorized entities must be able to read confidential data
Non-Reputation	--	Evidence that the sender / creator of a message issued the message.

After having explained the main goals regarding safety and security, the risk assessment challenge is another important issue to tackle. Some work as the one done by [22] presents a way in which safety and security risks are separately addressed by means of their separate methods but as part of the Hazard Analysis and Risk Assessment.

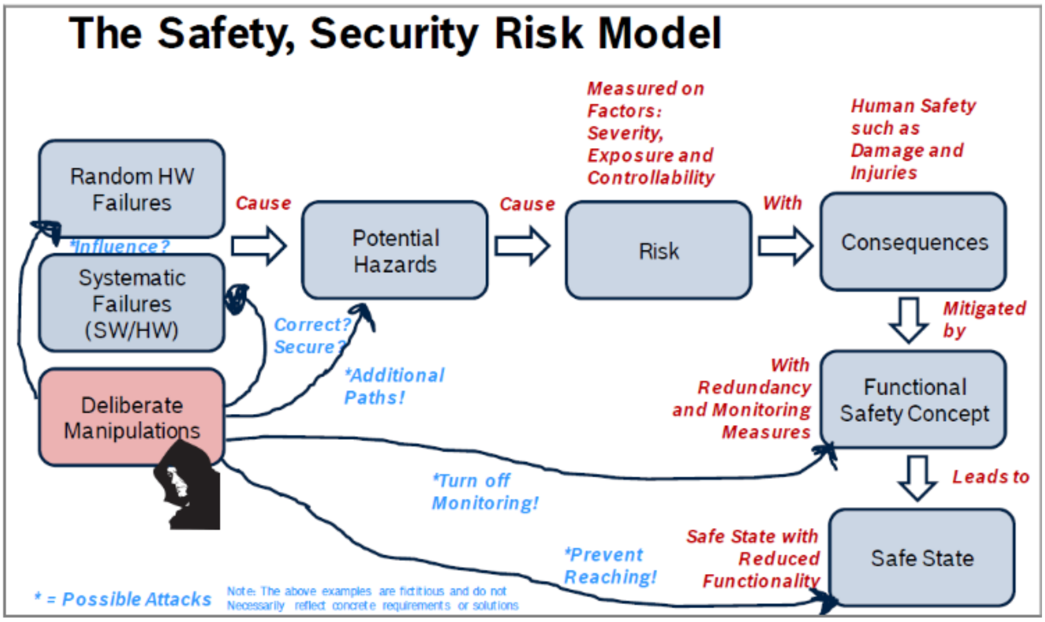
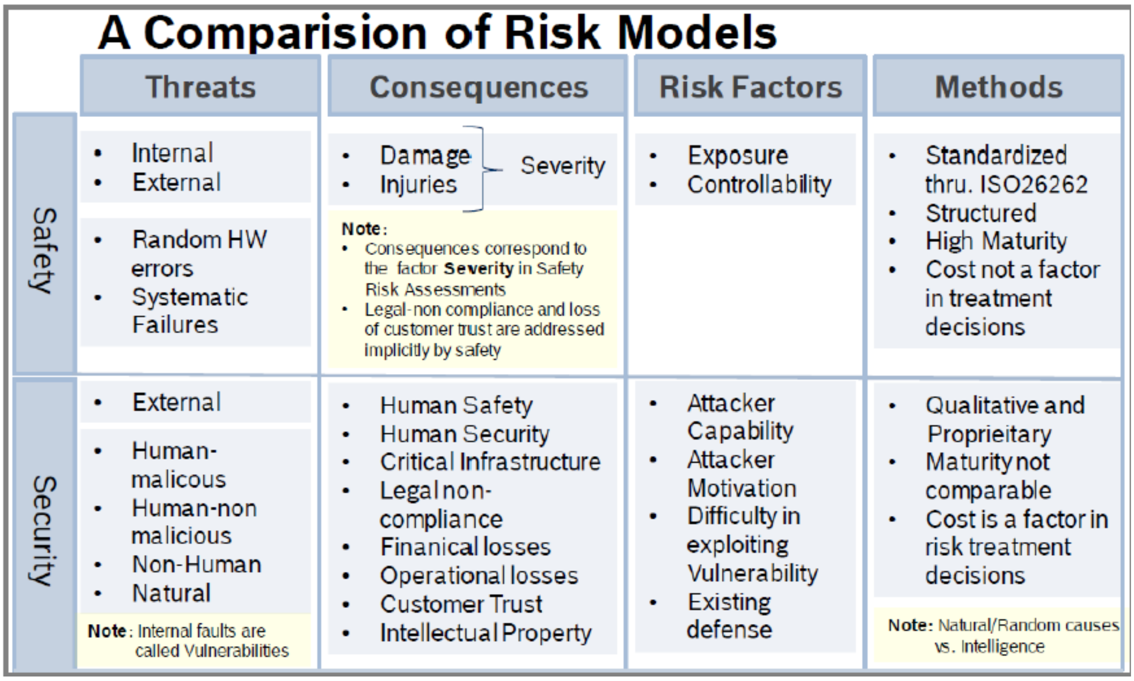

The Safety, Security Risk Model
Random HW Failures
*Influence?
Systematic Failures (SW/HW)
Deliberate Manipulations
Cause
Potential Hazards
Correct? Secure?
*Additional Paths!
Cause
Measured on Factors: Severity, Exposure and Controllability
Risk
With
Human Safety such as Damage and Injuries
Consequences
Mitigated by
With Redundancy and Monitoring Measures
Functional Safety Concept
Leads to
*Turn off Monitoring!
*Prevent Reaching!
Safe State with Reduced Functionality
Safe State
* = Possible Attacks
Note: The above examples are fictitious and do not Necessarily reflect concrete requirements or solutions

Figure 7: The Safety, Security Risk Model [15]

Whereas safety deals with risks arising because of natural random causes, security does it with threats inherit due to intentional causes.

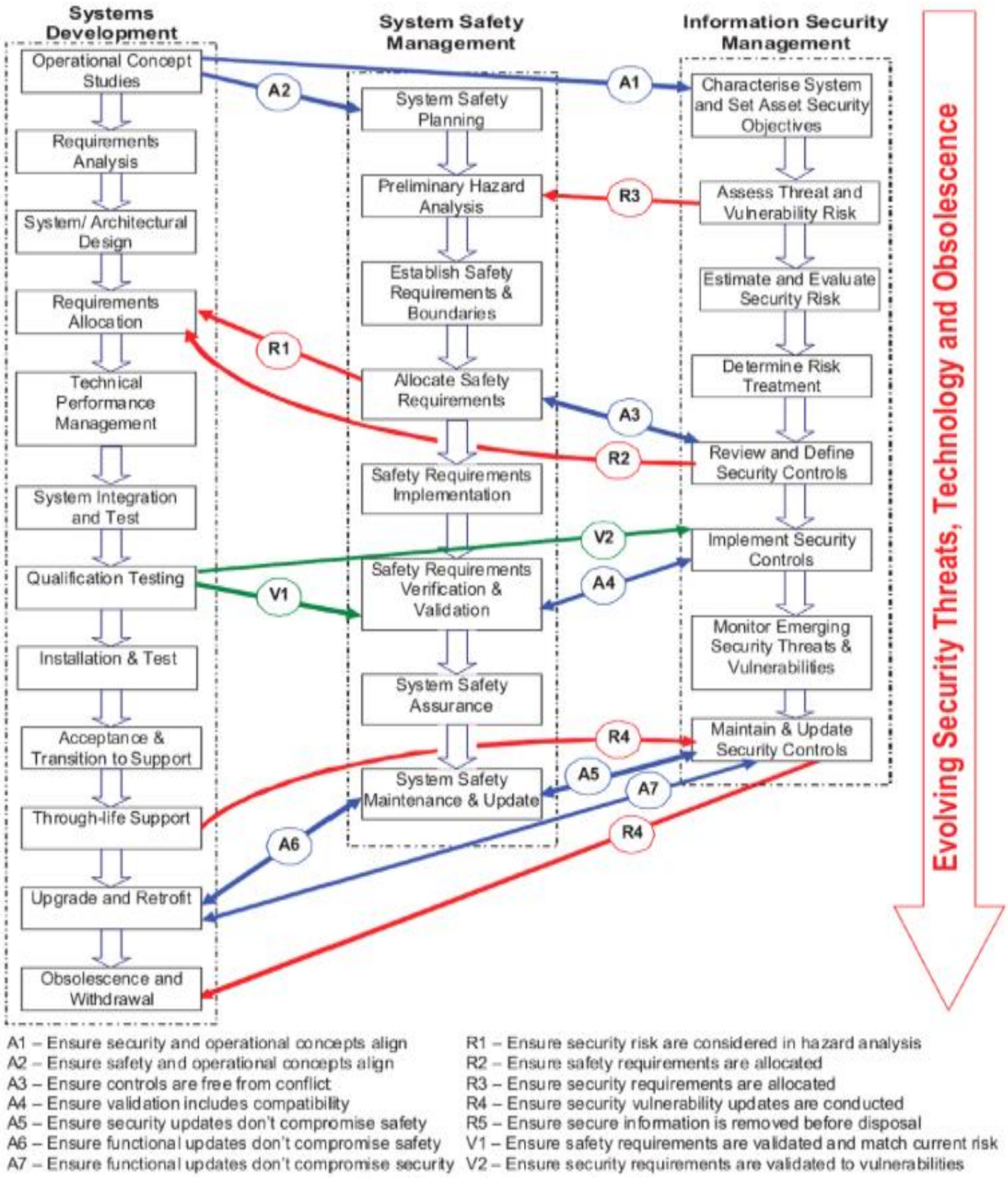


A Comparision of Risk Models

	Threats	Consequences	Risk Factors	Methods
Safety	• Internal • External • Random HW errors • Systematic Failures	• Damage • Injuries Severity **Note**: • Consequences correspond to the factor **Severity** in Safety Risk Assessments • Legal-non compliance and loss of customer trust are addressed implicitly by safety	• Exposure • Controllability	• Standardized thru. ISO26262 • Structured • High Maturity • Cost not a factor in treatment decisions
Security	• External • Human-malicous • Human-non malicious • Non-Human • Natural **Note**: Internal faults are called Vulnerabilities	• Human Safety • Human Security • Critical Infrastructure • Legal non-compliance • Finanical losses • Operational losses • Customer Trust • Intellectual Property	• Attacker Capability • Attacker Motivation • Difficulty in exploiting Vulnerability • Existing defense	• Qualitative and Proprieitary • Maturity not comparable • Cost is a factor in risk treatment decisions **Note:** Natural/Random causes vs. Intelligence

Figure 8: Comparison of the safety and security risk models [15]

It is important to remark that safety and security fields have been mostly treated as two different fields so far. Therefore, the need to understand how requirements and measures from one concern may impact the other one is of vital importance. To do so, two main approaches are considered: unification versus integration. Unification stands for a single methodology where the outcome is a single set of requirements describing safety and security. Conversely, the so-called integration or harmonization approaches investigate the similarities and differences of both concerns and tries to bring them into alignment by producing separate safety and security requirements. Once they are properly defined, the interaction between each other is shown in order to identify possible conflicts.


Systems Development
Operational Concept Studies
Requirements Analysis
System/ Architectural Design
Requirements Allocation
Technical Performance Management
System Integration and Test
Qualification Testing
Installation & Test
Acceptance & Transition to Support
Through-life Support
Upgrade and Retrofit
Obsolescence and Withdrawal
System Safety Management
System Safety Planning
Preliminary Hazard Analysis
Establish Safety Requirements & Boundaries
Allocate Safety Requirements
Safety Requirements Implementation
Safety Requirements Verification & Validation
System Safety Assurance
System Safety Maintenance & Update
Information Security Management
Characterise System and Set Asset Security Objectives
Assess Threat and Vulnerability Risk
Estimate and Evaluate Security Risk
Determine Risk Treatment
Review and Define Security Controls
Implement Security Controls
Monitor Emerging Security Threats & Vulnerabilities
Maintain & Update Security Controls
Evolving Security Threats, Technology and Obsolescence
A1
A2
R3
R1
A3
R2
V2
A4
V1
R4
A5
A7
A6
R4

A1 – Ensure security and operational concepts align
A2 – Ensure safety and operational concepts align
A3 – Ensure controls are free from conflict
A4 – Ensure validation includes compatibility
A5 – Ensure security updates don't compromise safety
A6 – Ensure functional updates don't compromise safety
A7 – Ensure functional updates don't compromise security

R1 – Ensure security risk are considered in hazard analysis
R2 – Ensure safety requirements are allocated
R3 – Ensure security requirements are allocated
R4 – Ensure security vulnerability updates are conducted
R5 – Ensure secure information is removed before disposal
V1 – Ensure safety requirements are validated and match current risk
V2 – Ensure security requirements are validated to vulnerabilities

Figure 9: Key alignment points between safety and security [17]

As already stated, both concerns share some similarities as well as differ in other aspects (see Table 2).

Table 2: Assessment similarities and differences between safety and security

Similarities			Differences		
Similar techniques to assess the impact of possible failures on the overall behaviour of a system			Basis for comparison	Safety	Security
Both forms of dependability			The intend behind	Hazards	Threats
Similar Fault Tolerance (FT) techniques	Safety FT Patterns/Mechanisms/Measures	Security Patterns/ Measures/Controls	Causes	Accidental	Deliberate
	HW, SW, temporal and informational redundancy	segmentation	Failing Criteria	Fail-silent/Fail-safe	Continuous operating/ Availability
Risk Assessment (e.g. Hazard Analysis and Risk Assessment/ Threats Analysis and Risk Assessment), Safety and Security Goals, Safety and Security Requirements, Functional Safety and Security Concepts (Fault Tolerant Architecture/ Intrusion Tolerant Architecture), Safety and Security Assurance			Assessing a security threat is different from assessing a safety hazard	Quantitative: SIL (Safety Integrity Level)	Qualitative: SL (Security Level)

In fact, safe systems need to be secured or in the other way around, if the they are not secure they are not safe. As depicted in Table 2, a safety analysis that does not consider hazards that could be caused by underlying security vulnerabilities is deficient. Novel methods such as FMVEA (Failure Modes, Vulnerabilities and Effects Analysis) [84] or extended fault trees need to be carried out.

Table 3 analyses several safety and security engineering tools and methodologies [18].

Table 3: Overview of safety and security tools and methods

Type	Safety-Oriented Approach	Adaptation to Security	Category (Means)
From Safety To Security			
Architectural Concepts	Fault-tolerant architectures	Intrusion-tolerant architectures	Fault Tolerance
		FRS technique; survivable networks	
		Diversity-based intrusion detection	
	Defense in depth	Defense in depth/security in depth	Fault Tolerance
Graphical Modelling	Fault Trees	Threat trees, attack trees	Fault Forecasting
	Dynamic Fault Trees	Dynamic attack trees	Fault Forecasting
	BDMP	BDMP for security	Fault Forecasting
Structured Risk Assessment	HAZOP	HAZOP for security	Fault Forecasting
		Vulnerability identification & Analysis HAZOPs	Fault Forecasting
	Sneak Circuit Analysis	Sneak path security Analysis	Fault Forecasting
	Zonal Analysis	Security Zonal Analysis	Fault Forecasting
	Safety Cases	Security Assurance Case	Other/Prevention
	FMEA	IMEA	Fault Forecasting
	GEMS	GEMS for security	Fault

			Prevention/Removal
	SIL (Safety Integrity Level)	SAL (Security Assurance Level) & SL (Security Level)	Fault Prevention/Removal
Testing	Fault Injection	Fault Injection , Fuzzing	Fault Removal/Forecasting
	Software reliability growth	Software security growth modelling	Fault Forecasting
From Security to Safety			
Architecture	Security Kernel	Safety Kernel	Fault Prevention/Tolerance
Graphical Modelling	Misuse case	Misuse case for safety	Fault Forecasting
	Misuse sequence diagram	Failure sequence diagram	Fault Forecasting
Formal Modelling	Non-interference property…Non-deducibility, causality	Safe behaviours formalization (fail-safe, fail-stop, etc.)	Fault Prevention
	Integrity-oriented access control models (e.g. Biba model)	Model with multiple levels of integrity (Totel´s Model)	Fault Prevention

In order to achieve the desired level of system dependability both safety and security must be ensured. Furthermore, a comprehensive integration of the functional safety and security analysis is very important and it is currently a challenging issue. This means that security must be balanced with the required level of functional safety. This way, security issues need also to be taken into account when preparing safety cases and engineers from both fields need to work together.

2.3 Concept of Security Assurance

Closely related to the aforementioned aspects, the concept of security assurance generally refers to the confidence in a system’s ability to mitigate and detect cyberattacks. The security assurance concept has multiple elements, including security controls, assurance processes, assurance techniques, the assessment of assurance level, and the generation of assurance evidence. The concept of security assurance is manifested in several inter-related terms, depending on the assurance target and the beneficiary of the assurance effort.

Information assurance is defined as the measure of confidence that the security features, practices, procedures, and architecture of an information system accurately mediates and enforces the security policy. Information assurance is a broader concept than security assurance, as it encompasses not only the protection of computing equipment against attacks but also management, personnel, training and law.

Software assurance is a term used commonly by software vendors to refer to the practice of reducing vulnerabilities, improving resistance to attack and protecting the integrity of their software products. The main focus of software assurance, from a vendor's point of view, is on the security, reliability, and quality of software products.

High-assurance systems refer to systems that are security-, safety-, and mission-critical. Although the denotation of high assurance to a system is subjective, it usually hints that the system requires dependability attributes in addition to security. High assurance systems require more rigors in analysis, verification, testing, and documentation, e.g. using assurance techniques such as formal verification or building upon verified architecture model and components.

Cyber-security assurance is a term used by regulators to force operators or asset owners to be compliant with assurance requirements or schemes. These assurance schemes specify policies and standards, baseline assurance processes and security controls according to identified risks, as well as the procedures for reporting. Cyber security assurance can be expressed in a checklist of specific security controls to which an asset owner must comply.

2.4 AMASS Challenges in Multi-Concern Assurance

In order to leverage the benefits of development methodologies, and progress beyond the state of the art, it is important to consider other aspects than safety as a part of the assurance framework. In the literature, “dependability” is often used as a general term to cover the various system characteristics that play a role in assurance.

The OPENCOSS project has developed an approach for mapping safety assurance artefacts, techniques and requirements across domains, using the OPENCOSS CCL to resolve the inconsistencies in terminology across the target domains and to support informed reuse of assurance assets. The SafeCer project has developed methodological guidelines (namely, Safety-oriented Process Line Engineering, extensible to Safety-oriented Process/Product/Safety Case Line Engineering) to enable cross-and intra-domain reuse via specification of commonality and variability. Also, the compositional certification approaches developed in OPENCOSS and SafeCer further support reuse by encapsulating assurance concerns for individual components into reusable assurance argument modules and by providing a mechanism to configure these modules to form an overall system assurance case.

In the AMASS project, we aim to exploit the existing OPENCOSS approach and extend it to provide a tool-supported methodology for the development of assurance cases, first tackling safety and security, with an extensible approach to address multiple dependability characteristics. We aim to extend the compositional certification approach to address multiple concerns. The OPENCOSS CCL metamodel is relatively generic, and its extension to support the reuse of assurance data relating to other dependability characteristics requires further domain modelling but no fundamental re-engineering of the approach. Similarly, the OPENCOSS vocabulary will require the addition of further concepts, but OPENCOSS techniques for using vocabulary to aid transfer and reuse of assets across domains are readily extensible. The ontology-based identification of commonalities and variability, explored within SafeCer, will be reconsidered to identify commonality between safety and security assessment processes. Reuse-based methodological solutions developed can SafeCer could also be re-used to enable the systematization of commonality and variability between safety and security processes. In AMASS, particular attention will be paid to security aspects of cyber-physical systems, where there are clear opportunities for the reuse of safety assets. As discussed below, there are some challenges to the extension of the OPENCOSS metamodels and technologies for multi-concern assurance, including for security.

There are three challenging aspects in AMASS for the development of assurance solutions with multiple system characteristics:

2.4.1 Dependability Assurance Modelling

Single-concern” safety cases are commonly used in the assurance and certification of systems in a variety of industrial sectors, most notably in defence, aerospace, the power industry, nuclear marine and medical devices. Reliability cases and maintainability cases are also used in some development contexts. The potential for developing security cases is also increasingly being realised. We must stress the importance of considering security throughout the system design and development, rather than using a post-hoc approach and identifying security flaws after the system is deployed.

The OPENCOSS CCL metamodel is relatively generic, and its extension to support the reuse of assurance data relating to other dependability-related attributes requires considerable further domain modelling, but no fundamental re-engineering of the approach. Similarly, the CCL vocabulary will require the addition of further concepts, but the vocabulary-based and model-based techniques for using mappings between concepts are readily transferable. From a methodological point of view, the SafeCer Safety-oriented Process Line Engineering and its initial vision-based extension of Security-informed Safety-oriented Process Line Engineering remain valid. However, their modelling means may require to be extended (though the AMASS CACM metamodel) to explicitly address additional dependability-related attributes.

In AMASS, we plan to define how to extend Safety Cases (as an artefact to provide a justified argument for assuring safety) with other relevant concerns such as Security, etc. The idea is to enrich the concept of "Assurance Case" with multi-concern aspects: dependencies, overlapping, contradictory arguments, etc. Here, the concerns could be complemented by other properties (availability, maintainability, etc.). The objective is to provide the mechanisms, means and guidelines to model dependencies, overlapping, contradictory goals/claims, etc.

2.4.2 Contract-Based Multi-concern Assurance

The various aspects of dependability coexist, sometimes in harmony with one another and sometimes in conflict, and there are complex dependencies and trade-offs between them. For example, there is a potential conflict between the safety of an aircraft and its availability. In order to fly the aircraft with all of its safety-related systems fully functional, there is a need to perform extensive, costly system maintenance, which means that the aircraft is regularly out of service. For the aircraft to fly without these systems, however, would adversely affect its safety. However, in some circumstances, for very limited periods of time, an aircraft is permitted to operate with less than its full complement of safety-related functionality: a reduction in the safety of the system is permitted (for example, by acceptance that the system can operate with reduced redundancy in its configuration) in order that the mission can be successfully completed. In order to provide assurance of the aircraft's suitability to carry out its intended function in its intended context, we need to record the relationships between the safety and availability aspects of the system, the decisions made during the development of the system to accommodate them and the effects of these decisions on safety, availability and any other concerns which they impact (in this case, maintainability, service retainability performance and potentially security).

OPENCOSS and SafeCer both looked at contract-based approach to compositional certification, which will need to be extended to accommodate this interplay between concerns. This is no small challenge, since many of the dependability aspects of interest are emergent properties of the system as a whole and cannot be addressed at component level alone. Similarly, the contracts developed previously sometimes rely specifically on explicit safety features of given components.

AMASS proposes to refine this approach to support the management of trade-offs between system characteristics. Here the goal is just to complement the work in WP3, to add further kind of assumptions/guarantees in contracts (security, robustness, etc.). The AMASS "contract-based" compositional solution should be versatile enough in order to support various kind of properties in assumptions/guarantees.

2.4.3 System Dependability Co-Analysis/Assessment

There have been several attempts to synergise safety and security as assurance qualities for mission-critical cyber-physical systems. Several models exist, which seek to demonstrate the extensibility of the "failure engineering" approach which underpins system safety assurance to a "threat engineering" approach for assuring security. The requirements of security and safety are not resolvable into a common high-level objective: these requirements are often not mutually compatible, and engineers very often need to trade off safety and security in the development of mission-critical systems. Nevertheless, the German VDE committee succeeded in proposing a somehow harmonized new edition of EN 50129 (railways) by integrating the lowest SL (security level) 1 of IEC 62443 (the well-established industrial network security standard) into the functional safety standard for railways. (Draft DIN VDE V 0831-104 (VDE V 0831-104), 2014).

With the increased networking capabilities, a number of serious challenges must be overcome before collaborative CPS can become a real business and social success instead of a promising vision. With this regard we perceive the challenges of functional safety particularly serious, since many of the typical application domains of collaborative embedded systems are inherently safety critical. Because of the openness and adaptability of such systems, we are faced with an enormous increase in complexity in Safety Engineering aspects, which cannot be dealt with the already established safety and quality assurance procedures. In addition, safety in collaborative embedded systems can no longer be considered in isolation from security, and it must be driven by the system architecture and architectural design.

3. State of the Art on Multi-Concern Assurance

In the context of AMASS, the state of the art on Multi-concern Assurance was divided into two main parts: 1) Dependability, 2) Safety and Security. This chapter introduces in first place the state of the Art on Dependability and in second place the state of the Art on Safety and Security.

3.1 Multiple Dependability Concerns

3.1.1 Co-design

The work presented in [5] begins by reviewing measures and existing techniques that are pertinent to dependability and security evaluation, showing how those techniques are currently applied in practice to the evaluation of certain security properties. While these applications suggest that there is merit to using stochastic techniques to evaluate security properties, they also suggest that significant new work is necessary to create a sound, model-based framework for quantifying system security.

At the highest level, the authors believe that this work falls into two categories:

1. Modelling attacker behaviour (cf. Figure 10);
2. Creating a single, comprehensive methodology for evaluating whether a design meets one or more high-level requirements related to security. The issues and challenges related to each of these needs are described.

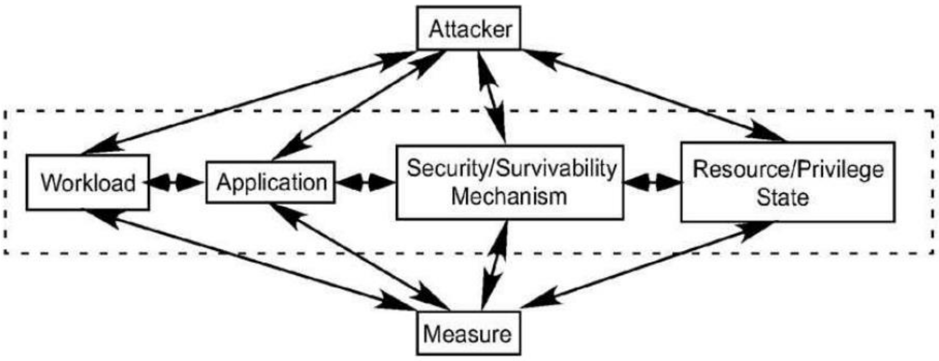


Figure 10: Probabilistic security model structure [5]

The authors conclude that stochastic evaluation techniques inspired by dependability evaluation methods have the potential to be used, with appropriate extension, for security evaluation. However, there are still significant obstacles to the creation of a comprehensive, integrated approach to the evaluation of multiple security properties, largely due to fundamental differences between the accidental nature of the faults and the intentional, human nature of cyber-attacks.

Safety-critical software (used in avionics, military or aerospace domains) must preserve their integrity, ensure a continuous operational state and enforce security of their data. There requirements are met through a dedicated development process that analyses and detects errors before system release. However, these methods are not sufficient and safety or security still occurs in such systems (e.g. explosion of Ariane 5, mission failure of Mars Climate Orbiter etc.). In addition, meeting safety and security requirements becomes more and more difficult due to an increasing number of functionalities. The work presented in [6] introduces a new method to build safety-critical systems and ensure their safety and security requirements. The approach proposes patterns for the specification of safe and secure systems. Then, a dedicated development process relies on them to (i) validate, (ii) automatically implement and (iii) certify the system, enforcing its requirements from the specifications to the code. System validation (i) detects specification errors, ensuring its correctness and feasibility prior any development effort. The automatic implementation process (ii) translates system specification into code and ensures their requirements enforcement. The certification (iii) aspect verifies that specification requirements are met in the implementation by analysing the system during its execution. It also evaluates its compliance against certification standards (such as DO178B).

The aim of the MAFTIA project [7] was to investigate the *tolerance paradigm* for security systematically, with the objective of proposing an integrated architecture built on this paradigm, and realising a concrete design that can be used to support the dependability applications. This project used fault tolerance techniques to build dependable systems that are tolerant to intrusion, and able to continue providing a secure service, in spite of the presence of malicious faults. MAFTIA's major innovation was a comprehensive approach for tolerating both accidental faults and malicious attacks in large-scale distributed systems, including attacks by external hackers and by corrupt insiders. It uniformly applied the tolerance paradigm to the dependability of complete large-scale applications in a hostile environment and not just to single components of such systems. There were three important focus points: (1) the architecture of MAFTIA: providing a framework that ensures the dependability of distributed applications in the face of a wide class of faults and attacks; (2) the design of dependability mechanisms and protocols; (3) the verification and assessment of the work.

3.1.2 Co-analysis

The work presented in [3] gives the main definitions relating to dependability, a generic concept including such attributes as reliability, availability, safety, integrity, maintainability, etc. Security brings in concerns for confidentiality, in addition to availability and integrity. Basic definitions are given first. The author highlights the close interactions between fault removal and fault forecasting, and motivates their gathering into dependability and security analysis. The presented approach aimed at reaching confidence in the ability to deliver a service that can be trusted, whereas the grouping of fault prevention and fault tolerance constitutes dependability and security provision, aimed at providing the ability to deliver a service that can be trusted. Another grouping of the means is the association of 1) fault prevention and fault removal into fault avoidance, i.e., how to aim for fault-free systems, and of 2) fault tolerance and fault forecasting into fault acceptance, i.e., how to live with systems that are subject to faults. Figure 11 illustrates the groupings of the means for dependability.

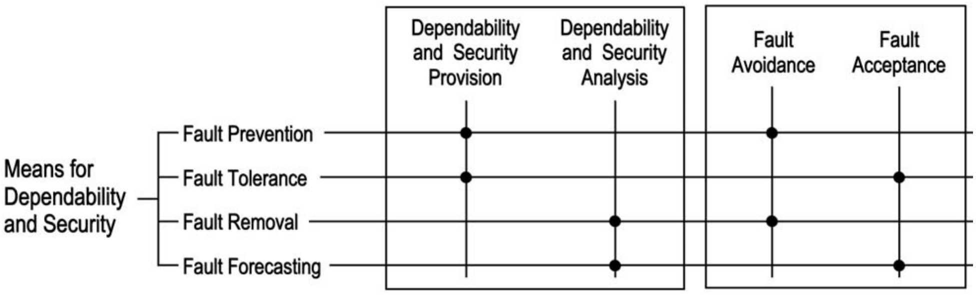

Dependability and Security Provision
Dependability and Security Analysis
Fault Avoidance
Fault Acceptance
Means for Dependability and Security
Fault Prevention
Fault Tolerance
Fault Removal
Fault Forecasting

Figure 11: Groupings of the means for dependability and security

In the work exposed in [8] the author presents a new approach to integrated security and dependability evaluation, which is based on stochastic modelling techniques. The proposal aims to provide operational measures of the trustworthiness of a system, regardless if the underlying failure cause is intentional or not. By viewing system states as elements in a stochastic game, the authors can compute the probabilities of expected attacker behaviour, and thereby be able to model attacks as transitions between system states Figure 12. The proposed game model is based on a reward-and-cost concept. A section of the paper is devoted to the demonstration of how the expected attacker behaviour is affected by the parameters of the game. This model opens up for the use of traditional Markov analysis to make new types of probabilistic predictions for a system, such as its expected time to security failure.

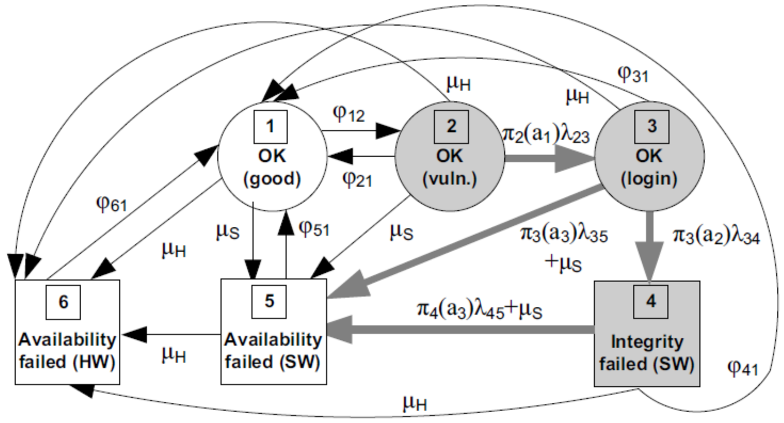


Figure 12: State transition model of DNS server with game elements identified

A first work was enhanced in [9] and gives results of a web services dependability analysis using standardized FMEA (Failure Modes and Effects Analysis) technique and its proposed modification IMEA (Intrusion Modes and Effects Analysis) technique. Obtained results of the FMEA-technique application were used for determining the necessary means of error recovery, fault prevention, fault-tolerance ensuring and fault removal. Systematization and analysis of web service intrusions and means of intrusion-tolerance were fulfilled by use of IMEA-technique. The author also proposes the architectures of the fault and intrusion-tolerant web services based on the components diversity and dynamical reconfiguration, and discuss principles and results of dependable and secure web services development and deployment by use of the F(I)MEA-technique and multi-version approach. Then following the same approach, the work presented in [10] presents results of a SCADA-based ICS dependability and security analysis using a modification of standardized FMEA technique. The technique mentioned takes into account possible intrusions and is called F(I)MEA (Failure (Intrusion) Modes and Effects Analysis). The F(I)MEA technique is applied for determining the weakest parts of ICS and the required means of fault prevention, fault detection and fault-tolerance ensuring. An example of F(I)MEA-technique applying for SCADA vulnerabilities analysis is provided. The solutions of SCADA-based ICS dependability improvement are proposed.

3.1.3 Co-Verification and Validation

The work presented in [3] introduces the means for the achievement of dependability and security:

- **Fault Removal During Development**

Fault removal during the development phase of a system lifecycle consists of three steps: verification, diagnosis, and correction. The focus is in what follows on verification that is the process of checking whether the system adheres to given properties, termed the verification conditions. If it does not, the other two steps have to be undertaken: diagnosing the fault(s) that prevented the verification conditions from being fulfilled, and then performing the necessary corrections. After correction, the verification process should be repeated in order to check that fault removal had no undesired consequences; the verification performed at this stage is usually termed non-regression verification.

Checking the specification is usually referred to as validation. Uncovering specification faults can happen at any stage of the development, either during the specification phase itself, or during subsequent phases when evidence is found that the system will not implement its function, or that the implementation cannot be achieved in a cost-effective way.

Verification techniques can be classified according to whether or not they involve exercising the system (see Figure 13).

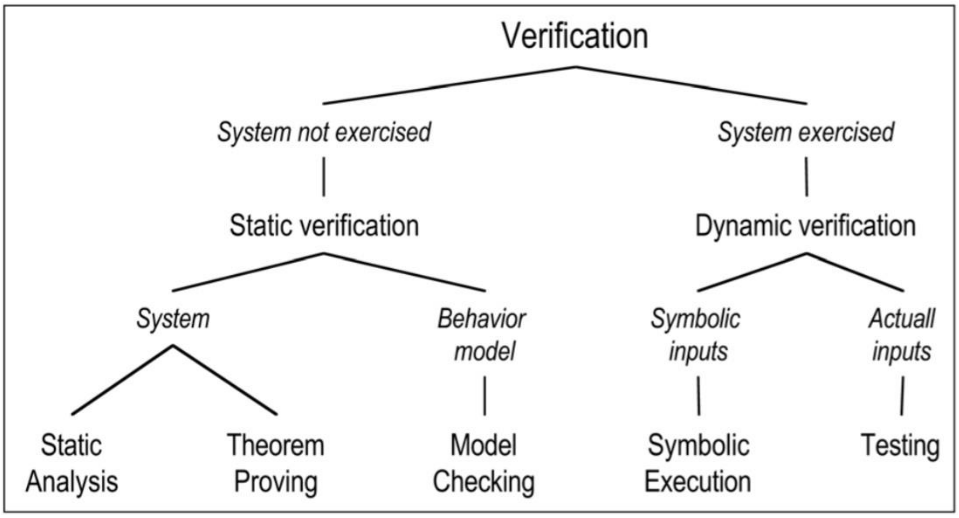


Figure 13: Verification approaches

Verifying a system without actual execution is static verification.

Such verification can be conducted:

- on the system itself, in the form of
  - (1) static analysis (e.g., inspections or walk-through, data flow analysis, complexity analysis, abstract interpretation, compiler checks, vulnerability search, etc.) or
  - (2) theorem proving;
- on a model of the system behaviour, where the model is usually a state-transition model (Petri nets, finite or infinite state automata), leading to model checking.

Verifying a system through exercising it constitutes dynamic verification; the inputs supplied to the system can be either symbolic in the case of symbolic execution, or actual in the case of verification testing, usually simply termed testing.

❖ Fault Removal during use

Fault removal during the use of a system is corrective or preventive maintenance. Corrective maintenance aims to remove faults that have produced one or more errors and have been reported, while preventive maintenance is aimed at uncovering and removing faults before they might cause errors during normal operation. The latter faults include (1) physical faults that have occurred since the last preventive maintenance actions, and (2) development faults that have led to errors in other similar systems. Corrective maintenance for development faults is usually performed in stages: The fault may be first isolated (e.g., by a workaround or a patch) before the actual removal is completed. These forms of maintenance apply to non-fault-tolerant systems as well as to fault-tolerant systems, that can be maintainable online (without interrupting service delivery) or offline (during service outage).

❖ Fault Forecasting

Fault forecasting is conducted by performing an evaluation of the system behaviour with respect to fault occurrence or activation. Evaluation has two aspects:

- qualitative, or ordinal, evaluation, that aims to identify, classify, and rank the failure modes, or the event combinations (component failures or environmental conditions) that would lead to system failures;

- quantitative, or probabilistic, evaluation, that aims to evaluate in terms of probabilities the extent to which some of the attributes are satisfied; those attributes are then viewed as measures.

The methods for qualitative and quantitative evaluation are either specific (e.g., failure mode and effect analysis for qualitative evaluation, or Markov chains and stochastic Petri nets for quantitative evaluation), or they can be used to perform both forms of evaluation (e.g., reliability block diagrams, fault-trees).

The two main approaches to probabilistic fault-forecasting (aimed to derive probabilistic estimates) are modelling and (evaluation) testing. These approaches are complementary since modelling needs data on the basic processes modelled (failure process, maintenance process, system activation process, etc.), that may be obtained either by testing, or by the processing of failure data.

3.1.4 Assurance

Traditionally, assurance has been separated into individual safety, security or reliability cases. Safety cases have been particularly used in railway, aerospace, power industry, defence, nuclear or medical devices, however, cases can be found less often in some development contexts regarding reliability or maintainability cases. Maintainability cases are for instance a requirement of the U.K. defence Standard DS-00-40 [11]. Furthermore, security cases are another remarkable example of concern specific cases. Thus, the main goal is to extend that concept from individual concern cases to assurance case and to overcome the possible challenges that can emerge regarding different aspects of dependability coexistence.

An assurance case consists of documented body of evidence that provides a convincing and valid argument that a specified set of critical claims regarding a system´s properties are adequately justified for a given application in a given environment.

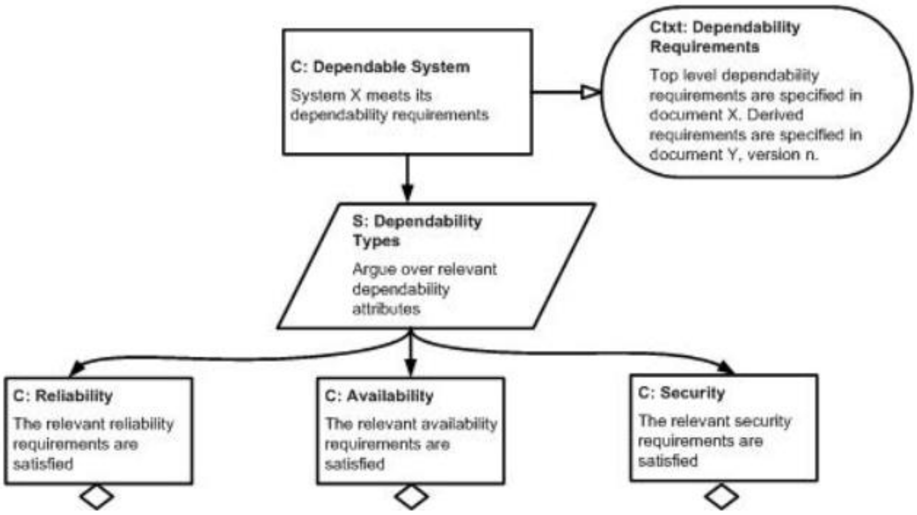

C: Dependable System
System X meets its dependability requirements
Ctxt: Dependability Requirements
Top level dependability requirements are specified in document X. Derived requirements are specified in document Y, version n.
S: Dependability Types
Argue over relevant dependability attributes
C: Reliability
The relevant reliability requirements are satisfied
C: Availability
The relevant availability requirements are satisfied
C: Security
The relevant security requirements are satisfied

Figure 14: An Assurance Case Fragment

Table 4 shows some of the most common objectives, arguments and evidences for safety, reliability, maintainability, availability, and security concerns.

Table 4: Objectives, Arguments and Evidence for the Dependability Attributes

Attribute	Objective	Typical Argument	Typical Evidence
Safety	System is adequately safe	Hazard mitigation argument	Hazard Analysis
Reliability	System meets reliability requirements	Enough redundancy, resilient components	Testing, simulation, analysis (e.g. FTA, FMEA, Markov)

Maintainability	System meets maintainability requirements	Modular cohesive design, plug and play devices, ease of replacing components	Simulation, expert opinion
Availability	Ability of a system to be kept in a functioning state	Availability of a system depends on the system's design reliability, its maintain-ability, and its maintenance support.	Analysis, expert judgement, simulation, testing
Security	Mission critical information is adequately protected	Assets protection argumentation	Access control, policies

Figure 15 illustrates the argumentation that a system is acceptably dependable based in GSN.

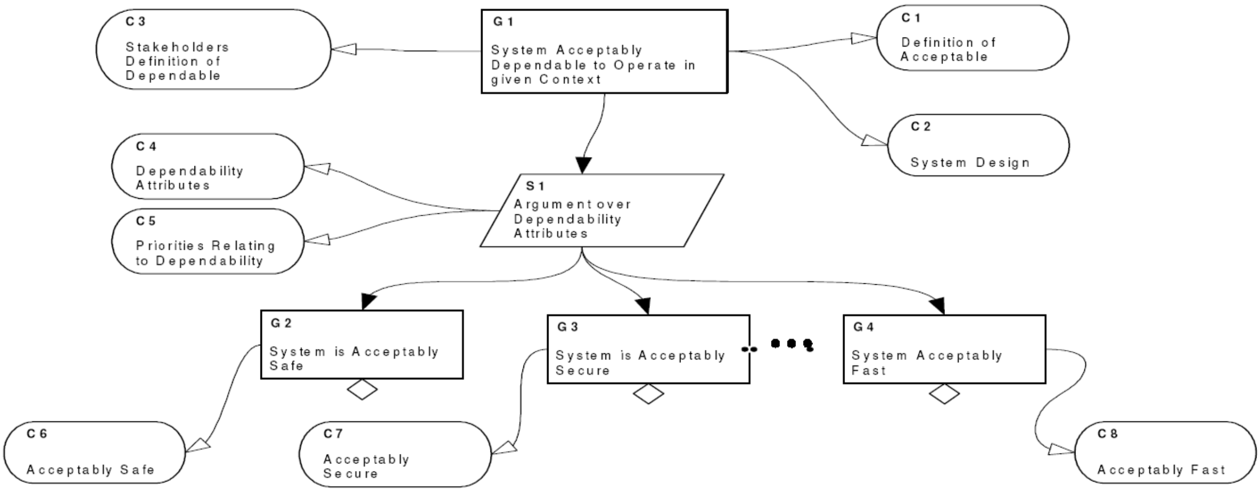

C3
Stakeholders Definition of Dependable
G1
System Acceptably Dependable to Operate in given Context
C1
Definition of Acceptable
C2
System Design
C4
Dependability Attributes
C5
Priorities Relating to Dependability
S1
Argument over Dependability Attributes
G2
System is Acceptably Safe
G3
System is Acceptably Secure
G4
System Acceptably Fast
C6
Acceptably Safe
C7
Acceptably Secure
C8
Acceptably Fast

Figure 15: A Possible Dependability Goal Structure

Assurance the dependability of a system requires the construction and evaluation of a dependability assurance case with explicit claims about system behaviour; supporting evidence for claims, arguments linking evidences to the claims and it is evaluated by independent assessors. Its main goal is to present an argument that a system is acceptably safe, secure, reliable etc. in a given context. This justification strategy can basically be divided in three main approaches [12]: goal-based approach, risk informed approach and rule-based approach. The justification of the goal-based approach demonstrates that the desired behaviour, property or reliability level has been achieved. Since it is necessary to demonstrate that the system will not have unacceptable behaviour, the second approach justifies that all the hazards have been addressed and that the risk coming from possible vulnerabilities have been reduced to an acceptable level. The rule-based approach relies on a well-structured development process and standards compliance to justify that the system has been properly designed and verified.

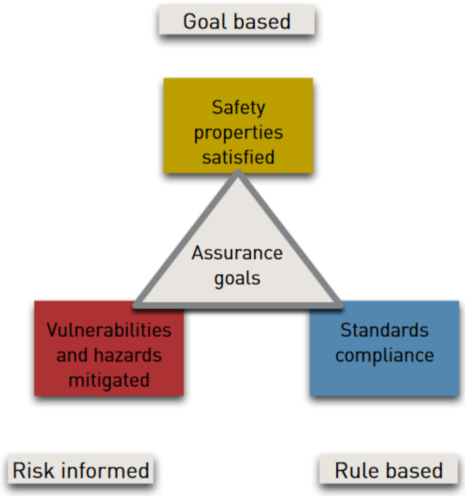


Figure 16: Justification strategies for assurance

According to Despotou [67] , seven remarkable challenges and different requirements belonging to each of them arise when referring to dependability cases:

Challenge	Dependability Case Requirement	Thesis Contribution
Multiple dependability attributes	It should record the relationships between the concerns and how they are affected by means of design decisions	Dependability Deviation Analysis (DDA) method to identify interaction of dependability attributes (concerns) (See Figure 19)
	It should be able to capture the conflicts and the trade-offs made during the development process of a system, as a way of justifying why those decisions were adopted	DDA: introduces the concept of failure maps (understand how one concern influences another)
		DDA: defines a dependability profile (capture and collate the requirements of System of systems (SoS) elements; structure the architecture of the dependability case)
Allocation and apportionment of requirements	The dependability case should be able to demonstrate assurance regarding the contribution of the underlying behaviour of SoS elements, in satisfying the overall dependability requirements of the SoS stakeholders	DDA to allocate and apportion the requirements, starting from the system stakeholders down to the allocation to system elements
Conflicting requirements	As dependability attributes can be in conflict with each other, stakeholders need to re-evaluate and trade their initial requirements identifying the most optimal solution to result in an acceptable system	Trade-Off Method (TOM) (See Figure 19)
Changing requirements	The dependability case should evolve at the same time that the system design does	Out of scope of this thesis
	The dependability case should be able to provide justification for the elicitation of	

	the dependability requirements	
	The dependability case should be able to "inform" the reconfiguration process of SoS, whilst maintaining acceptable assurance about the satisfaction of the overall dependability requirements	
Traceability	The dependability case of a system can be used to record the requirements and their subsequent decomposition and apportionment to SoS elements	Dependability Case Metamodel: document the associations between the contributions done by the thesis
	Definition of a rigorous dependability case framework requires well articulated relations hips between the concepts used during the evolution of a dependability case	Full traceable dependability cases by instantiating the metamodel and possibility of automating certain aspects of the dependability case evolution
Interaction of Case and Design	The argument should be created in parallel with the system development in order to help to evaluate, record and justify decisions regarding the evolution of the system	FANDA (Factors, Analysis and Decisions Alternatives) (See Figure 19)
Ownership of the dependability case	A framework should be defined under which the contribution of each of the contractors to the dependability case can be clearly and traceably identified	Out of scope of this thesis

In this work, a set of methods (see Figure 19) and different metamodels such as the dependability case metamodel were constructed. However, neither IV challenge nor VII were in the scope of the research. Furthermore, assurance levels and the presence of change in dependability cases were not considered.

As part of the dissertation process of the aforementioned PhD work, several further works were published. In [68] an approach that "fits" within the dependability case framework for identification and balancing of dependability objectives was presented. Since dependability is a combined term consisting of several attributes, they can be either in conflict or in harmony. Consequently, there might be the need to make and appropriately justify the trade-offs made in the design-phase. Besides, a modular approach to structure dependability arguments is described. This modularity feature is considered part of the capabilities of Goal Structuring Notation (GSN).

Modularity has been introduced into dependability cases in order to achieve the challenge of complexity and length of cases. Once a modular dependability case is created, a module can be referenced in other parts of the dependability case such as the different concerns or attributes associated with dependability or be reused along with the system associated.

Despotou highlighted the following advantages of using modular safety cases approach [80]:

- Reuse of arguments
- Containment of impact of change
- Contracts between argument modules provide additional barrier to propagation of change if the public (cross-referenced) goals from a supporting argument have changed.
- Limiting the cost of (re)generating evidence
- Integration of process and product arguments
- Standardisation of processes

He also addressed two challenges: (1) Complexity, in relation that reference between modules could increase complexity and reduce clarity. In this sense, coupling between arguments should be maintained at reasonable levels. (2) Loss of uniformity, and over-specification, in relation to the level of abstraction of the modules. Argument contracts should make clear the relationship of the referenced arguments.

An argument contract is seen as the mechanism to record the interdependencies existing between the argumentation modules that form the dependability case. These contracts are used to show how the claims from one module are supported by arguments from another module. The IAWG (Industrial Avionics Working Group) [81] proposed to take advantage of the GSN graphical notation within the argument contract as it provides more expressiveness and clarity than the tabular approach and also be integrated with the dependability argument. They also proposed a generic pattern for safety case contract modules (See Figure 17).

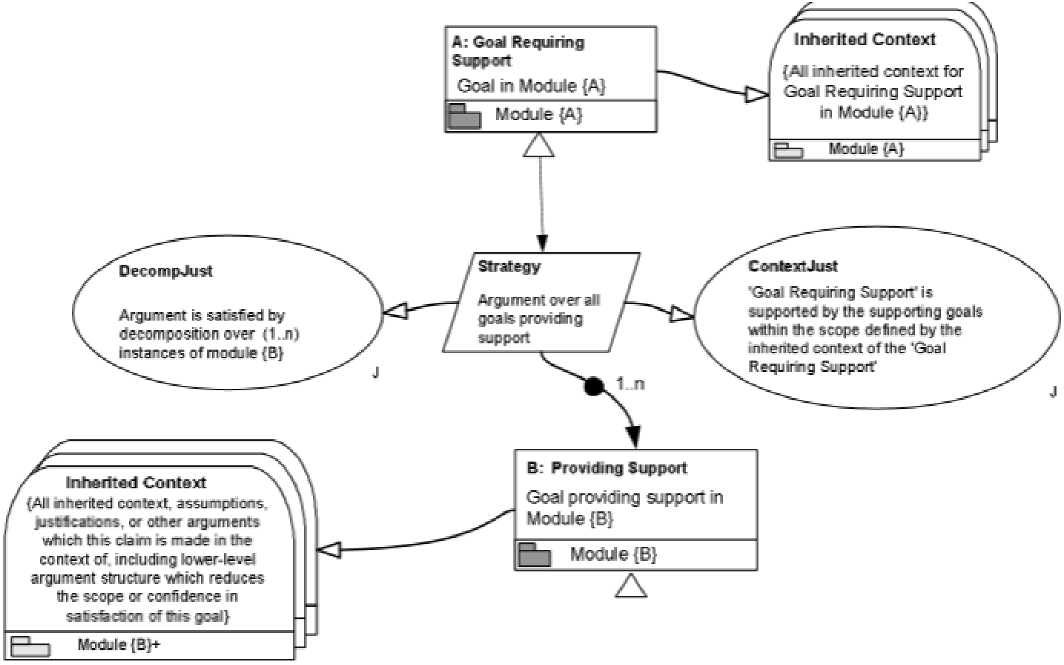

A: Goal Requiring Support
Goal in Module {A}
Module {A}
Inherited Context
{All inherited context for Goal Requiring Support in Module {A}}
Module {A}
DecompJust
Argument is satisfied by decomposition over (1..n) instances of module {B}
J
Strategy
Argument over all goals providing support
ContextJust
'Goal Requiring Support' is supported by the supporting goals within the scope defined by the inherited context of the 'Goal Requiring Support'
J
1..n
Inherited Context
{All inherited context, assumptions, justifications, or other arguments which this claim is made in the context of, including lower-level argument structure which reduces the scope or confidence in satisfaction of this goal}
Module {B}+
B: Providing Support
Goal providing support in Module {B}
Module {B}

Figure 17: Generic pattern for safety case contract modules

Ruiz in [82] went further with the formalization of the assurance contract. In this way, she defined a list of structured expressions to be used within the assertions. Assertion tend be done in natural language which introduces ambiguity and possible incompleteness. Her proposal included a categorised list of controlled expressions with the intention to reduce possible human factor error due to distraction or fatigue. The used of controlled expression reduces the learning curve and the barrier of introducing a formal language on the industry while improving the formalization and possibilities to include tool support for validation and verification.

In Figure 18 individual arguments for each of the dependability attributes are illustrated which encapsulate the convenient types of argument and evidence. On the top of all of them, the “top level dependability arguments” justifies that the dependability aims obtained are appropriate for the system’s use. On the other hand, in order to substantiate the acceptance of all attributes in the context of each other, the trade-off argument is shown at the bottom. Hence, Figure 18 depicts how the system is acceptably safe, secure and fast at the same time. To provide a context in which these trade-offs can be discussed, GSN was evolved by including “target” (the target requirement the system should satisfy such as the dependability objective) and “limit” (“the minimum condition that must be met so that the system will acceptably satisfy the objectives of the anticipated operation/scenarios symbols”).

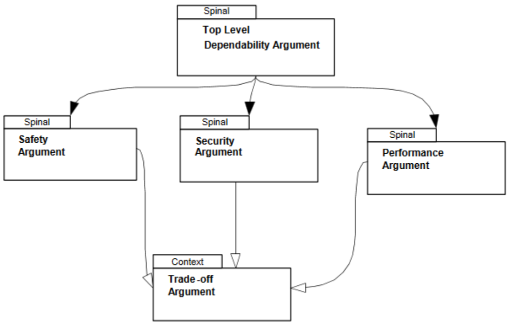


Figure 18: Modular View of Dependability

The authors presented the use of GSN contracts to architect a dependability case throughout the lifecycle of a system and capture the associations between the argument modules, which are self-contained arguments, of the case. These arguments can be characterized as either process (arguments regarding the development process) or product arguments (arguments regarding the system as artefact). Furthermore, at the same time that the design evolves the dependability case should do it likewise.

In fact, the dependability case can comprise the confluence between evolution of requirements and design, since it provides a way of documenting design trade-offs and decisions. This consists of analysing the system and, eliciting goals (requirements), identifying design alternatives, and resolving conflicts by making trade-offs. To support this co-evolution they presented a set of methods which can be seen in Figure 19.

Together with the previous achievement, the dependability profile was introduced in this work.

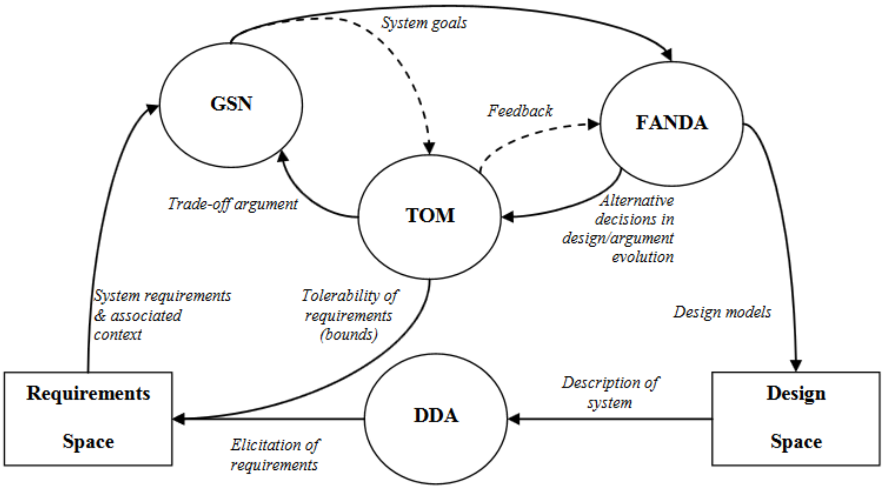

System goals
GSN
Feedback
FANDA
Trade-off argument
TOM
Alternative decisions in design/argument evolution
System requirements & associated context
Tolerability of requirements (bounds)
Design models
Requirements Space
DDA
Description of system
Design Space
Elicitation of requirements

Figure 19: Methods to support the co-evolution of dependability cases

The main aims of DDA (Dependability Deviation Analysis) are (i) to evoke the required and acceptable behaviour of the system w.r.t. dependability and (ii) to show in a graphical way how the no satisfying of a certain aspect of dependability attribute can affect the system´s behaviour. FANDA complements GSN in supporting the development of arguments so that a number of “meta” arguments targeting how each design alternative satisfy the different dependability concerns. In other words, since this method captures how features of proposed decision alternatives affect the goals of system, it facilitates the evolution of the case and system. To finish with, TOM (Trade-off Method) constitutes a method to establish bounds of acceptability by creating arguments of why a certain decision was done during the development of the system.

By following the same line, [85]presented a GSN argument pattern for making multi-attribute trade-offs:

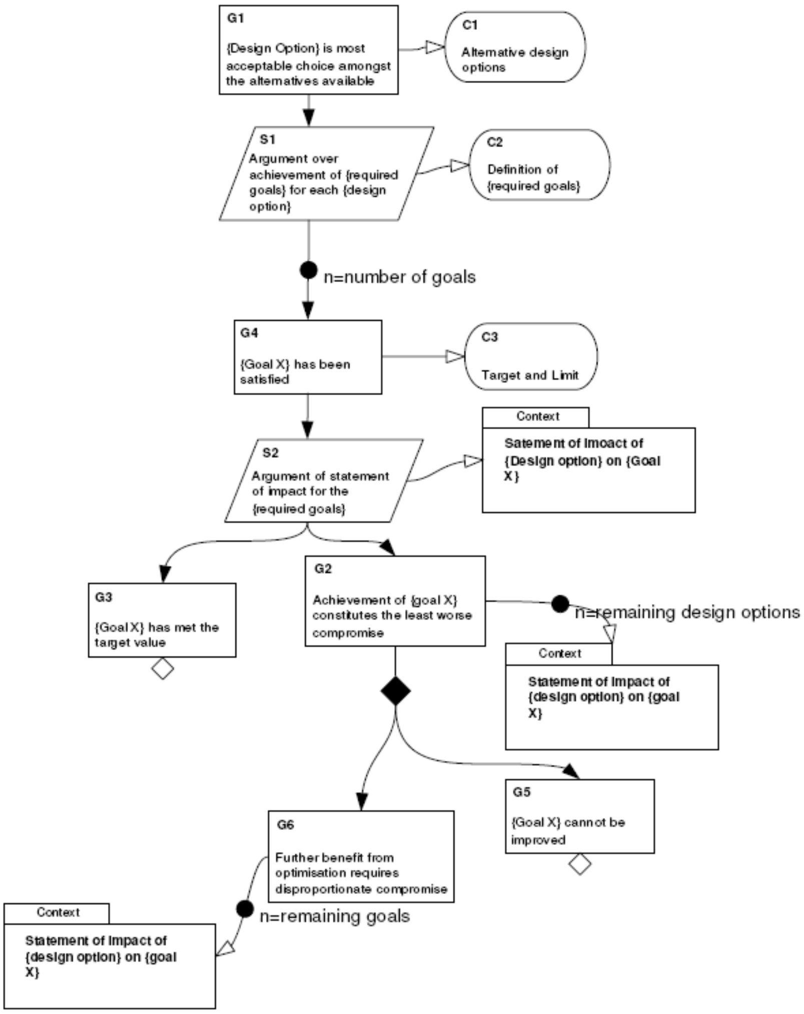

G1
{Design Option} is most acceptable choice amongst the alternatives available
C1
Alternative design options
S1
Argument over achievement of {required goals} for each {design option}
C2
Definition of {required goals}
n=number of goals
G4
{Goal X} has been satisfied
C3
Target and Limit
Context
Satement of Imoact of {Design option} on {Goal X}
S2
Argument of statement of impact for the {required goals}
G3
{Goal X} has met the target value
G2
Achievement of {goal X} constitutes the least worse compromise
n=remaining design options
Context
Statement of Impact of {design option} on {goal X}
G5
{Goal X} cannot be improved
G6
Further benefit from optimisation requires disproportionate compromise
n=remaining goals
Context
Statement of Impact of {design option} on {goal X}

Figure 20: GSN Argument Pattern for making multi-concern trade-offs

An important aspect to highlight and which constitutes one of the AMASS goals, is the one related to the use of contracts to structure the dependability case. This aspect was also targeted by the previous research work. Thanks to the use of modular argumentation [OPENCOSS], diverse benefits arise: high cohesion, low coupling (associations defined at module level and not at goal level), defined and documented interfaces and information hiding. To show how contracts are used to structure a case Figure 21 is presented and it is worth noting that contracts represent themselves not only as interfaces between modules but as arguments to argue about how the parent goals is decomposed.

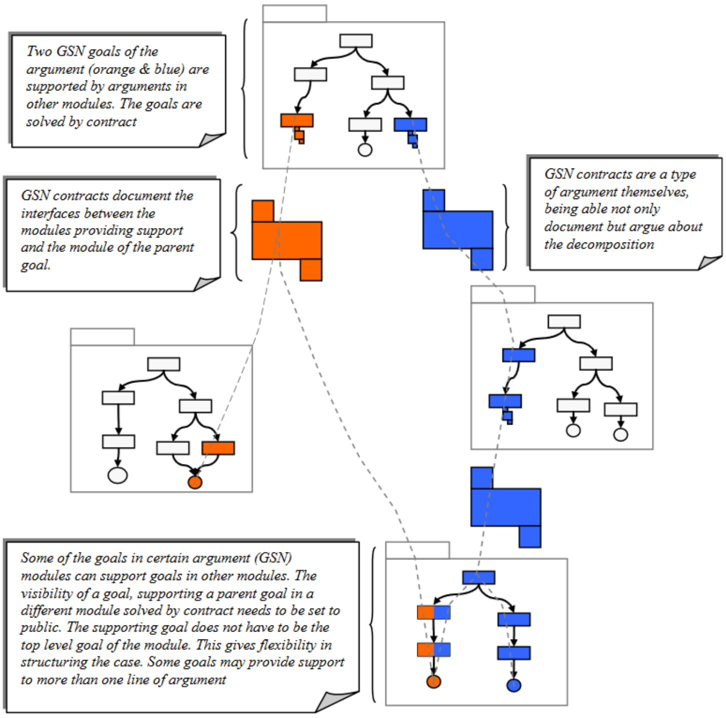

Two GSN goals of the argument (orange & blue) are supported by arguments in other modules. The goals are solved by contract
GSN contracts document the interfaces between the modules providing support and the module of the parent goal.
GSN contracts are a type of argument themselves, being able not only document but argue about the decomposition
Some of the goals in certain argument (GSN) modules can support goals in other modules. The visibility of a goal, supporting a parent goal in a different module solved by contract needs to be set to public. The supporting goal does not have to be the top level goal of the module. This gives flexibility in structuring the case. Some goals may provide support to more than one line of argument

Figure 21: Use of GSN contracts in structuring a case

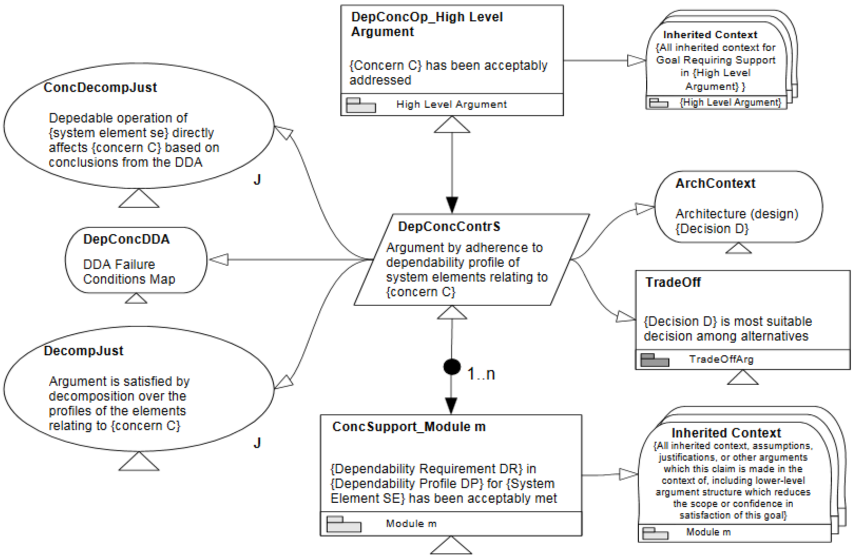

DepConcOp_High Level Argument
{Concern C} has been acceptably addressed
High Level Argument
Inherited Context
{All inherited context for Goal Requiring Support in {High Level Argument} }
{High Level Argument}
ConcDecompJust
Depedable operation of {system element se} directly affects {concern C} based on conclusions from the DDA
J
DepConcDDA
DDA Failure Conditions Map
DepConcContrS
Argument by adherence to dependability profile of system elements relating to {concern C}
ArchContext
Architecture (design) {Decision D}
TradeOff
{Decision D} is most suitable decision among alternatives
TradeOffArg
DecompJust
Argument is satisfied by decomposition over the profiles of the elements relating to {concern C}
J
1..n
ConcSupport_Module m
{Dependability Requirement DR} in {Dependability Profile DP} for {System Element SE} has been acceptably met
Module m
Inherited Context
{All inherited context, assumptions, justifications, or other arguments which this claim is made in the context of, including lower-level argument structure which reduces the scope or confidence in satisfaction of this goal}
Module m

Figure 22: Dependability case contract supporting the high-level dependability argument

Briefly speaking, GSN contracts are used as a way of integrating different arguments of which the dependability case is built upon.

The SafSec project was funded by the MoD Defence Procurement Agency who wished to reduce the cost and effort of safety certification and security accreditation for future military avionics systems and in-service upgrades. The SafSec Standard [13] helps to achieve the certifications with the minimum of duplicated work and the maximum of reuse of evidence between the different certifiers. SafSec focuses on dependable systems, therefore adding reliability and maintainability attributes to the baseline safety and security attributes.

The SafSec Standard establishes a well-defined terminology, and defines the top-level goal "*The system is demonstrably dependable"* as the goal to be achieved to be compliant with the standard. The top-level goal is decomposed using the GSN into 17 requirements directed towards the demonstration of dependability (cf. Figure 23). The associated Guidance expands on the objectives set out in SafSec with indications of how the objectives may be met while conforming to existing safety and security standards.

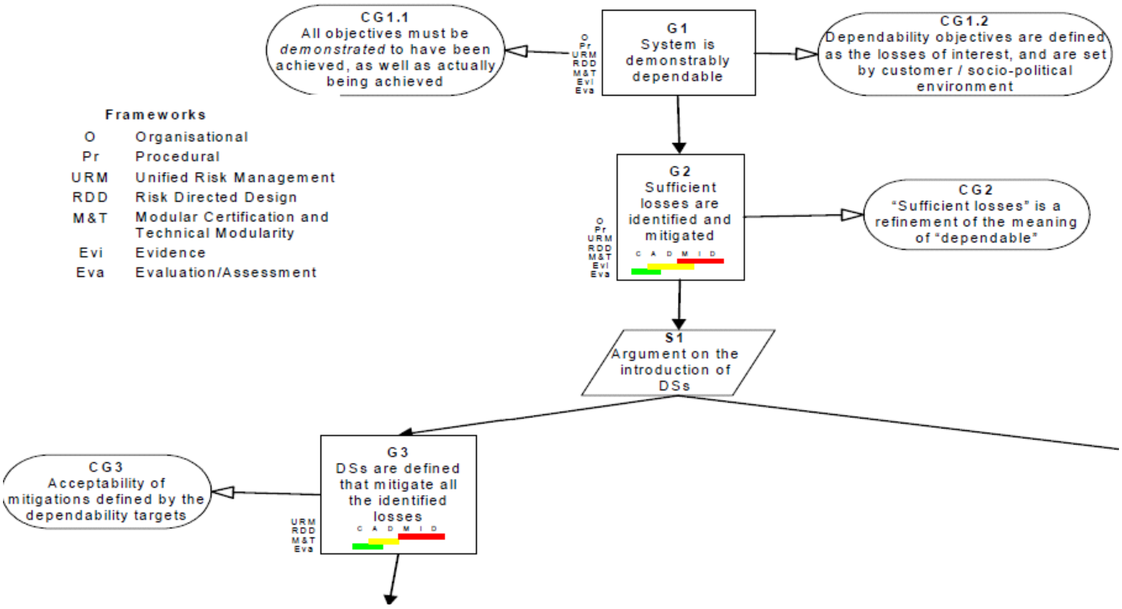

CG1.1
All objectives must be *demonstrated* to have been achieved, as well as actually being achieved
O Pr URM RDD M&T Evi Eva
G1
System is demonstrably dependable
CG1.2
Dependability objectives are defined as the losses of interest, and are set by customer / socio-political environment
Frameworks
O Organisational
Pr Procedural
URM Unified Risk Management
RDD Risk Directed Design
M&T Modular Certification and Technical Modularity
Evi Evidence
Eva Evaluation/Assessment
G2
Sufficient losses are identified and mitigated
C A D M I D
CG2
"Sufficient losses" is a refinement of the meaning of "dependable"
S1
Argument on the introduction of DSs
CG3
Acceptability of mitigations defined by the dependability targets
G3
DSs are defined that mitigate all the identified losses
URM RDD M&T Eva

Figure 23: Extract of goal structure for the achievement of a dependable system

Jackson et al. [14] presents the SafSec Standard (cf. Figure 24), and the safety and security regulatory shifts, which occurred in parallel to the elaboration of SafSec.

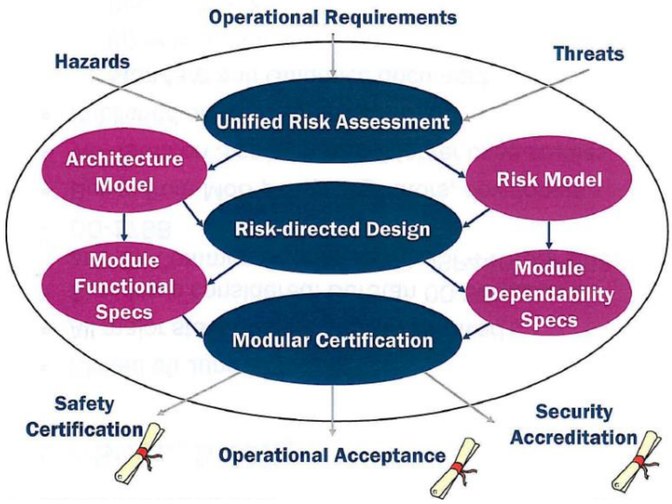


Figure 24: The SafSec approach

❖ Model-based dependability assurance

As previously defined, an Assurance Case is a set of auditable claims, arguments, and evidence created to support the claim that a defined system/service will satisfy the particular requirements [79]. Argument patterns capture successful argument approaches, which can be used within the safety case [79]. They support the extraction of the 'Best practice' arguments based on the company expertise or successful certification approaches used previously.

OPENCOSS project defined an argumentation metamodel [80]. This metamodel is based on the assurance case concept (Figure 25). Each assurance case can be composed of other assurance cases on its turn. Argumentations are keystones, and therefore, OPENCOSS defined its own metamodel for defining argumentations (Figure 26). This metamodel is based on SACM [79] from OMG, but with the particularity that the argument pattern concept is supported.

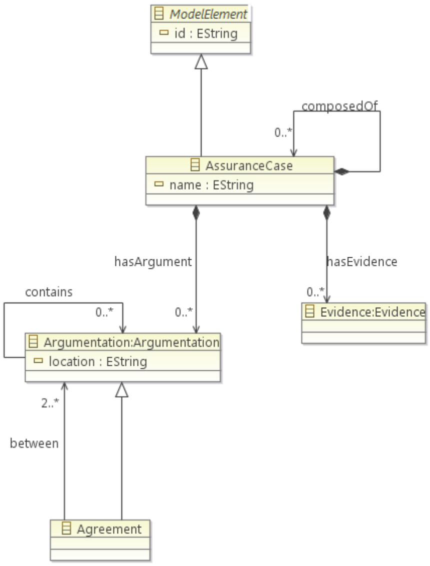


Figure 25: **Assurance Case class diagram**

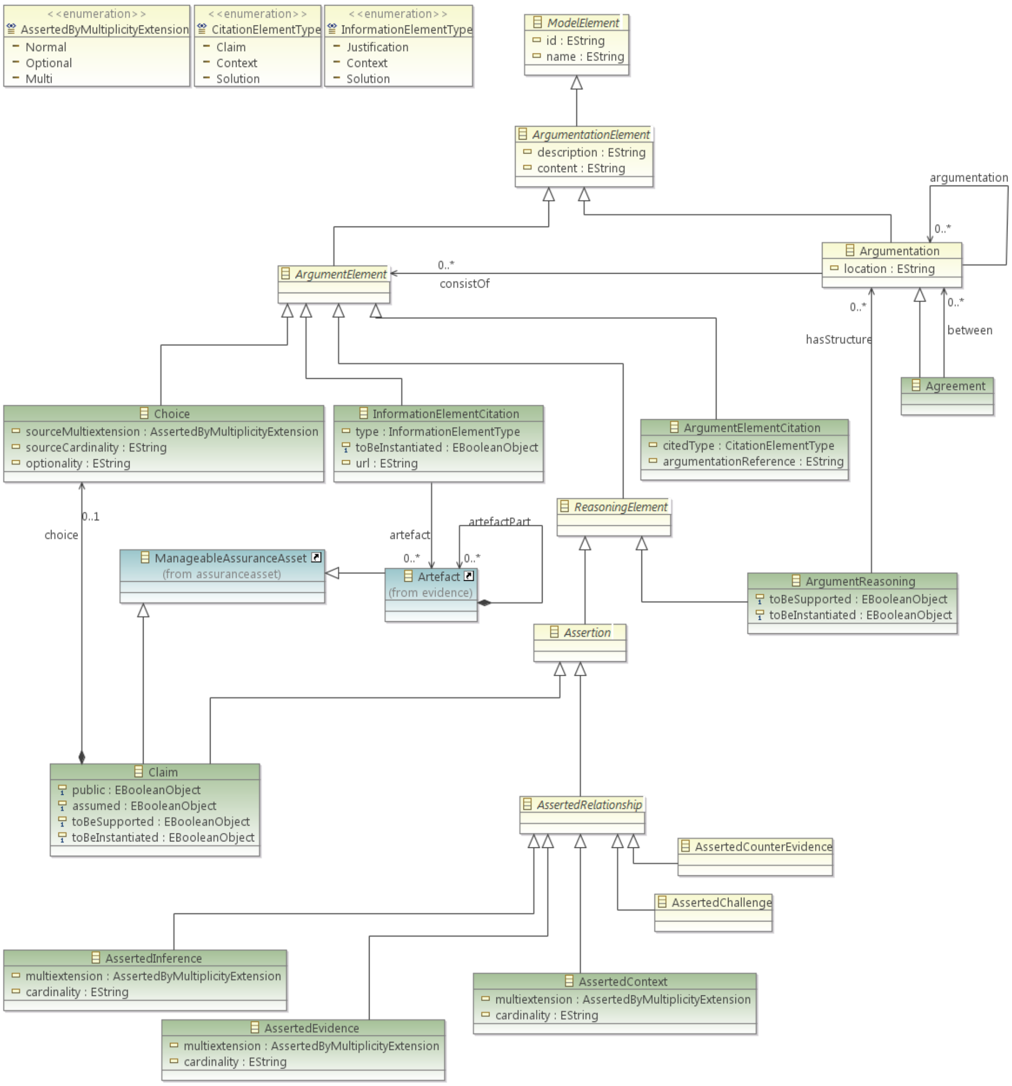


Figure 26: Argumentation Class Diagram

The assumptions and principles underpinning a compositional argument structure are defined for each assurance case. It is hard to write *re-usable* arguments about specific evidence ready for plug-and-play. Each assurance case and argumentations must be composed for a specific system in order to be useful or to demonstrate certain system properties. However, OPENCOSS approach defines the *re-usable patterns* concept for how such arguments may be structured, and reused in other contexts. For example, requirements specification for a software component can be composed by other components. In fact, a system relies on different order to demonstrate a system level failure probability. A re-usable pattern is defined based on the main strengths and weaknesses of this component. In fact, this pattern should take into account its main interfaces, and its behaviour. This pattern is instantiated for a specific component. We can also derive patterns from this pattern. The idea is to apply this approach to assurance cases. Suggested strategies for arguing about how a pattern can be successfully used and derived, it is required. However, project specificities cannot be re-used. Argument patterns contain text to be instantiated (e.g. names of components) as well as choices of strategies to follow when decomposing claims. Figure 27 shows which items may or may not be re-used during the compositional argument process.

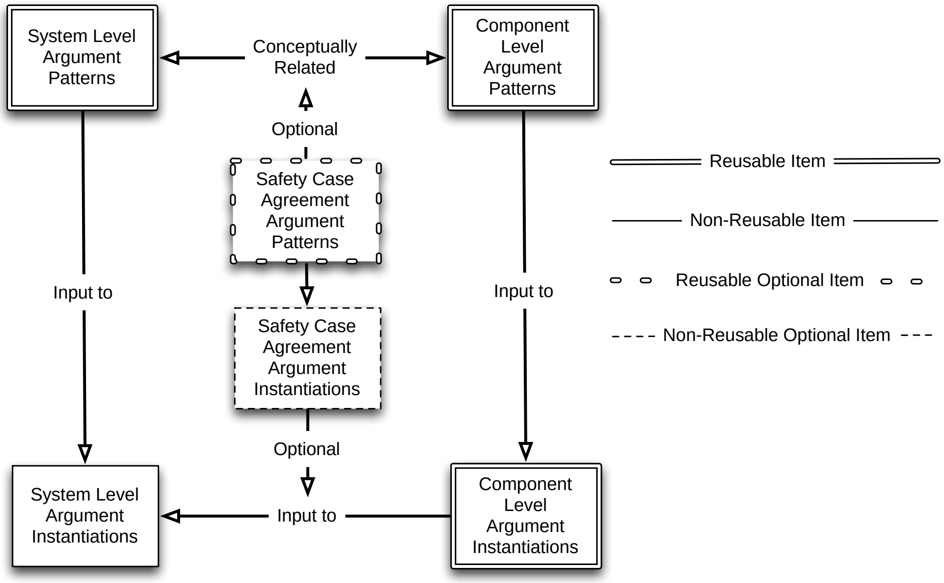

System Level Argument Patterns
Conceptually Related
Component Level Argument Patterns
Optional
Safety Case Agreement Argument Patterns
Input to
Input to
Safety Case Agreement Argument Instantiations
Optional
System Level Argument Instantiations
Input to
Component Level Argument Instantiations
Reusable Item
Non-Reusable Item
Reusable Optional Item
Non-Reusable Optional Item

Figure 27: Relationships between patterns, instantated patterns, and re-usable items in the argumentation approach

OPENCOSS suggests GSN to visually represent these concepts.

An argumentation class is a collection of argument elements. We can encapsulate arguments associated with one component in a module, or in a set of modules. Each component should contain at least one associated argument module in the final safety argument. There are several definitions of what a component is. An argument module is related to multiple components (e.g. for composed evidence).

The re-usable arguments are not only created in order to capture the end point of the development process, but also during component development, to help drive the development process. In a traditional, top-down, approach we would drive system development from system requirements down to component design and then construction. In a bottom-up approach (with a previously developed component) we would reconstruct the design process and "retro-fit", matching system level requirements.

OPENCOSS adopted a hybrid approach, top-down and bottom-up, as shown in Figure 28:

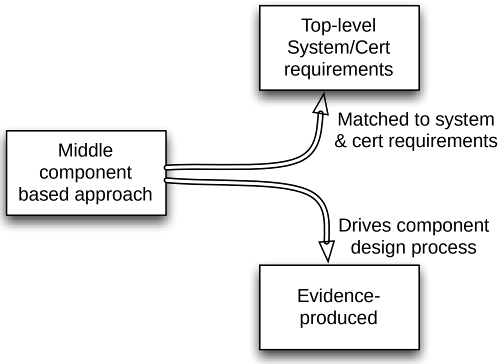

Top-level System/Cert requirements
Matched to system & cert requirements
Middle component based approach
Drives component design process
Evidence-produced

Figure 28: Combining top down and bottom up approaches for compositional certification

Each argument contains a number of different strands of information such as:

1. Common concerns to all arguments namely
    a. Compliance (e.g. to a company process or standard);

   b. Confidence (in both how compelling the argument is, as well as the quality/provenance of the supporting evidence);
   c. Risk (showing that hazards have been adequately met);
   d. Assumptions/Context - background information within which this argument was constructed;
2. Areas/strands of the development process
   a. High-level predictive analyses (for validation purposes and to drive the design, e.g. early FMEAs to derive safety requirements). Evidence and output from these is typically linked to further analyses and artefacts;
   b. Lower-level and confirmatory analyses or design artefacts (for verification purposes, e.g. static code analysis results, testing, code, functional properties). These may be more likely to be standalone, and not required as input to other items.

3.1.4.1 Security Assurance

As the concept of security assurance is manifested in the research community and the industry in several related but slightly different ways, we review the state of art on security assurance according to these related sub-concepts.

A comprehensive view to information assurance shows it involves both technical and organisational aspects. Figure 29 illustrates the so-called “information assurance ecosystem”. The centre of the assurance system is the assurance target, either a security control or the competence required to assess such security controls. The assurance techniques are methods or activities to assess the security target. The assurance schemes are specifications such as standards or qualifications that define (set) the assurance target and techniques, e.g. Common Criteria or ISO 27001. The use of assurance techniques to assess an assurance target generates audit or assessment evidence. Such evidence is used to assess compliance to an assurance scheme.

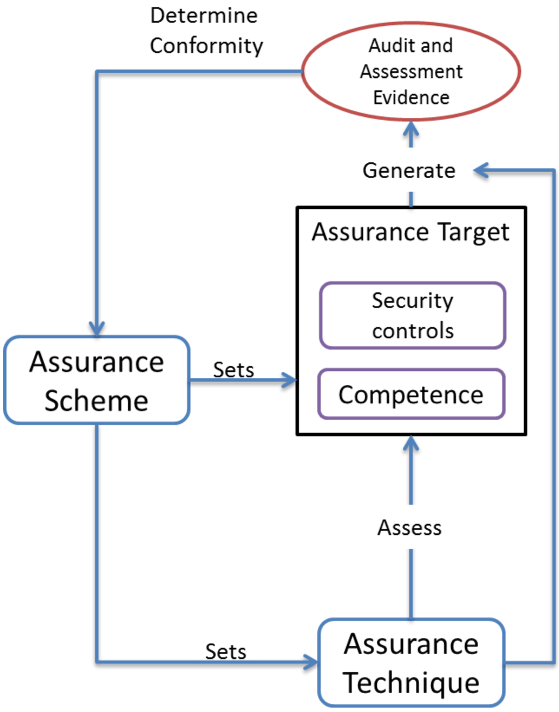


Figure 29: Information assurance ecosystem [65]

Common assurance techniques can be divided into five categories, including review, test, interview, observe, and independent validation. Review techniques include review of documented policy, procedure, and process, review of self-assessment form, threat assessment, architecture/configuration review, and code review. Test techniques include vulnerability scan, penetration test, red team simulated attacks, social engineering, static/dynamic analysis, fuzzing, formal verification, cryptographic validation, and emanation security analysis. The practice of observe is to identify deviation of system behaviour. Interview questions individuals to gather information related to security. Independent validation uses a third party such as an independent witness (in witnessed test) or the public (in public review).

Beside security controls, security assurance also means secure development process that identify, address, and eliminate security issues during the whole system lifecycle. Microsoft Secure Development Life Cycle (SDLC) is adopted in many software companies in various forms to increase software assurance. The principle of SDLC is a clear definition of activities and assurance techniques, from requirement, design, implementation, verification, to deployment and maintenance. The assessment of assurance is usually done by providing the customer with documentations of the development process or through a third party validation.

The High-Assurance Cyber Military Systems (HACMS) program [66] of DARPA aims to develop technology for high-assurance cyber-physical systems. In the project, high assurance is defined to mean functionally correct and satisfying appropriate safety and security properties. HACMS seeks a synthesizer capable of producing a machine-checkable proof that the generated code satisfies functional specifications as well as security and safety policies.

Industry- or country-specific standards are created to ensure the consensus of assurance criteria. They are often used by organizations as a basis for implementing security assurance mechanisms and assessing their assurance levels. For example, Common Criteria (ISO/IEC 15408) is a standard for assurance assessment, widely used for evaluating security in commercial IT products. It defines the process for the specification, implementation, and evaluation of security-critical, high-assurance systems. The security requirements of different classes of devices are captured in different Protection Profiles. It also defines assurance categories and levels and matches them to the Evaluation Assurance Levels (EALs). However, different devices might have different EALs in the same Protection Profile, which makes product comparison difficult. As a result, the National Information Assurance Partnership removed the EAL system and instead lists products as simply compliant to their Protection Profile. FIPS 140-2 is a U.S. government standard for addressing cryptographic modules assurance. Standards focusing on assurance in development process include ISO 27034 for applications, IEC 62443-4-1 for Industrial Automation and Control Systems, and SAE J3061 for automotive systems. It should be note that IEC 62443 and SAE J3061 have strong ties to industry-specific safety standards. For example, SAE J3061 is linked to ISO 26262.

3.2 Safety and Security Co-Engineering

3.2.1 Co-design

In Contract-Based Design, the component interfaces are enriched with formal specification of assumptions and guarantees. These are properties specified in terms of the components' input/output. When the components have different concerns such as safety and security, their contracts must formalize assumptions and guarantees related to multiple concerns. For example, a component may provide a certain service assuming that the user is authorized to use such service and assuming that there is at most a single failure in the underlying hardware components.

When the specification language is expressive enough, multi-concerns contracts can be specified uniformly in the same language. In that case, the interaction among concurrent multi-concern contracts will not be different from the interaction among concurrent single-concern contracts. For example, if a safety contract specifies that every room has an emergency exit that is always open from inside and a security contract specifies that a special room containing confidential material must be always locked, a formalization of these contracts will lead to a logical contradiction, which can be automatically checked.

Multiple Independent Levels of Security (MILS) is a high-assurance architecture for secure information sharing that builds on and extends a long tradition of work on architectural approaches to security and safety. [23] describes MILS approach developed as part of the Multiple Independent Levels of Security / Safety initiative of the Air Force Research Laboratory (AFRL). Its mechanisms are closely related to the *robust partitioning* employed for safety in Integrated Modular Avionics (IMA), and to the *separation kernels* employed in some secure systems. MILS is characterized by a two-level approach to secure system design: (1) at the policy level, a decomposition to a virtual architecture is performed while identifying the trusted components, the local policies and the communications channels; (2) at the resource sharing level, implementation of components is considered, which includes the allocation of components to shared physical resources. Security is rarely identified with a single, simple policy; the two-level approach of MILS was introduced as a rational way to organize the multiple cooperating components and sub-policies that realize a complete secure system (cf. Figure 30).

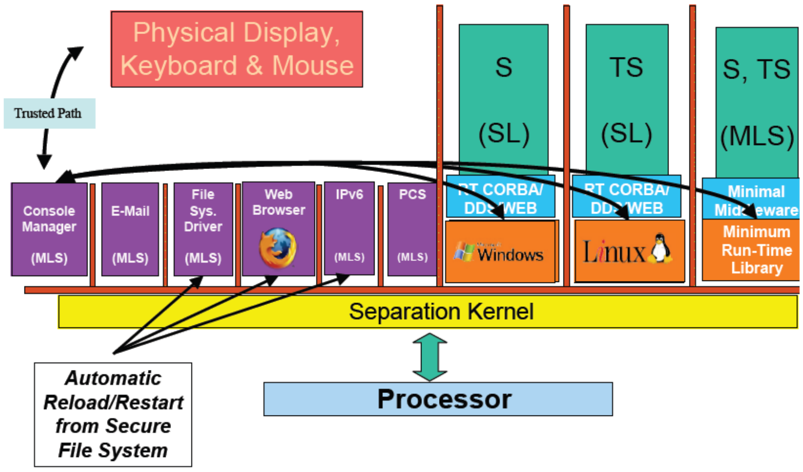


Figure 30: Conceptual Design for a MILS Workstation

A formal Contract-Based specification and analysis of safety and security contracts was developed in the D-MILS project. As an illustrative example, [24] provides an overview of the approach to safety and security. The MILS architecture is well known to ensure properties that are relevant to both safety and security. The project develops a distributed version of the MILS architecture: the D-MILS concept extends the capacity of MILS to implement a single unified policy architecture to a network of separation kernels. To accomplish this, each separation kernel is combined with a new MILS foundational component, the MILS Networking System (MNS), producing the effect of a distributed separation kernel. Robustness and determinism of the network is ensured through the use of Time-Triggered Ethernet (TTE). The project offers a rich tool set. This work focuses on the contract-based method extension, to prove that the composition of components that satisfy their contracts will meet the system requirements, provided that their integrity is protected (cf. Figure 31)in which contracts are represented by green scrolls). The approach is illustrated on a simple multi-level security case, whereby e.g. message authenticity and data confidentiality are shown to be preserved. This example was analysed with OCRA, developed in the SafeCer project.

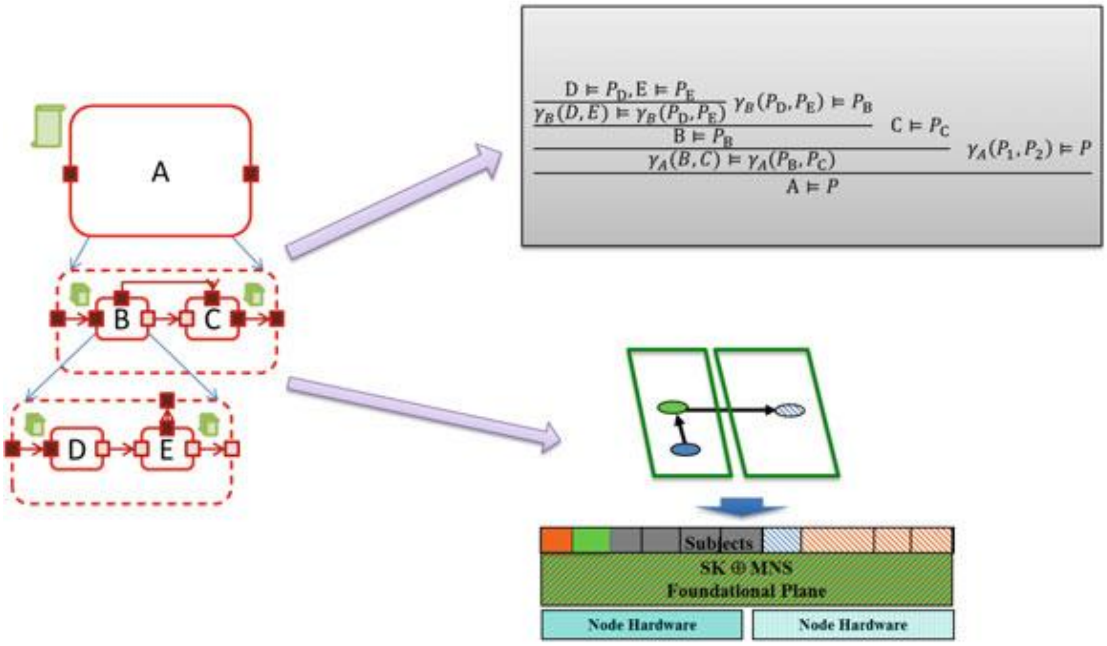

A
B
C
D
E
$D \models P_D, E \models P_E$
$\gamma_B(D,E) \models \gamma_B(P_D,P_E)$
$\gamma_B(P_D,P_E) \models P_B$
$B \models P_B$
$C \models P_C$
$\gamma_A(B,C) \models \gamma_A(P_B,P_C)$
$\gamma_A(P_1,P_2) \models P$
$A \models P$
Subjects
SK ⊕ MNS
Foundational Plane
Node Hardware
Node Hardware

Figure 31: Formal reasoning and platform configuration based on the system architecture [24]

The work presented in [25] introduces SysML-Sec, a SysML-based Model-Driven Engineering environment aimed at fostering the collaboration between system designers and security experts at all methodological stages of the development of an embedded system. A central issue in the design of an embedded system is the definition of the hardware/software partitioning of the architecture of the system, which should take place as early as possible. SysML-Sec aims to extend the relevance of this analysis through the integration of security requirements and threats. In particular, the author proposes an agile methodology whose aim is to assess early on the impact of the security requirements and of the security mechanisms designed to satisfy them over the safety of the system. Security concerns are captured in a component-centric manner through existing SysML diagrams with only minimal extensions. After the requirements captured are derived into security and cryptographic mechanisms, security properties can be formally verified over this design. To perform the latter (cf. Figure 32), model transformation techniques are implemented in the SysML-Sec toolchain in order to derive a ProVerif or UPPAAL specification from the SysML models.

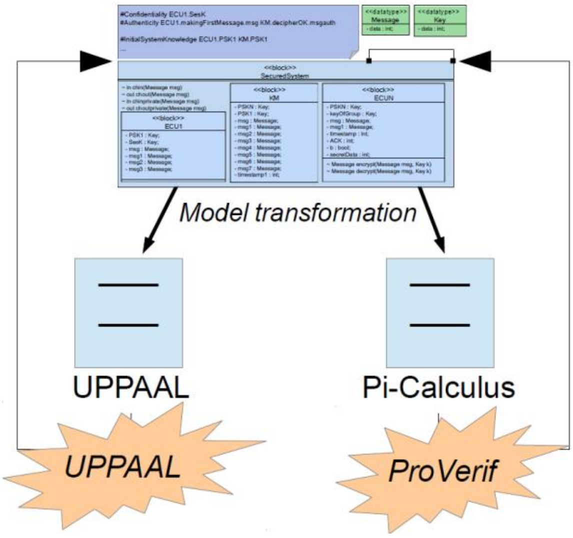


Figure 32: Model transformations for proving safety & security properties

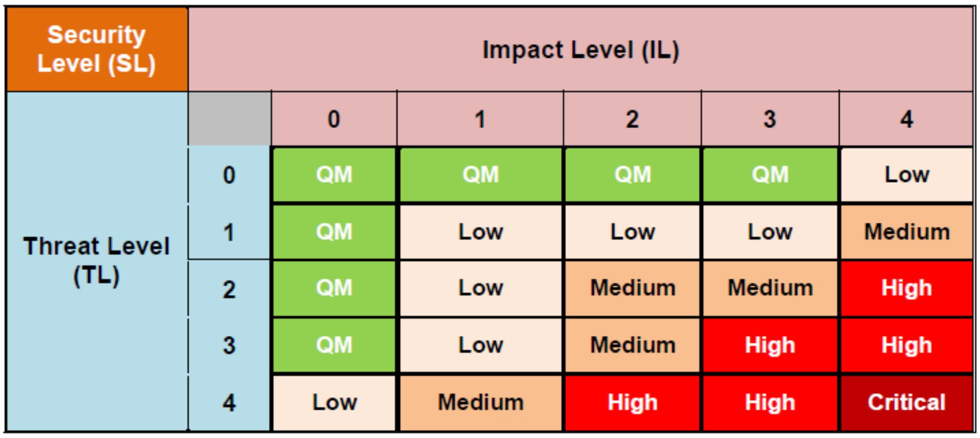
HEAVENS [26] was a Swedish research project aimed at finding security methodologies and tools for software security testing for the automotive domain. One of the outcomes of the project is a new security model - the HEAVENS security model [27]– that facilitates threat analysis and risk assessment and is more attuned to the needs of the automotive industry than existing approaches, as well as having a workflow similar to that of ISO 26262. This security model is one of the models recommended in the SAE J3061 cybersecurity handbook, and has been proposed as part of the input material for a new ISO standard for automotive security. In addition, the project has analysed the interplay between safety and security, for instance to understand how safety mechanisms impact security and vice versa to better handle the trade-offs between safety and security requirements, and how to perform security testing [28].

❖ Heavens security model

For threat analysis, i.e. identification of potential threats for the target of evaluation (TOE) asset, Microsoft's STRIDE approach [29] is used. In the risk assessment phase, a new model is proposed where a security level is assigned to each TOE asset. It builds upon knowledge from earlier initiatives such as EVITA [30]. The security level, in turn, is based on impact level and threat level (see Figure 33). The threat level is based on the properties expertise, knowledge about TOE, window of opportunity, and equipment needed. For each of these properties, there are assessment guidelines. For instance, if standard equipment is enough for performing an attack, the equipment score is 0, if specialized equipment is needed, the score is 1, for bespoke equipment 2, and multiple bespoke 3. When all properties are assessed, an evaluation table will give a final threat level between 0 and 4, where 4 is the most severe threat. The impact level is similarly assessed from the properties safety (risk of injuries), financial (risk of losing money), operational (risk of disrupting operation), and privacy/legislative (risk of violating legislation or privacy). The safety property is the same as for ISO 26262, that is no injury, light injury, severe injury, and fatal injuries. The safety and financial risks have a higher weight in this model, see Figure 34. A table is then used to sum the values and assign an overall impact level. Based on threat level and impact level, the security level is assigned, as shown in Figure 33. The security level is further assigned to security requirements in the same manner as ASIL is assigned to safety requirements in ISO 26262. Such security requirements are formulated for each asset that has a security risk.

❖ Relevance for AMASS

In AMASS, we will investigate whether it is possible to use the HEAVENS security model for threat analysis and risk assessment when creating the security aspect of multi-concern assurance cases that, as this model is mentioned in SAE J3061 standard for cybersecurity in the automotive domain.



Security Level (SL)		Impact Level (IL)				
		0	1	2	3	4
Threat Level (TL)	0	QM	QM	QM	QM	Low
	1	QM	Low	Low	Low	Medium
	2	QM	Low	Medium	Medium	High
	3	QM	Low	Medium	High	High
	4	Low	Medium	High	High	Critical

Figure 33: HEAVENS security model – Security Level (SL) matrix.

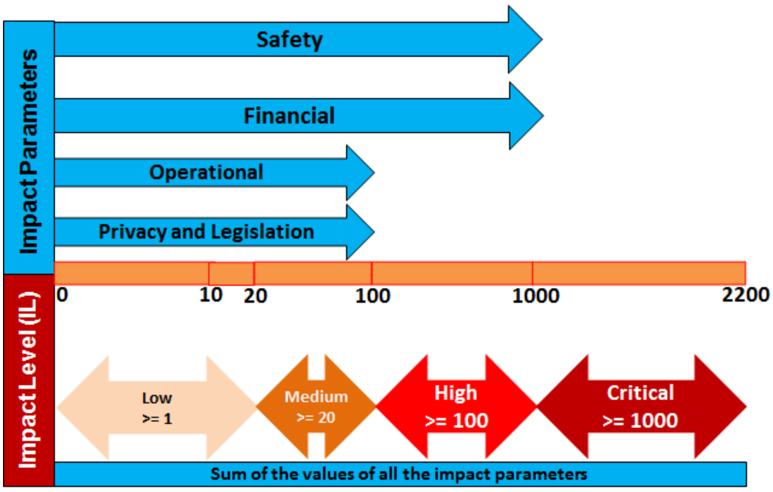


Figure 34: HEAVENS security model – Impact Level (IL) parameters and weights.

3.2.2 Co-analysis

In this section, the state of the art on safety and Security co-analysis is presented through diverse works and projects enhancing methodologies and approaches commonly known and used in safety and security domains.

3.2.2.1 Model-Based Safety Analysis

There exist several approaches using for performing safety analysis based on UML/SysML. The approaches in general used models to perform classical safety analysis like hazard analysis, fault tree (FT) generation and analysis (FTA), failure mode and effects analysis (FMEA).

Mhenni et al. [31] described a methodology based on SysML models for generating (semi-) automatically FMEA and FTA artefacts. FMEAs are created firstly in the functional and structural design based on SysML structural diagrams. For FTA, the structural diagrams including the component FMEA results are transformed into directed graphs and a graph traversal is performed to identify different patterns and progressively build fault trees. Specific fault trees regarding to specific undesired top events can be derived from the auto-generated one. David et al. [32] proposed to incorporate FMEA flow into the functional design using SysML. Xiang et al. [33] propose an approach for generating static Fault Trees from Maude specifications obtained from SysML models. In their approach, SysML models are translated into Reliability Configuration Model (RCM) with required system information on its structure, functional dependencies between components, etc. A Static Fault Tree Model (SFTM) is defined and Fault trees are generated from SFTM based on the reliability configuration information presented in the RCM. Both the RCM and SFTM are specified using the algebraic formal specification language called Maude.

Yakymets et al. [34] presented Sophia4Safety, a tool suite integrated within the Papyrus UML/SysML modeler. It supports model-based safety analyses methodology and provides various MBSA services including FTA, FMEA, hazard analysis, requirement engineering, etc. Sophia includes facilities to automatically perform these safety analyses and generate safety-related artefacts, to make semantic connections with formal tools, e.g. Altarica, NuSMV, to represent safety analysis results in the system modelling environment. Within the tool suite, safety property verification and reachability analysis can be performed. Sophia4Safety framework complies with the requirements of some standards for its proposed analysis, e.g. the hazard analysis and risk assessment as specified in ISO 26262.

Biggs et al. [35] introduced the SafeML profile, a SysML profile designed for modelling the safety-related concerns of a system with the help of supporting tools. SafeML models common safety concepts from safety standards and safety analysis techniques like FMECA and FTA; integrated with system design information.

Similar studies are also undertaken with other modelling languages such as Simulink, Architecture Analysis and Design Language (AADL). Tajarrod et al. [36] constructed fault trees from MATLAB Simulink models. To do so, nominal Simulink models are extended with system's failure behavioral information, and then fault tree for a specific top event is automatically constructed. Joshi et al. [37] proposed an automatic generation of fault trees from AADL models. AADL helps to specify the system architectural model and annotate it with fault and failure information using the AADL Error Annex. Based on the annotated model the Fault Trees can be generated automatically.

3.2.2.2 Model-Based Security Analysis

Few model-based approaches targeting security analysis exist. Sophia4Security [38] is a model-based security framework that is built on top of Papyrus tool as for Sophia4Safety. Sophia4Security offers different security analysis methods like for example the EBIOS, a qualitative method which supports ISO 27005 standard. Other techniques proposed in the framework are vulnerability and threat analysis, attack trees analysis, and detectability analysis that allows defining countermeasures against attacks. The framework allows automated assessment of the criticality in the models and provides the results as risk matrices.

3.2.2.3 Model-Based Safety and Security Co-Analysis

Certain efforts have been put into investigation to assess jointly safety and security of systems through the model driven engineering process. Many of these approaches adapt classic safety analyses, e.g. FTA, to introduce and address security concerns together with the safety concerns.

Fovino et al. [39] presented an attack tree analysis based on an extended Fault tree to capture malicious risks. The approach enables quantitative analysis by assigning probabilities to leaf nodes and calculating the probabilities of higher nodes through propagation. Kornecki et al. [40] used FTA to develop safety and security requirements of a system and associated mitigation measures. The approach help identify both hazards and threats that can lead to accident on the same fault tree model and provide quantitative analysis on the model. Bezzateez [41] used FTA to model risks related to safety and security concerns on the same tree and check how security can affect safety of the system. Steiner et al. [42] also presented an approach extended fault tree of component to perform attack trees and model security events that can influence system safety. In the approach, minimal cut sets that can contain either only safety events, either only security events, or both safety and security events can be assessed.

Approaches based on UML to co-assess safety and security issues within a system have been developed.

Sindre [43] used UML use cases combining both safety and security threats of the system to help elicit requirements. The Chassis method is an UML-based approach for safety and security assessment [44]. The method comprises different activities for functional, safety and security requirements elicitation and specification relying on UML diagrams (use case, sequence) and traditional analysis methods such as HAZOP and FMEA. The UMLsafe/sec approach [45] extends UML for safe and secure systems development. UMLsafe/sec profile allows modelling of safety/security requirements, failure/attack scenarios and assess these scenarios. The approach is tool supported and provide automated analysis of the models. SysML-sec [46], similarly to UMLSafe/Sec, is an extension of SysML as a profile for designing safe and secure systems
Other approaches supported by alternative modelling language exist:

Kornecki et al. [47] used Bayesian belief network to address safety and security jointly, and assess the impact of one to the other. Mitchel et al uses Stochastic Petri nets in their approach for analysing intrusion and the influence on the reliability of a system. Roth et al. [48] defined an approach that introduce security concerns into a safety model and enables quantitative analysis of safety and security. The approach combines fault tree and state charts that are translated into Petri nets for the quantitative analysis. The Boolean logic Driven Markov Processes is a modelling formalism, initially dedicated for safety and reliability assessment, and that have been extended to introduce security-related analysis. BDMP combines fault tree with Markov processes to enable advanced quantification analysis as well as modelling detection and response mechanisms against attacks [49].

Following projects elaborated the support for Model-Based Safety and Security Co-Analysis:

SESAMO - Support for model-based safety and security assessment was investigated in the context of SESAMO ARTEMIS JU project; for instance, a dedicated tool chain was used to analyse safety and security related system requirements and specify critical parts of the system architecture. The tool chain comprises the Medini Analyse and the CHESS tools. Medini Analyze was used to specify the safety and security related requirements and their allocation to system components. Hazard analysis and risk assessment was also applied to the modelled entities allowing to refine the requirements, catch interference of these requirements and so the specification of the safe/secure system design. Then the system requirements and the system model were transferred through automatic model transformation from Medini Analyse tool to the CHESS tool for further refinements. In particular the CHESS tool provides support for dependability and schedulability analysis to be applied at model level; in SESAMO CHESS analysis support was used to analyse the impact of security and safety functions on the performance of system critical tasks. By applying the aforementioned support, safety and security mechanisms can be properly tailored and adjusted to fulfill the system performance requirements during the early phase of the system design.

CONCERTO - The CONCERTO project [50], spanning from May 2013 until April 2016, focused on the development of support technology for the use of model-based engineering artefacts and solutions for the end-to-end development process of critical real-time software systems that target multicore processors. The CONCERTO toolset embraces a model-based component-oriented approach to the design of software systems. The CONCERTO toolset highlights include a model-based analysis framework for dependability and real-time concerns, augmented with back propagation to the user model (to ensure consistent handling of analysis feedback) and automatic generation of architectural code to host the application functions and deploy them in trusted run-time containers.

The CONCERTO modelling language and technology solution have been defined and implemented as contribution and extension to the CHESS [51] ones (currently hosted in the Eclipse Polarsys ecosystem [52]) to provide a larger and wider coverage of industrial domains.

In CONCERTO/CHESS modelling language [53], timing properties (e.g. period, worst case execution time) and constraints (e.g. deadline) for components can be specified by using a subset and (little) extension of the OMG MARTE modelling language. Then feasibility in the timing domain for the system under design can be verified by specialised state-of-the-art schedulability analysis consistent with the intended regime of execution on the target platform.

Through the Dependability Profile defined in CONCERTO (part of the CHESS modelling language), functional models of the system can be enriched with information regarding its behaviour with respect to faults and failures, thus allowing properties like reliability, availability, and safety to be documented and analysed.

In CONCERTO, the Dependability Profile has been extended to address socio-technical systems by incorporating modelling capabilities related to Human factors. More specifically, via the extended profile, not only technology (hardware and software) is taken into consideration. Organizations and humans are also considered and are interpreted as composite components, constituted of sensor- and actuator-like subcomponents, and their nominal as well as failure behaviour can be specified and then used to calculate the failure behaviour emerging at system level.

MERgE - A Model-Based Safety and Security Assessment approach was developed in the context of the MERgE ITEA project [54]. This approach [55] consists of decoupling the system architecture model from safety & security views (cf. Figure 35). So, every engineer, whether architect, or security/safety engineer, can focus solely on his concerns, with dedicated tools and terminology. As of now, it is possible to use two separate views: one for the safety concern and the second for the security concern. The main motivation for this separation is that the safety and security domains are quite different in terms of practices, concepts used and wording.

As the safety and security views rely on the system architecture model, the required information is extracted (e.g. function interactions, ports and their links, data…) from the architecture model and an initial safety and security views are set up in Safety Architect (ALL4TEC tool). Starting from this, safety and security engineers enrich their respective views by adding safety and security dysfunctional behaviour. The safety and security views contain two kinds of information:

- the safety and/or security dysfunctional behaviours; a safety dysfunctional behaviour represents how errors are propagated in the system architecture, and a security dysfunctional behaviour represents how the effects of security threats are propagated in the system architecture;
- the safety and/or security properties / requirements that the system architecture must satisfy, e.g. the integrity of the output data shall be preserved even under specific attacks.

These two views are then combined to produce a formal Alloy model containing all the necessary inputs for the analysis. The Kodkod tool formally validates the safety and security properties. If a property is violated, Kodkod will show a readable counter-example. Thus, the engineers can identify the best way to correct the architecture.

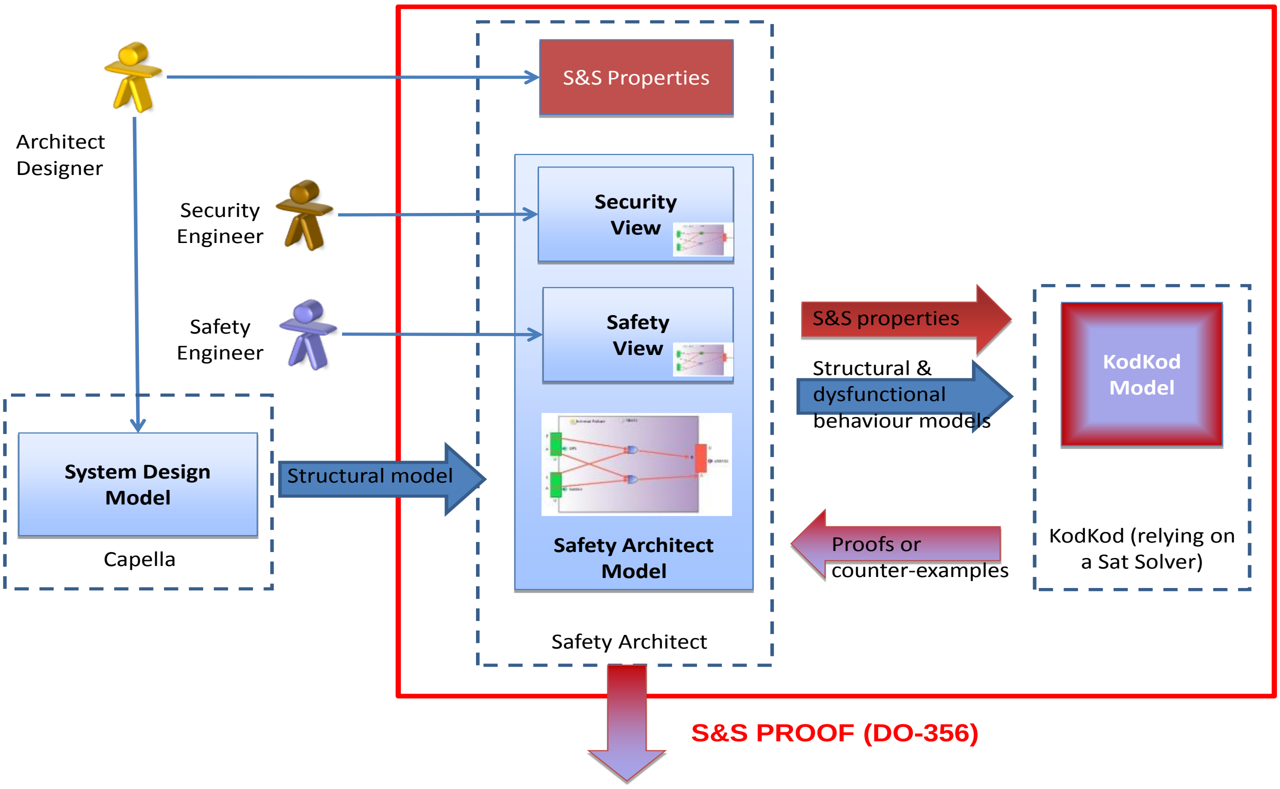


Figure 35: Model-Based Safety and Security Assessment integrated approach developed in the MERgE project

The main principles of this approach are illustrated in the Figure 35. The first model transformation, from Capella tool, produces an initial Safety Architect model from the System Design Model. This transformation is trivial as it only reflects the structural part of the architecture. Together with Capella, projects made from the usual modeling tools such as MagicDraw, Rhapsody, SCADE Architect or Papyrus are directly importable [70]. Papyrus together with the CHESS profile extension constitute the basic building block that supports the System Component Specification in AMASS.

Then, safety engineers and security engineers can work within Safety Architect, either using separated views or a merged view, to describe the way failures and security threats propagate inside the architecture: this activity is called dysfunctional modelling. Then dysfunctional analysis techniques already available in Safety Architect can be applied, such as the automatic generation of fault trees or attack trees.

The last part of the approach consists of taking benefits from the Alloy formal specification and verification method, and its relying Java Kodkod API, to enhance the analyses that can be performed. The idea is to build a Kodkod model that represents the architecture of the system and the rules that describe the way failures and threats propagate inside each block, and along the architecture. Then all the expressive power of the Alloy logic and the verification capability of Alloy Analyzer/Kodkod can be used to check various kinds of properties. In the proposed approach, the building of the Kodkod model and the verification of formal properties are prototyped and called from Safety Architect.

MERgE project's outcome (i.e. Safety Architect tool) is compatible with the system architecture driven assurance approach proposed in AMASS project. Indeed, the main principle of MERgE approach is that assurance viewpoints (safety and/or safety views) rely on the system architecture model annotated by assurance artefacts or imported from system architecture modelling tools (e.g., Capella, Papyrus, SCADE Architect) to dedicated assurance analysis tools (e.g., Safety Architect for Safety/Security co-analysis).

The same principle is used in AMASS project by exploiting the CHESS architectural design. In order to exploit the results of the MERgE project, a bridge between AMASS platform (CHESS tool) and dedicated tools (Safety Architect tool) is proposed. The AMASS project proposes in addition other architecture driven assurance methods on technical level (Contract-Based Safety Analysis or Simulation-Based Fault Injection) or on process level to collect all artefacts produced during the early phases of the system design to be used as evidences in the Safety case.

3.2.3 Co-Verification and Validation

The TURTLE UML profile and the open source toolkit TTool define a formal modelling and verification framework for communicating embedded systems design. The work presented in [71] extends TURTLE and TTool with network calculus techniques. Dimensioning diagrams enable system dimensioning prior to usual object-oriented design and facilitates formal verification of design models. The method associated with TURTLE uses an architectural design pattern where two or several protocol entities rely on a pre-existing communication service. Modelling the pre-existing service with empirical values is error-prone and hampers large space exploration during the communication architecture validation. The work presented in [72] relies on the Network Calculus theory to parameterizes the service with realistic upper bounds. The revisited TURTLE method (cf. Figure 36) includes a dimensioning step between the requirement and analysis steps. This new step is based on a "Dimensioning Diagram" that describes the network in terms of traffic and equipment behaviour, and a "Dimensioning-oriented Use Case Diagram" that categorizes the flows conveyed by the network. The paper applies this method to a video conference system as example.

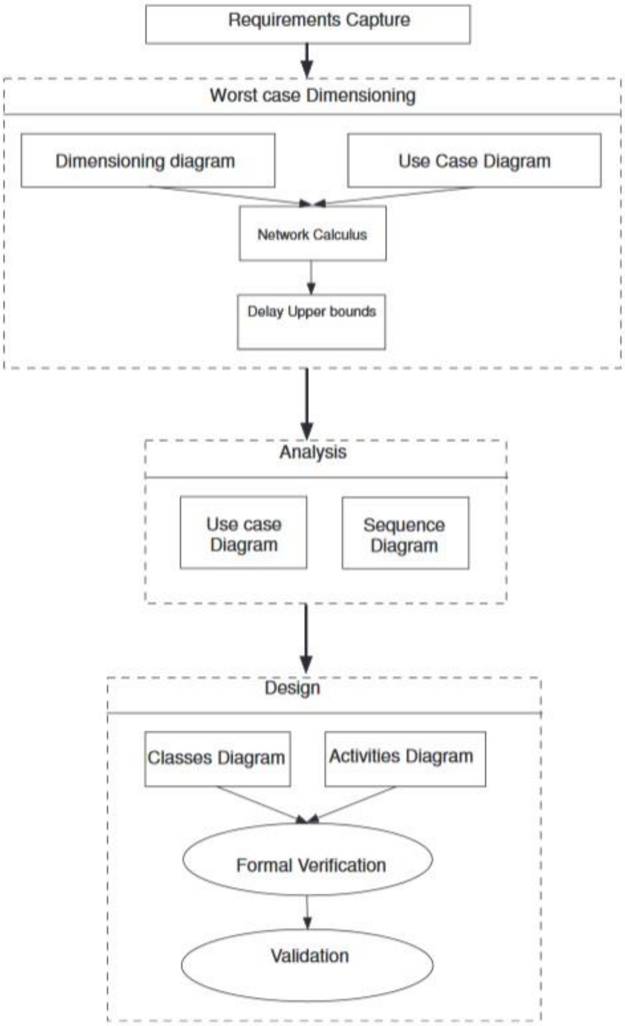


Figure 36: New TURTLE method

AVATAR is a real-time extension of SysML supported by the TTool open-source toolkit. So far, formal verification of AVATAR models has relied on reachability techniques that face a state explosion problem. Apvrille et al. explores in [73] a new avenue: applying structural analysis to AVATAR model, so as to identify mutual exclusion situations. In practice, TTool translates a subset of an AVATAR model into a Petri net and solves an equation system built upon the incidence matrix of the net. TTool implements a push-button approach and displays verification results at the AVATAR model level. The approach is not restricted to AVATAR and may be adapted to other UML profiles.

Brunel et al. [74] proposes an approach based on Alloy to formally model and assess a system architecture with respect to safety and security requirements. In this paper, the authors illustrate this approach by considering as a case study an avionic system, which provides guidance to aircraft. They show how to define in Alloy a metamodel of avionic architectures with a focus on failure propagations. Then to express the specific architecture of the case study in Alloy and finally, to express and check properties that refer to the robustness of the architecture to failures and attacks.

3.2.4 Assurance

The goal of the SQUALE (Security, Safety and Quality Evaluation for Dependable Systems) research project is to develop assessment criteria for obtaining confidence that a system will achieve its dependability objectives [56], [57]. The SQUALE assessment framework and the four confidence providing processes are shown in Figure 37. The framework is applied recursively at each refinement level of the system development process. Hence each refinement step needs to start with hazard analysis that leads to defining the dependability targets. The four confidence providing processes are: dependability requirement validation, correctness verification, system dependability validation, and process quality.

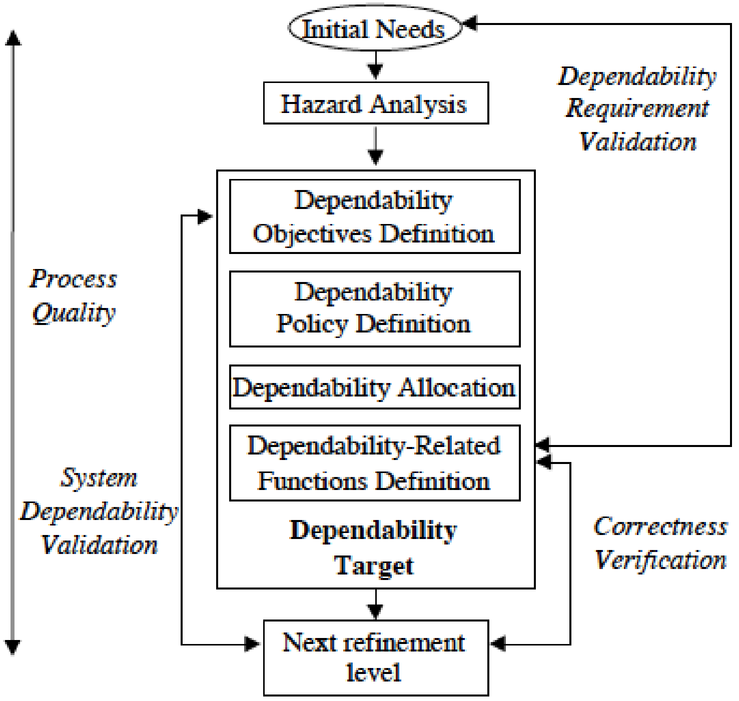

Initial Needs
Hazard Analysis
Dependability Requirement Validation
Dependability Objectives Definition
Dependability Policy Definition
Dependability Allocation
Dependability-Related Functions Definition
Dependability Target
Process Quality
System Dependability Validation
Correctness Verification
Next refinement level

Figure 37: SQUALE assessment framework and confidence providing processes [56]

The Dependability Target is a document where the part of the system that needs to be assessed is described. The initial version of this document is developed early in the development process, and then it is refined at each level of decomposition. The document includes: a description of the system and its environment; the results of the hazard analysis; a set of dependability attribute objectives; the dependability policy; identification and specification of dependability-related functions; the dependability allocation; the dependability profile for each dependability function; and the dependability plan.

SQUALE proposes to use confidence levels from 1 to 4 for all the attributes of the dependability taxonomy by Laprie and Avizienis. For example, confidence levels in availability would be denoted with A1-A4, confidentiality C1-C4, Reliability R1-R4, etc. A0, C0, R0, etc., are used to denote that there is no requirement for the corresponding attribute. A combination of the confidence levels for each of the dependability attributes is referred to as the system dependability profile. The overall assessment focuses on selecting the confidence providing processes and checking that they have been carried out properly to achieve the required confidence.

For further information, please refer to D3.1 and D6.1.

3.2.5 Integration of Safety and Security

In the work presented in [58] the author recalls that Safety has a long tradition in many engineering disciplines. Standards (e.g. IEC 61508), methods of risk and hazard analysis, and certification methods have evolved long before IT. Security has evolved quite recently with networked IT-systems and concerns about privacy, data integrity, authenticity and protection. Both communities have developed their own standards, methods and system views – and neither in standardization nor in application areas they co-operate well. The paper takes a holistic view of critical systems and proposes a unified approach to system dependability, integrating both safety and security, arguing that in case of massively deployed embedded systems security issues have severe safety impact and vice versa.

EMC2 – 'Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments' [59] is an ATREMIS Joint Undertaking project in the Innovation Pilot Programme 'Computing platforms for embedded systems' (AIPP5). Embedded systems are the key innovation driver to improve almost all mechatronic products with cheaper and even new functionalities. They support today's information society as inter-system communication enabler. A major industrial challenge arises from the need to face cost efficient integration of different applications with different levels of safety and security on a single computing platform in an open context. The objective of EMC2 is to establish Multi-Core technology in all relevant Embedded Systems domains and enable mixed-criticality applications for such systems. Focus was on developing and enabling multi-core platform for multiple domains and ensuring safety and security for critical applications based on such domains. Key achievements for safety and security were:

- **Integration of Safety and Security** Engineering to handle the impact of security on safety
- Conditional **runtime certification** enabling safety checks of dynamic system compositions

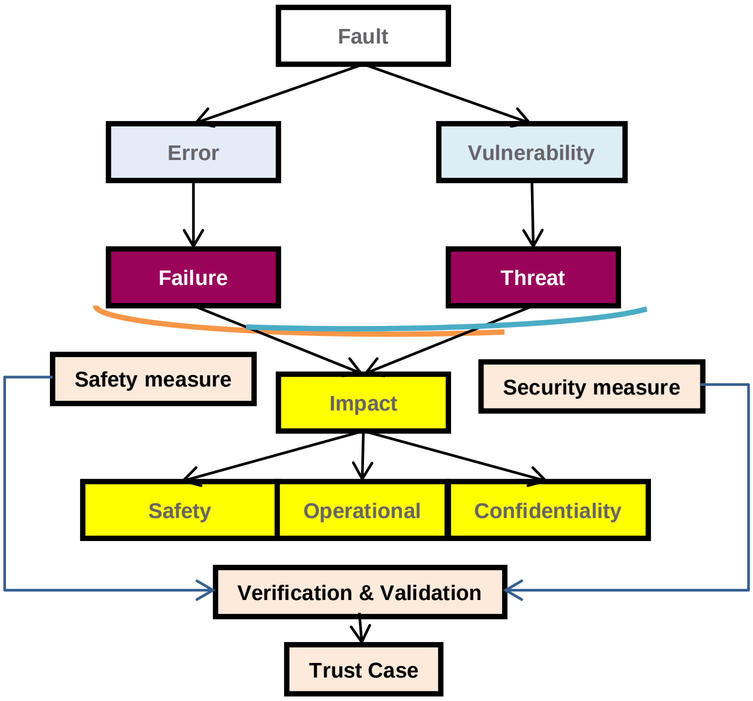


Figure 38: Interactions of safety and security

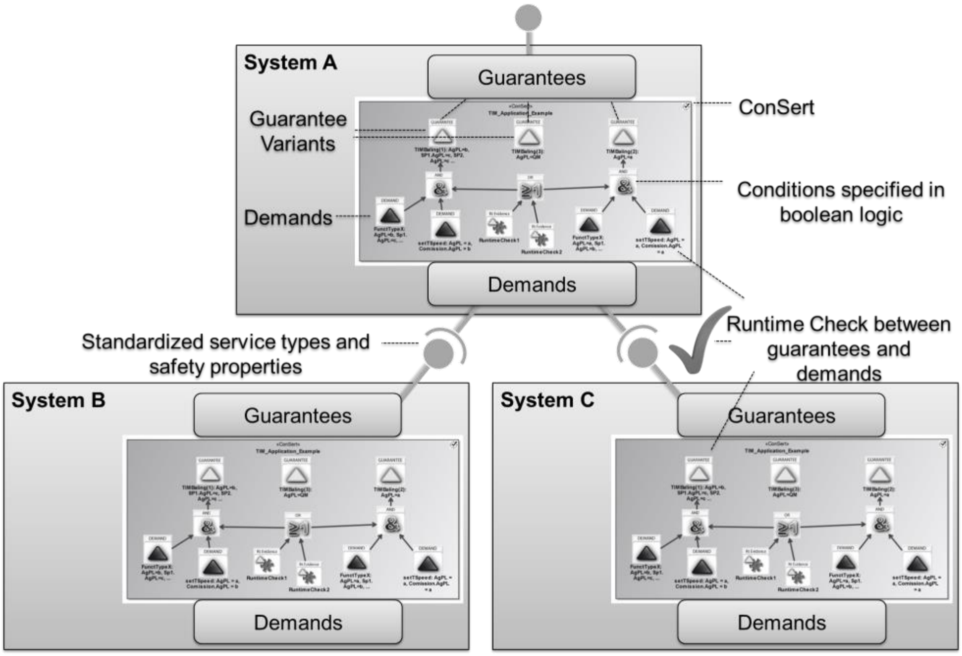

System A
Guarantees
Guarantee Variants
Demands
Demands
ConSert
Conditions specified in boolean logic
Runtime Check between guarantees and demands
Standardized service types and safety properties
System B
Guarantees
Demands
System C
Guarantees
Demands

Figure 39: Conditional Runtime Certificates

Kirillov et al. [60] proposes a new engineering paradigm of integrated safety and security. The author describes at first, a motivation for searching of the new paradigm that is made via brief synopsis of evolution and appropriate limits of the former safety concepts – reliability-based and risk-informed ones. Attention is also focused on two specific features of current situation world-wide – necessity to react or pro-act on multi-hazard threats and inability to protect (to provide guarantee for un-interruptible functioning and full scale performance) all assets at risk. Then, a working definition of building/structure resilience from community resilience and organizational resilience are delineated. In conclusion, a set of the hypotheses, which are important for systematic development and operationalization of multi-hazard resilience paradigm, are introduced.

The work exposed in [61] proposes a comparative analysis of the notions of Safety Levels and Security Levels as defined (under various names) by the relevant standards. This comparison is a basis for the elaboration of a harmonised process to develop and validate embedded systems having to comply with both safety and security requirements (including related certification requirements when applicable), which is the objective of the French collaborative project SEISES [62]. An important case corresponds to systems for which security requirements come from safety needs i.e., the necessity to preserve safety properties even in case of security threats. In such a case it is necessary to identify clearly the dependencies between the Safety and the Security Levels of the system.

Kriaa in [63] believes that in the general case, the risk analysis process should combine both safety and security. In Figure 40 , the author provides a high level view of a safety and security risk analysis process inspired from the generic approaches identified earlier. The first step of this integrated risk analysis process is to perform a hazard analysis to identify the hazardous/unsafe states of the system. Considering the definition of safety in the context of this survey, the hazardous states originate from the system and have an impact on the system's environment. Next, safety and a security risk analyses are realized separately by safety and security experts: safety-related scenarios are identified based on failure mode analysis and security-related scenarios are identified based on an analysis of threats and vulnerabilities that lead to unsafe states. Then, the scenarios are ranked according to frequency and impact, and appropriate safety and security requirements are defined. The two sets of safety and security requirements are next integrated and examined together in order to identify possible interactions. The treatment step addresses the different interactions identified (e.g., conflicting requirements). This step requires collaboration of safety and security experts in order to find solutions that satisfy both sides. New safety and security requirements are considered and interactions are then derived. The system modifications resulting from this first pass may introduce new risks; this is why the process iterates until all interactions are identified and no modifications are needed.

This risk analysis process can be applied to the development or the operational phase of the system lifecycle (with items being requirements or design features in the development phase or actual countermeasures in the operational phase).

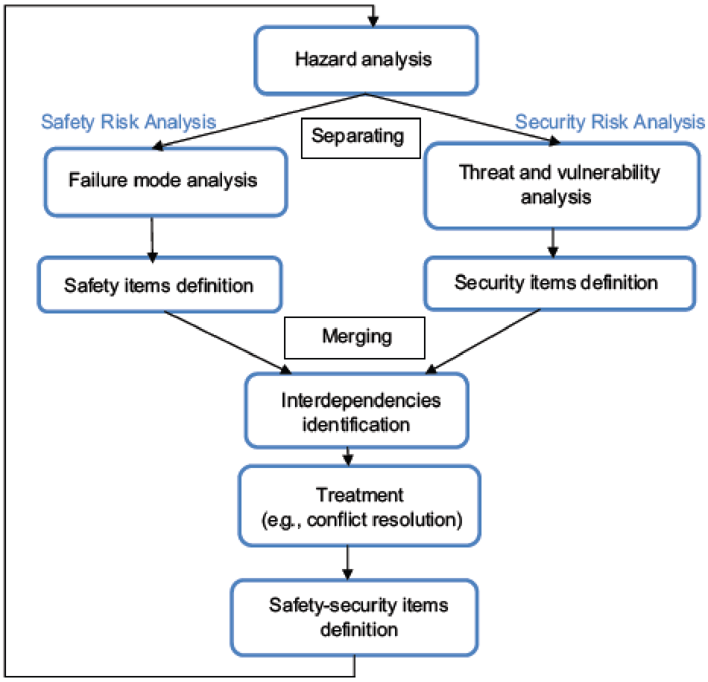

Hazard analysis
Safety Risk Analysis
Separating
Security Risk Analysis
Failure mode analysis
Threat and vulnerability analysis
Safety items definition
Security items definition
Merging
Interdependencies identification
Treatment (e.g., conflict resolution)
Safety-security items definition

Figure 40: Safety security integrated risk analysis process

The work reported in [64] describes and summarizes the results of the authors' research of dependability models that can be used to arrive at an integrated and analysable model for safety and security issues of a system, and their interdependence. The report proposes a combinational model-based on the integration of attack trees into fault trees for a qualitative and quantitative safety/security analysis Figure 41. The strengths and weaknesses of different model variants are discussed.

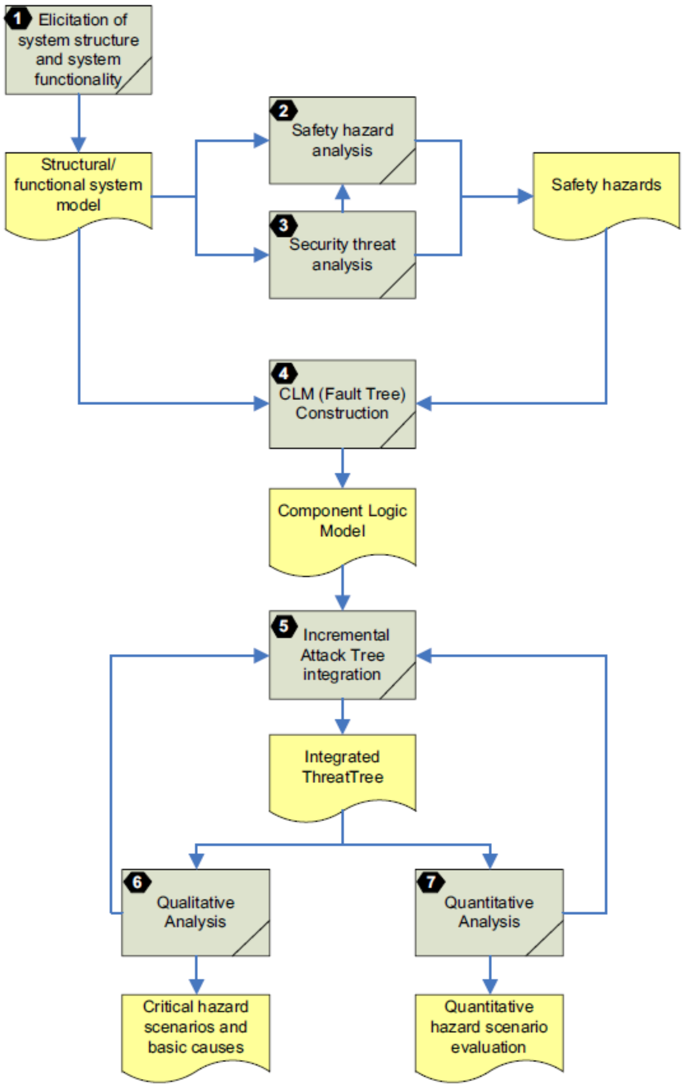

1 Elicitation of system structure and system functionality
Structural/ functional system model
2 Safety hazard analysis
3 Security threat analysis
Safety hazards
4 CLM (Fault Tree) Construction
Component Logic Model
5 Incremental Attack Tree integration
Integrated ThreatTree
6 Qualitative Analysis
7 Quantitative Analysis
Critical hazard scenarios and basic causes
Quantitative hazard scenario evaluation

Figure 41: Threat tree assessment process

4. State of the Practice on Multi-Concern Assurance

Multi-concern assurance is one of the main objectives of the AMASS project. Within these concerns, we have identified that safety and security assurance are among the main focuses for the stakeholders in the industry regarding cyber-physical system of systems (SoS) engineering. Traditionally many safety systems are designed under the assumption that they will be operated in a trusted operational environment. In such environment, threats from malicious activities are ruled out as improbably. Such assumptions are increasingly unjustifiable. A number of recent incidents show that many systems from automobiles to medical devices can be made to malfunctioning remotely leading to safety hazards.

Another aspect of safety and security is in the system development lifecycle. Software vulnerabilities and weaknesses are commonly introduced into a product in the design and implementation phase. A rigorous process of software development lifecycle does improve security, as manifested by Microsoft's Secure Development Lifecycle (SDL) program. Designing safety-critical systems is typically a result of rigorous software development process. Therefore, safety and security activities are likely to be integrated in a holistic way in existing development methodologies and lifecycle models.

Table 5 lists some of the current standardization approaches w.r.t. safety and security concerns.

Table 5: Safety and security standard initiatives for different domains

Domain	Standard	Information
Railway	EN 50129	Pre-standard **DIN V 0831-104**. Integrative approach (with IEC 62433, SL1). Activities particularly in Germany (DKE)
Avionics	DO-326A (ED-202A)	Airworthiness Security Process Specification
	DO-355 (ED-204)	"Information Security Guidance for Continuing Airworthiness" covers operations and maintenance, published in June, 2014
	DO-356	"Airworthiness Security Methods and Considerations", published in September 2014
Nuclear Power Plants	IEC 62589	"Nuclear power plants –Instrumentation and control systems – Requirements for coordinating safety and cyber security"
Automotive	J3061	Create Cybersecurity Guidebook for Cyber-Physical Automotive Systems Consistent with risk methodology in ISO 26262 Functional Safety Standard. Contains automotive Cybersecurity framework and processes. Evaluates Threat Analysis and Risk Assessment (TARA) methods. Simple approach to allow effective implementation across the automotive industry.
	J3101	Define a common set of requirements for security to be implemented in hardware for ground vehicles to facilitate security enhanced applications, developing expectations for necessary functionality to achieve an ideal system for hardware protection for ground vehicle applications, including examples, but not explicitly detailing implementation requirements.
Process	IEC 61511	Proposed the Cybersecurity lifecycle to be integrated with the process safety management (ISA TR 840009 (DRAFT)).
Functional Safety (General)	IEC 61508	IEC 61508-2:2010. Security is addressed but in an informative way. Malevolent and unauthorized actions have to be addressed during the hazard and risk analysis. If a security threat is seen as being reasonably foreseeable, then a security threats analysis should be carried out and if security threats have been identified then a vulnerability analysis should be

		undertaken in order to specify security requirements. IEC 61508-3 preparation: Security aware safety guidelines (Nov. 2014)
Machinery	IEC TC44	Safety of machinery, electro-technical aspects (first considerations). Separation of safety and security already at requirements level. OEM the only responsible, not the machinery manufacturer.
Cybersecurity (General)	IEC 62443	Industrial automation and control systems security/ Network and system security for industrial-process measurement and control. **Basis of security for safety**

These approaches also indicate the increased relevance of safety and security engineering. Several recent standards promote safety and security co-engineering. For example, for the automotive domain SAE J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems provides guidance for safety and security co-engineering. J3061 standard also defines safety and security interaction point approach corresponding to the automotive functional safety standard ISO 26262. For the industrial control domain, IEC 62443 give guidance on how security threats for safety-critical control systems shall be treated. The standard identifies zones and conduits of different level and elaborates on appropriate security measures, taking into account safety risks for the determination of security levels. In addition, the newly formed IEC TC65 WG20 "Industrial-process measurement, control and automation– Framework to bridge the requirements for safety and security" is also working on the issue of safety and security co-engineering. This new working group is looking into standards for safety and security for industrial systems from the industrial and other domains to define an applicable framework for bridging safety and security. For the railway domain, a cybersecurity extension of the CENELEC standards EN 50128/29 and EN 50159 has been elaborated by the German Association of electro-technicians VDE, named DIN/VDE V 0831-104, "Electric signalling systems for railways – Part 104: IT Security Guideline based on IEC 62443". The standard uses concepts of IEC 62443 and demands an extension of the safety case to include security measures.

As a component of the engineering process, assurance cases are generally developed to support claims. ISO/IEC 15026-2 specified minimum requirements of the structure and contents of an assurance case. IUSO/IEC 15026-3 specifies the concept of integrity levels with corresponding integrity level requirements to be met. One important use of integrity level is to aid in assuring safety, economic, and security characteristics between suppliers and in acquiring a system or product. The Object Management Group (OMG) has standardized a meta-model for representing structured assurance cases, called the Structured Assurance Case Meta-model (OMG SACM).

On the commercial market, technology and engineering companies start to show interests in safety and security concerns. For example, Thales develops an in-house "Safe and Secure Computing Platform Engineering Process". It is not yet formalized and integrated in the Thales CHORUS Reference system. However, best practices are shared between the safety and security engineering teams.

In the following, domain-specific standards with dependability attributes in addition to safety and security are collected and analysed. Note that the purpose of the review is to gain an overview of the perspectives and approaches to dependability in different domains, in order to facilitate the development of multi-concern assurance concept and toolchain in the rest of the project.

4.1 Automotive domain

4.1.1 Assurance concerns

From a normative perspective, the automotive industry is currently mainly concerned with safety as described in ISO 26262. In addition to that, availability, reliability and performance are also considered. Unlike safety where a well-defined lifecycle exists, availability, reliability and performance are not treated in a standardized way. It should be remarked that the safety lifecycle also is adopted according to the needs of a company and tailored for each specific development.

In the last years security arises as a new concern, which is currently treated by each automotive company with varying rigor. Security is more established for V2X applications.

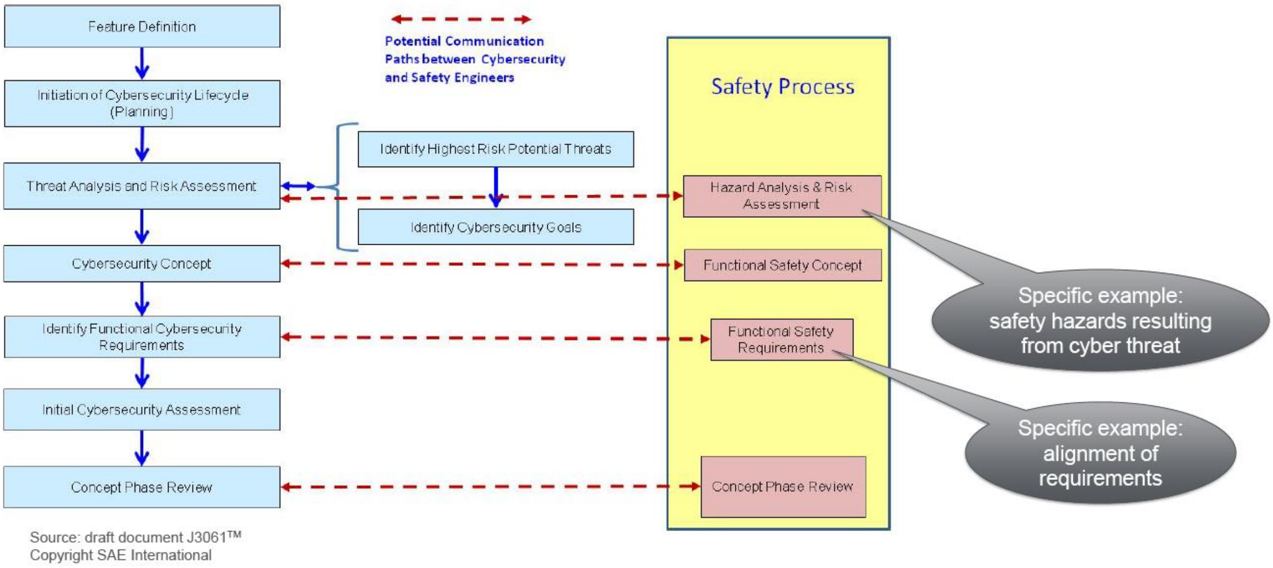
ISO 26262 prescribes a safety case, but only requires that the work products from the application of ISO 26262 are included, the structure and format are not as defined as in other sectors, e.g. railway domain.

The trend towards higher automation driving capabilities as well as higher connectivity poses a challenge in terms of safety and security concerns. The automotive industry has already begun of being aware of the impact that cybersecurity issues could have on safety concerns. Consequently, the Society of Automated Engineers (SAE)[86] published the world's first standard on automotive cybersecurity.

J3061 [87] details how cybersecurity assurance can be introduced in parallel with safety assurance or integrated into a common safety and cybersecurity product development process. Its main goals are the following:

- Facilitate the coordination of cybersecurity and safety by using a common process framework. The cybersecurity process framework is based on ISO 26262 process framework.
- Build security by design instead of adding it at the end of the development (as applies to system safety).
- Identify tools and methods to make the application of the process easier.
- Include cybersecurity activities along the whole product development lifecycle i.e. design, verification and validation, deployment, service and communication plans
- Provide a recommended practice that is goal-based rather than prescriptive
- It establishes a complete cybersecurity V Model Relationship between system, hardware and software development activities as seen in Figure 43.

In the section 2.2 Safety and Security Co-Engineering Background, different safety and security approaches have been pointed out. More specifically, unification and integration/harmonization solutions have been introduced. J3061 follows the second one, where potential communication paths must be set between cybersecurity and safety engineers at concept level.


Feature Definition
Initiation of Cybersecurity Lifecycle (Planning)
Threat Analysis and Risk Assessment
Cybersecurity Concept
Identify Functional Cybersecurity Requirements
Initial Cybersecurity Assessment
Concept Phase Review
Potential Communication Paths between Cybersecurity and Safety Engineers
Identify Highest Risk Potential Threats
Identify Cybersecurity Goals
Safety Process
Hazard Analysis & Risk Assessment
Functional Safety Concept
Functional Safety Requirements
Concept Phase Review
Specific example: safety hazards resulting from cyber threat
Specific example: alignment of requirements

Source: draft document J3061™
Copyright SAE International

Figure 42: Safety and Security alignment at concept level [89][90]

Besides, Figure 43 depicts the cybersecurity V model and the relationships between system, hardware and software development activities.

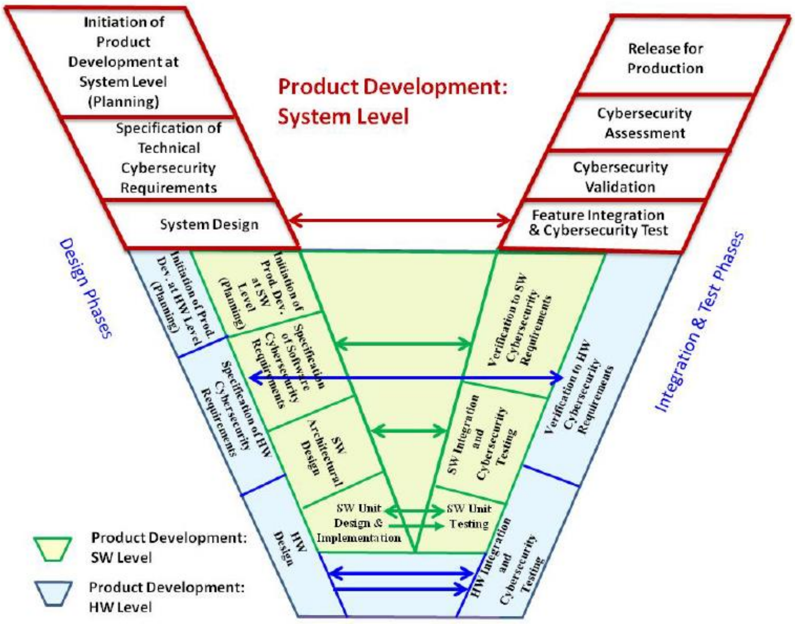

Initiation of Product Development at System Level (Planning)
Specification of Technical Cybersecurity Requirements
System Design
Product Development: System Level
Release for Production
Cybersecurity Assessment
Cybersecurity Validation
Feature Integration & Cybersecurity Test
Design Phases
Integration & Test Phases
Initiation of Prod. Dev. at HW Level (Planning)
Initiation of Prod. Dev. at SW Level (Planning)
Specification of Software Cybersecurity Requirements
Specification of HW Cybersecurity Requirements
SW Architectural Design
SW Unit Design & Implementation
HW Design
Verification to SW Cybersecurity Requirements
Verification to HW Cybersecurity Requirements
SW Integration and Cybersecurity Testing
SW Unit Testing
HW Integration and Cybersecurity Testing
Product Development: SW Level
Product Development: HW Level

Figure 43: Cybersecurity V model

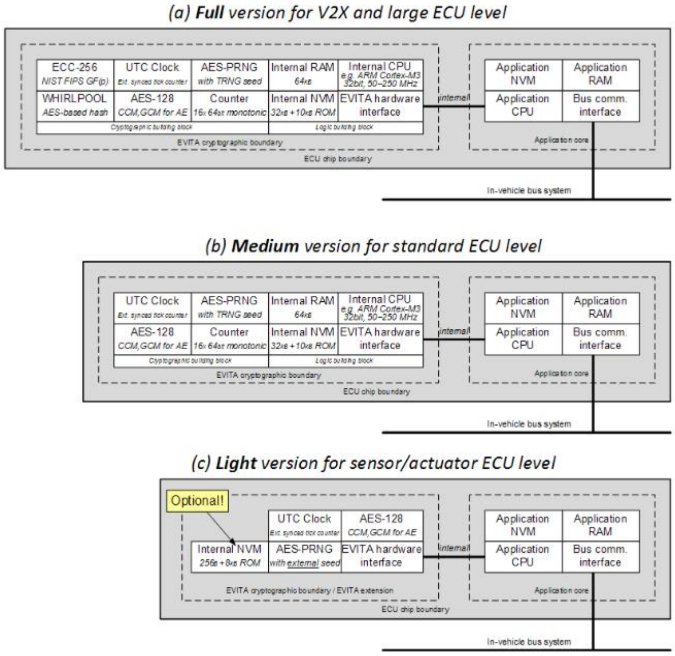
In few words, the process framework described in J3061 enables organizations to align its cybersecurity and safety activities. It also mentions specific methods to conduct Threat Analysis and Risk Assessment or for cybersecurity Testing Methods (Table 6).

Table 6: Safety/Security Analysis and Testing methods in terms of ISO 26262 and J3061

Hazard Analysis and Risk Assessment (HARA) and Safety Analysis Methods	Threat Analysis, Risk Assessment (TARA) and Vulnerability Methods	Safety Testing Methods	Cyber security Testing Methods
Brainstorming, checklists, quality history, FMEA and field studies to extract hazards FMEA, FTA, HAZOP, Markov Analysis	EVITA Method (E-safety Vehicle InTrusion protected Applications) EVITA Applied at the Feature Level using THROP (Threat and Operability Analysis) TVRA (Threats, Vulnerabilities and Risks (TVR) of a system to be analysed) OCTAVE (Operationalyy Critical Threat, Assest, and Vulnerability Evaluation) HEAVENS (HEAling Vulnerabilities to Enhance Software Security and Safety) Attack Trees Software Vulnerability Analysis	Fault Injection	Penetration testing, Fuzz testing, red teaming

The previous recommended practice is complemented by J3101 "Requirements for Hardware-Protected Security for Ground Vehicle Applications"[91]. It main purpose is to define a common set of requirements for security to be implemented in hardware for ground vehicles to facilitate security enhanced applications, developing expectations for necessary functionality to achieve an ideal system for hardware protection for ground vehicle applications, including examples, but not explicitly detailing implementation requirements /]. This way, the security privileged functions are separated from the applications of the ECU through hardware. In this line, the automotive industry also considers the proven industry standards, such as AES-128, ECC-256, SHE (Secure Hardware Extension), EVITA Hardware Security Module (HSM)[30] and Trusted Platform Module (TPM). Some of the most relevant ones are further explain below:

- **EVITA**: the European Project EVITA defined an architecture for secure on-board automotive networks.



(a) ***Full*** *version for V2X and large ECU level*

(b) ***Medium*** *version for standard ECU level*

(c) ***Light*** *version for sensor/actuator ECU level*

Figure 44: EVITA versions

- **<u>Trusted Platform Module (TPM):</u>** written by the Trusted Computing Group (TCG) and standardized as ISO/IEC 11889. It is designed for dedicated microprocessors that integrate cryptographic keys into devices. This group has recently released a TPM profile for secure automobile data and operation.
- **<u>Secure Hardware Extensions (SHE):</u>** From the German OEM consortium Hersteller Initiative Software (HIS). On-chip extensions providing both a set of cryptographic services to the application layer and key isolation.

In ISO 26262 Edition 2 the identification of communication channels between functional safety and related disciplines (e.g. cybersecurity) is expected to be addressed. This will include examples of potential interface points.

It is worth noting that within the different task forces from SAE, one of them relates to the definition of Automotive Cybersecurity Integrity Level (ACsIL) [92]. Different TARA methods are being reviewed and deciding on one or a tailored version of one. Besides, they are trying to set how to relate the ACSIL for safety-related threats to the ASIL from ISO 26262.

4.1.2 Applicable standards

The identified standards applicable to the automotive domain in the context of AMASS are described in the Table 7:

Table 7: Applicable standards to the automotive domain in the context of AMASS

Standard	Description	Status	Attributes treated
ISO 26262 "Road vehicles – Functional safety"	Functional safety standard for electrical, electronic and programmable electronic systems in production automobiles. In the current version only safety is considered, and the scope is restricted to passenger vehicles.	Published, V2 in preparation	Safety
SAE J3061	SAE Guidebook, published end of 2015,	Published	Security

	focusing on automotive cybersecurity. The lifecycle process and work products are modelled after the ISO 26262 structures.		
ISO/SAE WD 21434 “Automotive Cybersecurity”	New ISO/SAE standard in development, taking up the work and experience from SAE J3061 and developing an automotive cybersecurity standard.	Working draft, publication expected in 2020	Security
NWIP “Road vehicles – Safety of the Intended Functionality”	With automated driving safety depends increasingly also on the nominal behaviour of systems, e.g. the detection rate of sensors under faultless operation conditions. This new work item tries to define requirements and state of the art for such concerns.	In development	Safety, Performance, Reliability, Availability,
ETSI ITS Standards	ETSI Intelligent Transport Systems (ITS) standards are umbrella standards that include telematics and all types of communications in vehicles, between vehicles (e.g. car-to-car), and between vehicles and fixed locations (e.g. car-to-infrastructure).		Security
ISO/IEC 27001	The Information Security Management standard is widely adopted. There are attempts to apply a subset of the standard to address IT security in the automotive domain.	Published	Security
ISO/IEC 15408	Common criteria for security assurance evaluation is partially applied to electronic systems used in automotive domain, e.g. a subset of a ECU, an operating system, or smart card. Due to the extensive effort incurred in implementing the standard, its usage in automotive domain is envisaged to be confined to small scale sub-systems.	Published	Security
IEC 61508: Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-related Systems	Basic functional safety standard applicable to all kinds of industry. It defines functional safety as: “part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.”	Published	Safety
SAE J3101: Requirements for Hardware-Protected Security for Ground Vehicle Applications	Define a common set of requirements for security to be implemented in hardware fo r ground vehicles to facilitate security enhance d applications, developing expectations for necessary functionality to achieve an ideal system for hardware protection for ground vehicle applications, including examples, but not explicitly detailing implementation requirements.	Work in progress	Security

Most of the standards mentioned above are available to the public; in particular the new cybersecurity working draft is not. Therefore, some in-advance information in given in the following.

Since recently, the new standard on Automotive Cybersecurity Engineering – ISO/SAE 21434 - has been under development in ISO TC22 SC32 WG11. According to the agreed schedule, a WD is to be completed in April, the CD in September 2018. In March 2019 the DIS shall be circulated, the FDIS is scheduled for December 2019, and finally the Publication is expected in May 2020.

The standard is intended to be more prescriptive and rigorous than the above-mentioned SAE-J3061, which is rather a guidebook. Essential is in the new standard draft that, compared to ISO 26262, the risk management lifecycle is extended to include operation and production phases, since security needs to be maintained during the entire operation of the system. The standard is organized in four groups, namely Risk Management, Product Development, Operation/Maintenance & other processes, and finally a Process Overview with Interdependencies.

According to the current working draft, Cybersecurity Assurance Levels (CAL) are derived for individual items during the concept phase. This can be done based on impact and likelihood of attacks. Each CAL specifies a set of goal-based assurance requirements on the engineering process in terms of levels of rigour and is related as well to Security functional requirements. Details of the methodology are currently under discussion in the working group. A description of the contents of the current ISO/SAE WD 21434 can be found in AMASS D8.10 Standardization Plan, which was published at the end of March 2018.

4.2 Railway domain

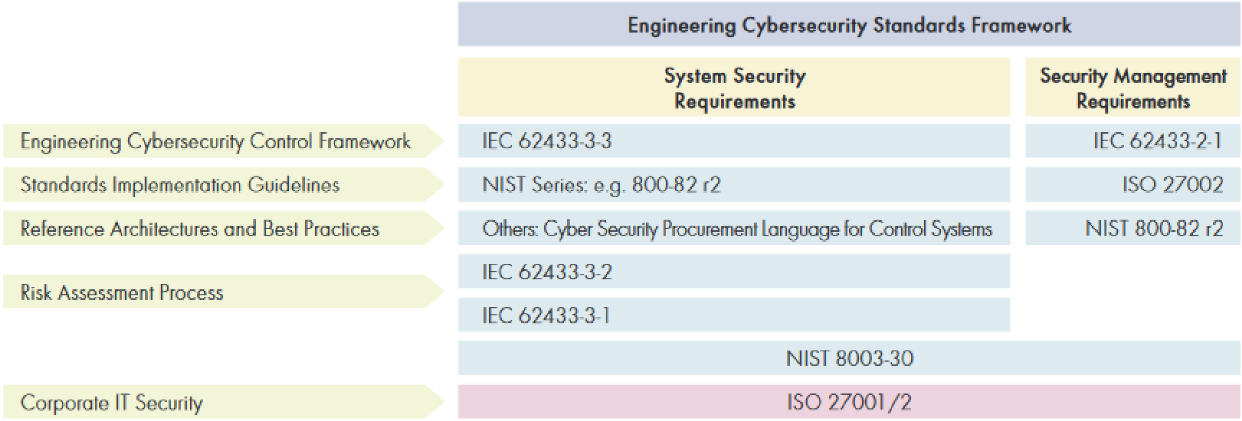
4.2.1 Assurance concerns

The railway domain has traditionally a very high safety culture and is today using detailed domain specific standards issued by CENELEC. These standards (EN 50126, 50128, 50129 and 50129) aim at functional safety and provide detailed concepts for RAMS. This acronym comprises the quality attributes reliability, availability, maintainability and safety, but security was widely disregarded. Summarizing we may state that the railway segment has already established multi-concern assurance. The interplay between hardware or system reliability and availability with safety is - for the higher SIL levels 3 and 4 - treated by quantitative fault tree analyses. The norms prescribe the calculation of a tolerable hazard rate and detailed considerations on how reliable a safety function must be to come with safety requirements. For security is, however, there is only to a minimal guidance.

This almost total exclusion of security considerations was justified as railway systems were not accessible from outside and strictly separated from public networks. But by and by systems are getting more networked.

In order to be able to offer excellent rail service, rail operators make widespread use of information technologies and automated computer systems. These systems are under a constant and increasing threat. The protective measures against cyberattacks in the railway sector are not yet fully developed [92]. There is often a lack of awareness with respect to new risks. Railways in particular, with their highly developed security philosophies, do not take the risks into account because they are convinced of the high level of security of their systems. This is, however, the result of a misunderstanding: Security in the railway industry is operational safety. But cybersecurity is about protecting information systems against theft or damage, ergo defending them against external and internal attacks and risks, in particular as a result of criminality. That is a significant difference. Indeed, modern railway systems have highly specific control and safety technology infrastructures with signalling technology, e. g. electronic interlockings, radio-based signalling systems such as the European Train Control System and the specially developed mobile radio standard GSM-R, to which cyber criminals cannot simply gain access.

However, these complex technologies only work because they are integrated into the Internet-based data communication system and run via corresponding servers like any other application. There, experts see the risk of attacks or interventions from unauthorised parties. To provide a systematic way to address information security, industry associations and standardization bodies have developed a set of international standards on the topic of security. Some of them are presented in figure below [92]:



	Engineering Cybersecurity Standards Framework	
	System Security Requirements	Security Management Requirements
Engineering Cybersecurity Control Framework	IEC 62433-3-3	IEC 62433-2-1
Standards Implementation Guidelines	NIST Series: e.g. 800-82 r2	ISO 27002
Reference Architectures and Best Practices	Others: Cyber Security Procurement Language for Control Systems	NIST 800-82 r2
Risk Assessment Process	IEC 62433-3-2	
	IEC 62433-3-1	
	NIST 8003-30	
Corporate IT Security	ISO 27001/2	

Figure 45: Engineering Cybersecurity Standards Framework

Recently, the German association for electro-technics, electronics and information technology "VDE" developed a German security standard, complementary to the CENELEC standards for functional safety, namely DIN VDE V 0831-104 " Electric signalling systems for railways – Part 104: IT Security Guideline based on IEC 62443". Based on the concepts of IEC 62443 for security in industrial communication networks, DIN VDE V 0831-104 extends the protection "against unauthorized access" as required by CENELEC EN 50129 by "unauthorized access by means of technical systems".

4.2.2 Applicable standards

The identified standards applicable to the railway domain in the context of AMASS are described in the Table 8.

Table 8: Applicable standards to the railway domain in the context of AMASS

Standards	Description	Status	Attributes treated
IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems	Basic functional safety standard applicable to all kinds of industry. It defines functional safety as: “part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.”	Published	Safety
EN 50126: Railway applications - The specification and demonstration of	Defines the terms of RAMS, their interaction and a process based on the system lifecycle for managing RAMS.	Published	Safety Availability Reliability Maintainability

Reliability, Availability, Maintainability and Safety (RAMS)			
EN 50128: Railway applications - Communication, signalling and processing systems	Specifies procedures and technical requirements for the development of programmable electronic systems for usage in railway control and protection applications, aimed at usage in any area where there are safety implications.	Published	Safety Availability Reliability Maintainability
EN 50129: Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling	Specifies those lifecycle activities which shall be completed before the acceptance stage, followed by additional planned activities to be carried out after the acceptance stage.	Published	Safety Availability Reliability Maintainability
EN 50159: Railways, Safety related communications	Gives the basic requirements needed to achieve safety-related communication between safety-related equipment connected to the transmission system. This European Standard is applicable to the safety requirement specification of the safety-related equipment connected to the transmission system, in order to obtain the allocated safety integrity requirements.	Published	Safety
Pre-standard DIN/VDE_V_0831-104, Electric signalling systems for railways – Part 104: IT Security Guideline based on IEC 62443.	The German pre-standards is based on the concepts of IEC 62443 for security in industrial communication networks, DIN VDE V 0831-104 extends the protection "against unauthorized access" as required by CENELEC EN 50129 by "unauthorized access by means of technical systems". Published 10-2015	Published	Security

4.3 Aerospace domain (including ATM)

4.3.1 Assurance concerns

❖ **Security engineering:**

The Airworthiness Security Process Specification (EUROCAE ED-202) is a resource for certification authorities and the aviation industry for developing or modifying aircraft systems and equipment when there is the possibility of adversely affecting the safety of flight from human action involving information or information system interfaces. It specifies data requirements and compliance objectives of an airworthiness security process, presented using a set of representative generic activities for managing data and objectives.

❖ **Safety engineering:**

The ED79A/ARP4754A addresses the total life cycle for Systems that implement aircraft level functions. It excludes specific coverage of detailed Systems, software and hardware design processes beyond those of significance in establishing the safety of the implemented system. More detailed coverage of the software aspects of design are dealt with in EUROCAE/RTCA document ED-12B/DO-178B. Coverage of complex hardware aspects of design are dealt with in ED80/DO254. Methodologies for safety assessment processes are outlined in SAE document ARP4761.

In ED79A/ARP4754A, the process includes validating requirements, and verifying that requirements are met, together with the necessary configuration management and process assurance activities. As development assurance level assignments are dependent on classification of Failure Conditions, the safety analysis process is used in conjunction with the development assurance process to identify Failure Conditions and severity classifications which are used to derive the level of rigor required for development. The level of validation and verification rigor is determined by the function development assurance level(s) for the aircraft or system (FDAL) and item development assurance level(s) for the item (IDAL).

The application of independence is also dependent upon the development assurance level and is commensurate with the development assurance level. Two tables identify the validation and verification methods and data as a function of the allocated development assurance level A-E. According to the Assurance Level, methods and data could be either recommended for certification, as negotiated for certification, or simply not required for certification for level E.

The guidelines and methods provided in ED135/ARP4761 are intended to be used in conjunction with other applicable guidance materials, including ARP4754, ED12B/DO178B, ED80/DO254, and with the advisory material associated with CFR/JAR Parts 25.1309 and 23.1309.

A process is needed, which establishes levels of confidence that development errors that can cause or contribute to identify Failure Conditions have been minimized with an appropriate level of rigor. This henceforth is referred to as the Development Assurance process. The determination of the classification of the Failure Condition Effects is accomplished by analysing accident/incident data, reviewing regulatory guidance material, using previous design experience, and consulting with flight crews, if applicable. The depth of analysis undertaken depends on the Development Assurance Level (DAL) associated with a particular system. The DAL is allocated depending on the potential criticality and risk associated with a system failure. The classifications are: Catastrophic (DAL A), Severe-Major/Hazardous (DAL B), Major (DAL C), Minor (DAL D) and No safety effect (DAL E).

❖ **Multi-concern assurance:**

Multi-concern assurance in the avionic domains (including ATM in AMASS) developed during the standardisation process [1]. Modern Aircrafts are highly connected systems, communicating and interacting with other Aircrafts, the Airport Air Traffic Control (ATC), Ait Traffic Management (ATM) and specific components even communicate lifetime data to the manufacturer for maintenance purposes. In the absence of comprehensive rules and guidance for how cyber-security affects safety Aircraft type certification were used. FAA, Transport Canada and EASA used an adhoc process in the form of ‘Special Conditions’ to address specific security concerns for specific aircraft model. In 2006 RTCA SC-216, Aeronautical Systems Security was formed as a Special Committee, which should address the topic of cybersecurity and safety in collaboration with EUROCAE WG-72. In 2014/15 three new documents, addressing this topic were released. During the development of the standards the safety&security interaction model developed. It started with security as a sub concern in the safety process, moved to separate security and safety processes, and finally to safety and security processes which both influence the system development, but not directly each other.

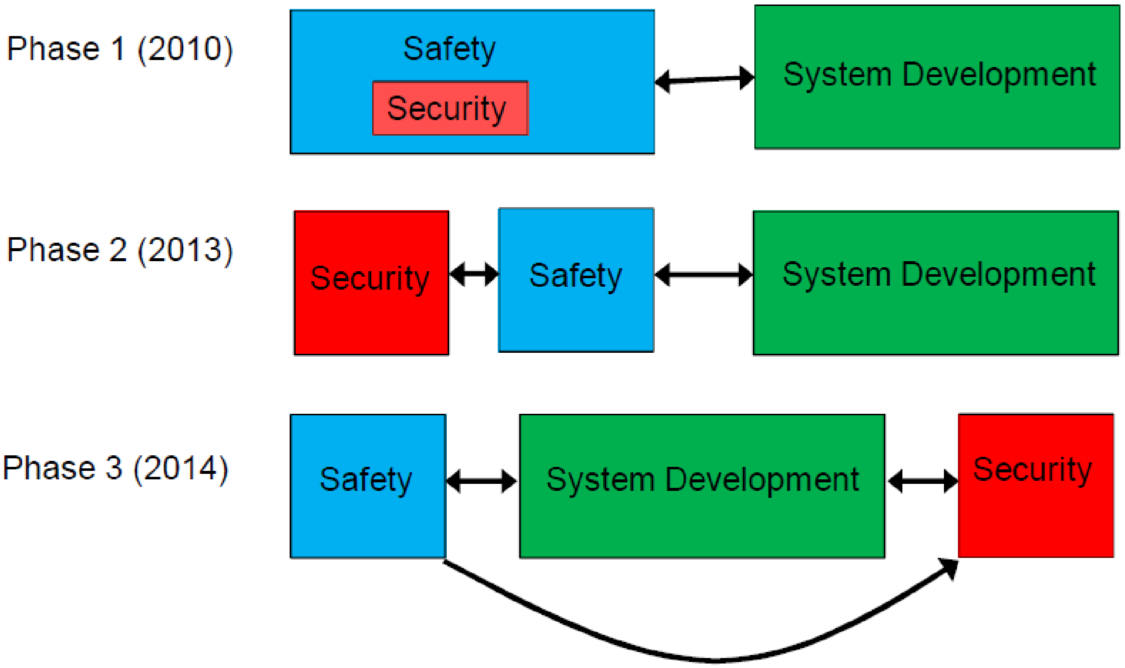


Figure 46: Safety and security alignment process in the avionics domain

4.3.2 Applicable standards

The identified standards applicable to the avionics domain in the context of AMASS are described in the Table 9.

Table 9: Applicable standards to the avionics domain in the context of AMASS

Standards	Description	Status	Attributes treated
IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems	Basic functional safety standard applicable to all kinds of industry. It defines functional safety as: “part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.”	Published	Safety
RTCA DO-278A: Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems	Provides guidelines for the assurance of software contained in non-airborne CNS/ATM systems and provides recommendations for the production of that software commensurate with a level of confidence in safety. DO-278A defines a set of objectives recommended to establish assurance that the developed CNS/ATM software has the integrity needed for use in a safety-related application.	Published	Safety
RTCA DO-178B/C: Software Considerations	Provides guidance and recommendations for the production of software for airborne	Published	Safety Robustness

in Airborne Systems and Equipment Certification	systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements		
RTCA DO- 326A: Cyber-Security and Safety for Aircraft and Aircraft Systems	Provides guidance for aircraft certification to handle the threat of intentional unauthorized electronic interaction to aircraft safety. It adds data requirements and compliance objectives, as organized by generic activities for aircraft development and certification	Published	Safety Security Robustness
RTCA DO-355: Information Security Guidance for Continuing Airworthiness	It deals with the activities that need to be performed in operation and maintenance of the aircraft related to information security threats. It also supports harmonizing security guidance documents among Design Approval Holders (DAH), which is deemed beneficial to DAHs, operators and civil aviation authorities. It is a companion document to DO-326A that supports security in the development and modification part of the airworthiness process.	Published	Safety Security
RTCA DO-356: Airworthiness Security Methods and Considerations	The methods and considerations of this document address the assessment of the acceptability of the airworthiness security risk and the design and verification of the airworthiness security attributes as related to system safety and airworthiness. It provides guidance for accomplishing the activities identified in DO-326A in the areas of Security Risk Assessment and Effectiveness Assurance. It also provides specific methods for Security Risk Analysis and Network Security Domains.	Published	Safety Security Robustness
RTCA DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations	Contains guidance for Integrated Modular Avionics (IMA) developers, application developers, integrators, certification applicants, and those involved in the approval and continued airworthiness of IMA systems in civil certification projects. It is focused on IMA-specific aspects of design assurance.	Published	Safety Security Performance Availability Reliability Maintainability Robustness
RTCA DO-330: Software Tool Qualification Considerations	Provides software tool qualification guidance. Additionally, clarification material is provided in the form of Frequently Asked Questions (FAQs).	Published	Safety Robustness
RTCA DO-331: Model-Based Development and Verification Supplement to DO-178C and DO-278A	This supplement contains modifications and additions to DO- 178C and DO-278A objectives, activities, explanatory text and software life cycle data that should be addressed when model-based development and verification are used as part of the software life cycle. This includes the artifacts that would be expressed using models and the verification evidence that could be derived	Published	Safety Robustness

	from them.		
RTCA DO-332: Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A	This supplement identifies the additions, modifications and deletions to DO-178C and DO-278A objectives when object-oriented technology or related techniques are used as part of the software development life cycle and additional guidance is required. This supplement, in conjunction with DO-178C, is intended to provide a common framework for the evaluation and acceptance of object-oriented technology (OOT) and related techniques (RT)-based systems	Published	Safety Robustness
RTCA DO-333: Formal Methods Supplement to DO-178C and DO-278A	This supplement identifies the additions, modifications and substitutions to DO-178C and DO-278A objectives when formal methods are used as part of a software life cycle, and the additional guidance required. It discusses those aspects of airworthiness certification that pertain to the production of software, using formal methods for systems approved using DO-178C.	Published	Safety Robustness
SAE-ARP 4754/4754A: Guidelines for development of civil aircraft and systems	Guideline for development of civil aircraft and systems with an emphasis on safety aspects. Revision A is a substantial rewrite of the document which describes the safety process as a part of an Integrated Development Process. A significant new section is devoted to the process of determining Development Assurance Level (DAL) which determines the rigor of complex hardware and software development and verification activities.	Published	Safety
RTCA DO-254: Design assurance guidance for airborne electronic hardware	This document is intended to help aircraft manufacturers and the suppliers of aircraft electronic systems assure that electronic airborne equipment safely performs its intended function. The document identifies design life cycle processes for hardware that includes line replaceable units, circuit board assemblies, application specific integrated circuits (ASICs), programmable logic devices, etc. It also characterizes the objective of the design life cycle processes and offers a means of complying with certification requirements.	Published	Safety Performance Reliability
ARP 4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment	Recommended Practice defines a process for using common modelling techniques to assess the safety of a system being put together.	Published	Safety

4.4 Space domain

4.4.1 Assurance concerns

❖ **Security engineering:**

Currently there is no global specific management standard to be applicable to security engineering in space.

- However, a number of standards address the security of space transmissions, e.g.:
- The Consultative Committee for Space Data Systems (CCSDS) Space Data Link Security (SDLS) protocol provides security services to the existing CCSDS family of Space Data Link (SDL) protocols: Telecommand (TC), Telemetry (TM) and Advanced Orbiting Systems (AOS),
- European Space Agency (ESA) Procedures Standards and Specifications (PSS) / European Cooperation for Space Standardization (ECSS) space data link standard with encryption.

❖ **Safety engineering:**

The ECSS safety (Q-ST-40C) and dependability (QST-30C) standards introduce a 4-level scale for categorizing systems, functions and hardware and software components implementing them, based on a ranking of severity of consequences of their potential failures.

- At system level, the allocated criticality category impact is twofold: Generic product safety requirements with direct impact on the design;
- Process safety requirements with direct impact on the activities to perform.

System level product safety requirements:

The ECSS standards do not set requirements in terms of maximum probability of occurrence for the events in the various categories. However, they impose a minimum number of independent faults for any combination that could lead to a failure in the most two severe categories: no combination of two independent faults (resp. no single (or common mode) fault) can induce catastrophic (resp. critical) consequences. This has a direct impact on the level of redundancy and diversification to implement in the architecture of the system.

System level process safety requirements

The ECSS standards state rules applicable to the safety and the dependability programs, roles and responsibilities, safety and dependability engineering, analysis and verification, and their articulation in the system life cycle along the various phases of a space program from phase 0 (mission analysis) to phase E (disposal). They also require that the way and rigor to implement these rules must be adapted to the category, but without provided guidance on these adaptations, left to be negotiated and agreed for each project.

Indeed, the safety standard (ECSS-Q-QST-40C) states also specific rules applicable to “safety critical systems”, corresponding to the most two demanding categories. However, these two categories are the only ones corresponding to significant safety effects of potential failures, whereas the other two categories are indeed a subdivision, from mission perspective and dependability, of a single safety category corresponding to “minor or no safety effect”.

Regarding safety and security interdependencies, the section 5.3 of ECSS safety (Q-ST-40C) depicts: “the implementation of safety requirements shall not be compromised by other requirements. NOTE For example: security requirements”.

4.4.1.1 Applicable standards

The identified standard applicable to the space domain in the context of AMASS is described in the Table 10.

Table 10: Applicable standard to the Space domain in the context of AMASS

Standards	Description	Status	Attributes treated
ECSS-Q-30	This Standard defines the dependability assurance programme and the dependability requirements for space systems. Dependability assurance is a continuous and iterative process throughout the project life cycle.	Published	Dependability
ECSS-Q-30-02	This Standard defines the principles and requirements to be adhered to with regard to failure modes, effects (and criticality) analysis (FMEA/FMECA) implementations in all elements of space projects in order to meet the mission performance requirements as well as the dependability and safety objectives, taking into account the environmental conditions. It defines the requirements and procedures for performing a FMEA/FMECA.	Published	Dependability FMEA/FMECA
ECSS-Q-40	This Standard defines the safety programme and the technical safety requirements that are implemented in order to comply with the ECSS safety policy as defined in ECSS-Q-00. It is intended to protect flight and ground personnel, the launch vehicle, associated payloads, ground support equipment, the general public, public and private property, and the environment from hazards associated with European space systems.	Published	Safety
ECSS-Q-40-02	This Standard details the hazard analysis requirements of ECSS-Q-ST-40. It defines the principles, process, implementation, and requirements of hazard analysis.	Published	Safety Hazard Analysis
ECSS-Q-40-12	This Standard defines requirements for the performance of Fault Tree Analysis (FTA) on space projects and incorporates the IEC 61025 standard into the ECSS system.	Published	Fault Tree Analysis
ECSS-Q80	This Standard defines a set of software product assurance requirements to be used for the development and maintenance of software for space systems. Space systems include manned and unmanned spacecraft, launchers, payloads, experiments and their associated ground equipment and facilities. Software includes the software component of firmware. This Standard also applies to the	Published	SW Product Assurance

	development of non-deliverable software which affects the quality of the deliverable product or service provided by a space system, if the service is implemented by software.		

4.5 Industrial automation domain

4.5.1 Assurance concerns

For the industrial automation domain, the main concerns refer to safety, security and availability. Respect to safety aspects the reference standard is IEC 61508. This standard address the functional safety of programmable electronic systems and it is well established in the industrial process control and automation industry. In the last years, the systems of this domain are exposed to cyber-attacks. In this sense, the cyber security is a principal concern. Standards such as: IEC 62351 and IEEE 1686 address the security mechanisms for the Industrial Automation Control System (IACS). These standards cover the cyber security of the electrical infrastructure in several aspects: access control, communications, security evens register, etc. Related to certification, the standard IEC 62443 defines requirements and procedures for implementing electronically secure automation and control systems and security practices, and assessing electronic security performance.

The case study CS1: Industrial Automation Control System (IACS) will address safety, security and availability aspects taking into account the standards IEC 61508, IEC 62351 and IEEE 1686.

4.5.2 Applicable standards

The identified standards applicable to the industrial automation domain in the context of AMASS are described in the Table 11.

Table 11: Applicable standards to the industrial automation domain in the context of AMASS

Standards	Description	Status	Attributes treated
IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems	Basic functional safety standard applicable to all kinds of industry. It defines functional safety as: “part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.”	Published	Safety
ISO/TS 15066:2016: Safety requirements for collaborative industrial robot systems and the work environment	Specifies safety requirements for collaborative industrial robot systems and the work environment, and supplements the requirements and guidance on collaborative industrial robot operation given in ISO 10218-1 and ISO 10218-2. It applies to industrial robot systems as described in ISO 10218-1 and ISO 10218-2. It does not apply to non-industrial robots, although the safety	Published	Safety

	principles presented can be useful to other areas of robotics.		
ISO 10218-1:2011: Robots and robotic devices -- Safety requirements for industrial robots	Specifies requirements and guidelines for the inherent safe design, protective measures and information for use of industrial robots. It describes basic hazards associated with robots and provides requirements to eliminate, or adequately reduce, the risks associated with these hazards.	Published	Safety
IEC 62443: Industrial communication networks -Security for industrial automation and control systems	Series of standards, technical reports, and related information that define procedures for implementing electronically secure Industrial Automation and Control Systems (IACS). This guidance applies to end-users (i.e. asset owner), system integrators, security practitioners, and control systems manufacturers responsible for manufacturing, designing, implementing, or managing industrial automation and control systems.	Published	Security
IEC TC65 WG20	In addition to IEC62443, IEC TC65 WG20 “Industrial-process measurement, control and automation– Framework to bridge the requirements for safety and security” is also working on the issue of safety and security co-engineering. This new working group is looking into standards for safety and security for industrial systems from the industrial and other domains to define an applicable framework for bridging safety and security.	In developme nt	Safety Security
IEEE 1686: Standard for Intelligent Electronic Devices Cyber Security Capabilities	Defines the functions and features to be provided in intelligent electronic devices (IEDs) to accommodate critical infrastructure protection programs. It also addresses Security regarding the access, operation, configuration, firmware revision and data retrieval from an IED.	Published	Security
IEC 62351: Information Security for Power System Control Operations	Defines cyber security aspects for IACS in the electrical substation domain.	Published	Security

5. Consolidation and Way Forward

In the previous sections, we have presented the results of the analysis of the state of the art and of the state of the practice on multi-concern assurance. It is the output of the task 4.1, which falls within the scope of the Scientific and Technical Objective 2 (STO2), which focuses on Multi-concern Assurance. We now synthesise and consolidate the results in order to point out possible gaps between the state of the art and the stat of the practice and to draw the way forward to fulfil the objectives of AMASS WP4.

In the AMASS project, we aim to exploit the existing OPENCOSS and SafeCer approaches and extend them to provide a tool-supported methodology for the development of assurance cases which address multiple system characteristics.

As it is shown in the state of the art, the safety engineering community is more and more engaged in multi-concern engineering and, more specifically, safety and security co-engineering. In fact, the increasing amount of cyber-attacks around the world demonstrates that safety-critical systems are not that safe as the safety engineering community aim, if those critical systems are not secure enough. Following this significant raising interest for jointly addressing safety and security and for security co-engineering by safety community, there was a multiplication of safety standards updates that include further dependability attributes concerns, including security.

Dependability Assurance

The work related to dependability assurance has been reviewed in section 3.1. It was important to focus, at a first time, on dependability as a set of attributes as presented in [1][2][3][4].

- Co-design: A couple of approaches are presented in addition to the MAFTIA project [7] that investigates the *tolerance paradigm* for security systematically. They aim at proposing an integrated architecture built on this paradigm, and realising a concrete design that can be used to support the dependability applications. A couple of approaches are presented, e.g. the work presented in [6] that introduces a new method to build safety-critical systems and ensure their safety and security requirements. The approach proposes patterns for the specification of safe and secure systems.
    - Co-analysis: A first approach [3] highlights the close interactions between fault removal and fault forecasting, and motivates their gathering into dependability and security analysis, it aims at reaching confidence in the ability to deliver a service that can be trusted. A new approach [8] to integrated security and dependability evaluation, which is based on stochastic modelling techniques, it aims to provide operational measures of the trustworthiness of a system, regardless whether the underlying failure cause is intentional or not. Another work [9] gives results of a web services dependability analysis using standardized FMEA technique and its proposed modification IMEA technique. Obtained results of the FMEA-technique application were used for determining the necessary means of error recovery, fault prevention, fault-tolerance ensuring and fault removal.
    - Verification and validation: The work presented in [3] introduces the means for the achievement of dependability and security such as Fault removal during the development phase or the using phase and Fault forecasting.
    - Certification: Several approaches were presented in the state of the art w.r.t dependability certification. For example, how the modularity has been introduced into dependability cases in order to achieve the challenge of complexity and length of cases, this approach [81] highlights the advantages of using modular safety cases. There is also the IAWG (Industrial Avionics Working Group) [81], that proposed to take advantage of the GSN graphical notation within the argument contract as it provides more expressiveness and clarity than the tabular approach. Ruiz in [82] went further with the formalization of the assurance contract. In this way, she defined a list of structured

expressions to be used within the assertions. Another important aspect that constitutes one of the AMASS aims, is the one related to the use of contracts to structure the dependability case. In this aspect, thanks to the use of modular argumentation (as in OPENCOSS), diverse benefits arise: high cohesion, low coupling (associations defined at module level and not at goal level), and defined and documented interfaces and information hiding. A couple of projects tackling dependability certification are presented. For example, the SafSec project that focuses on dependable systems and aims to reduce the cost and effort of safety certification and security accreditation for future military avionics systems and in-service upgrades. The SafSec Standard [13] helps to achieve the certifications with the minimum of duplicated work and the maximum of reuse of evidence between the different certifiers. The OPENCOSS project also that defines an argumentation metamodel [80], which is based on the assurance case concept. Each assurance case can be composed of other assurance cases on its turn. Argumentations are keystones, and therefore, OPENCOSS defined its own metamodel for defining argumentations. This metamodel is based on SACM [79] from OMG, but with the particularity that is support the argument pattern concept.

Safety and Security Assurance

It is important to focus, in the state of the art on safety and security assurance, on the integration of safety and security. So, the work related to this topic was reviewed in the section 3.2.

- ❖ Co-design: Regarding safety and security co-design, several approaches and related projects have been presented, e.g. in contract-based design, where the component interfaces are enriched with formal specification of assumptions and guarantees. These are properties specified in terms of the components' input/output. When the components have different concerns such as safety and security, their contracts must formalize assumptions and guarantees related to multiple concerns. There is also the Multiple Independent Levels of Security (MILS) approach developed as part of the Multiple Independent Levels of Security / Safety, which is characterized by a two-level approach to secure system design: (1) at the policy level, a decomposition to a virtual architecture is performed while identifying the trusted components, the local policies and the communications channels; (2) at the resource sharing level, implementation of components is considered, which includes the allocation of components to shared physical resources. Following this approach, a formal Contract-Based specification and analysis of safety and security contracts was developed in the D-MILS project. The D-MILS concept extends the capacity of MILS to implement a single unified policy architecture to a network of separation kernels. Another work presented in [25] introduces SysML-Sec, a SysML-based Model-Driven Engineering environment aimed at fostering the collaboration between system designers and security experts at all methodological stages of the development of an embedded system. The HEAVENS project [26] is also presented, in order to investigate in AMASS, whether it is possible to use the HEAVENS security model for threat analysis and risk assessment when creating the security aspect of multi-concern assurance cases that, as this model is mentioned in SAE J3061 standard for cybersecurity in the automotive domain.
- ❖ Co-analysis: Diverse works and projects enhancing methodologies and approaches commonly known and used in safety and security domains were presented. For example, work related to model based safety analysis, security analysis or safety and security co-analysis. There are also approaches based on UML to co-assess safety and security [43], or an attack tree analysis based on an extended Fault tree to capture malicious risks [39] [41]. Several projects elaborated the support for Safety and Security Co-Analysis, such as: SESAMO, CONCERTO and MERgE (see section 3.2.2).
- ❖ Verification and validation: Some approaches were investigated such as, the TURTLE UML profile and the open source toolkit TTool that define a formal modelling and verification framework for communicating embedded systems design [71]. There is also the AVATAR real-time extension of SysML for formal verification, supported by the TTool open-source toolkit. Another approach was

presented in the [74] based on Alloy to formally model and assess a system architecture with respect to safety and security requirements.

- ❖ Certification: With respect to safety and security certification, a couple of projects were presented. For example, the OPENCOSS and the SQUALE (Security, Safety and Quality Evaluation for Dependable Systems) projects are introduced, which aim to develop assessment criteria for obtaining confidence that a system will achieve its dependability objectives [56], [57].

This deliverable has presented initial, yet large and diverse, results from the state of the art and the state of the practice of multi-concern assurance. As listed in Table 5 (Section 4), this deliverable also shows that several standards which promote Safety and Security Co-engineering are available or in preparation. The standard development towards more security-awareness is still at the beginning. However, there are established cybersecurity design and process principles used to ensure safe and secure cooperation of systems. These include in particular:

- **Defence in depth**. Multiple layers of defence are applied. Even if a layer of defence is breach, e.g. due to a zero-day-vulnerability the system will be resilient and prevent a cybersecurity breach.
- **Incorporate preventive, Detective and Recovery Controls.** To succeed in addressing today's sophisticated cybersecurity attacks, the security solution must incorporate strong preventive mechanisms but also the ability to detect and quickly recover from cybersecurity attacks without affecting safety and system availability.
- **Design patterns**. Use of proven design patterns and protocols, such as secure gateway, web application firewall or secure device like on-board multi-layer firewall and secure virtual private network (VPN) gateway which guarantees message authenticity using cryptographic key exchange and prevents unauthorized access. In addition to the security controls message sequencing, message integrity and safety control provide the last layer of protection against any authorized message or malicious software in the network. For example, ensure protocols are based on One-Channel-Safe principles which include internal cyclic redundancy check (CRC) checks (over and above the IP protocols), continuous communication with latency and timeout supervisions, and persistency and consistency checks which prevent 'insertion' of messages
- **Risk based Approach**. Subsystem requirements and design trade-off are based on cost benefit analysis from threat and risk assessments.

Embed safety and cybersecurity in the project Lifecycle. Any project involving a cyber-physical system should include cybersecurity and safety activities at every step. Whether it is at the bid process, design and planning phase, testing process, or a technology upgrade, ensure cybersecurity is involved. This will ensure the assets and the processes are following dedicated organization's security and safety policies and standards. D4.1 will be used as the main basis in task T4.2 to extend the existing CCL and SafeCer metamodels and vocabulary (and possibly S-TunExSPEM) for safety assurance to encompass concepts relating to the further assurance concerns. The task will also investigate the refinement of the compositional assurance approach developed in OPENCOSS to address the wider challenges for multi-concern assurance that have been identified for AMASS.

Abbreviations and Definitions

Definitions are common to the whole AMASS project and are given in the AMASS glossary.

Abbreviation	*Explanation*
AADL	Architecture Analysis and Design Language
AFRL	Air Force Research Laboratory
AIPP	ARTEMIS Innovation Pilot Project
API	Application Programming Interface
ARP	Aerospace Recommended Practice
ARTEMIS	ARTEMIS Industry Association is the association for actors in Embedded Intelligent Systems within Europe
ASIC	Application -Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
ATM	Air Traffic Management
BDMP	Boolean logic Driven Markov Processes
CACM	Common Assurance and Certification Metamodel
CCL	Common Certification Language
CD	Committee Draft (a development stage in standards preparation)
CENELEC	Comité Européen de Normalisation Électrotechnique (European Committee for Electrotechnical Standardization)
CHESSML	CHESS Modelling Language
CNS	Communication, Navigation and Surveillance
CPS	Cyber-Physical Systems
DAL	Design Assurance level
DARPA	Defense Advanced Research Projects Agency
DDA	Dependability Deviation Analysis
D-MILS	Distributed MILS (an FP7 STREP project)
DIN	Deutsches Institut für Normung
DIS	Draft International Standard (a development stage in standards preparation)
DKE	Deutsche Kommission Elektrotechnik Elektronik Informationstechnik
EAL	Evaluation Assurance Levels
EBIOS	Expression des Besoins et Identification des Objectifs de Sécurité (Expression of Needs and Identification of Security Objectives)
ECSEL	Electronic Components and Systems for European Leadership
ECU	Electronic Controller Unit
EMC^2	Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments
EN	European Norm
ETSI	European Telecommunications Standards Institute
EUC	Equipment Under Control
E/E/PE	Electrotechnical/Electronic/Programmable Electronic
FANDA	Factors, Analysis and Decisions Alternatives
FAQ	Frequently Asked Questions
FDIS	Final Draft International Standard (a development stage in standards preparation)
FIPS	Federal Information Processing Standard

FMEA	Failure Modes and Effects Analysis
F(I)MEA	Failure (Intrusion) Modes and Effects Analysis
FMVEA	Failure Modes, Vulnerabilities and Effect Analysis
FP7	Framework Programme 7 (a former research programme of the European Union)
FRS	Fragmentation-Redundancy-Scattering
FT	Fault Tolerance
FT	Fault Tree
FTA	Fault Tree Analysis
GEMS	Generic Error-Modelling System (a human error classification scheme by James Reason)
GSM-R	Global System for Mobile Communications – Railway
GSN	Goal Structured Notation
HACMS	High-Assurance Cyber Military Systems
HAZOP	HAZard and OPerability study
HEAVENS	HEAling Vulnerabilities to ENhance Software Security and Safety (Swedish Vinnova funded research proj.)
HW	Hardware
IACS	Industrial Automation Control System
IAWG	Industrial Avionics Working Group
ICS	Industrial Control Systems
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Devices
IEEE	Institute of Electrical and Electronics Engineers
IL	Impact Level
IMA	Integrated Modular Avionics
IMEA	Intrusion Modes and Effects Analysis
ISO	International Organization for Standardization
ITEA	Information Technology for European Advancement
ITS	Intelligent Transport Systems
JU	Joint Undertaking
MARTE	Modelling and Analysis of Real Time and Embedded systems
MBSA	Model-Based Safety Analysis
MLS	Multilevel security or multiple levels of security
MNS	MILS Networking System
MILS	Multiple Independent Levels of Security
MoD	Ministry of Defence
NuSMV	New Symbolic Model Verifier (a symbolic model checker tool for finite state systems)
NWIP	New Work Item Proposal (early proposal for a new standardization activity)
OCRA	Othello Contracts Refinement Analysis
OMG	Object Management Group
OOT	Object-Oriented Technology
OSLC	Open Services for Lifecycle Collaboration
PhD	Philosophiae Doctor (neolatin; = doctor of philosophy)
QM	Quality Management
RAMS	Reliability, Availability, Maintainability, Safety (and Security)
RCM	Reliability Configuration Model

RTCA	Radio Technical Commission for Aeronautics
RT	Real-Time
RT	Related Techniques
SACM	Structured Assurance Case Metamodel
SAE	Society of Automotive Engineers
SAL	Security Assurance Level
SCADA	Supervisory control and data acquisition
SDLC	(Microsoft) Secure Development Life Cycle
SEMA	Referential Framework for System (->Environment), Environment (->System), Malicious, Accidental
SFTM	Static Fault Tree Model
SIL	Safety Integrity Level
SL	Security Level
SoS	System of Systems
SOTA	State of the art
SQUALE	Security, Safety and Quality Evaluation for Dependable Systems
STO	Scientific and Technical Objective
STREP	Specially Targeted REsearch Programme
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
SW	Software
SysML	System Modelling Language
S&S	Safety and Security
TARA	Threat Analysis and Risk Assessment
TL	Threat Level
TOE	Target Of Evaluation
TOM	Trade-Off Method
TTE	Time-Triggered Ethernet
UML	Unified Modelling Language
UPPAAL	UPPsala university & AALborg univ. (tool environment for RT-system modeling, simulation & verification)
V&V	Verification and Validation
WD	Working Draft (a development stage in standards preparation)
WP	Work Package

References

[1] Jean-Claude Laprie - Dependability: Basic Concepts and Terminology [Book]. - Vienna : Springer, 1992. - Vol. 5

[2] Jonsson Erland and Olovsson Tomas On the Integration of Security and Dependability in Computer Systems // http://publications.lib.chalmers.se/records/fulltext/167782/local_167782.pdf

[3] Avizienis Algirdas et al. Basic concepts and taxonomy of dependable and secure computing - 2004. // http://www.nasa.gov/pdf/636745main_day_3-algirdas_avizienis.pdf

[4] Jonsson Erland Towards an integrated conceptual model of security and dependability [Conference] // First International Conference on Availability, Reliability and Security (ARES). - [s.l.] : IEEE, 2006. DOI: 10.1109/ARES.2006.138

[5] Nicol David M., Sanders William H. and Trivedi Kishor S. Model-based evaluation: from dependability to security [Journal] // IEEE Transactions on Dependable and Secure Computing. - [s.l.] : IEEE, 2004. - 1 : Vol. 1. - pp. 48-65. - DOI: 10.1109/TDSC.2004.11

[6] Delange Julien Security and dependability integration for the construction of critical middleware (in French: Intégration de la sécurité et de la sûreté de fonctionnement dans la construction d'intergiciels critiques) [Report]: PhD Thesis / Laboratoire Traitement et Communication de l'Information, UMR 5141 ; Département informatique et réseau. - Paris : Ecole Nationale Supérieure des Télécommunications (TELECOM ParisTech), 2010 - pastel-00006301

[7] Malicious-and Accidental-Fault Tolerance for Internet Applications [Online] // LAAS. - IST MAFTIA Project n°11583, 01 01 2000. - 25 09 2014. - http://webhost.laas.fr/TSF/cabernet/maftia/

[8] Sallhammar Karin, Helvik Bjarne E. and Knapskog Svein J. Towards a Stochastic Model for Integrated Security and Dependability Evaluation [Conference] // First International Conference on Availability, Reliability and Security. - Washington : IEEE, 2006. - DOI: 10.1109/ARES.2006.137

[9] Gorbenko Anatoliy et al. : F(I)MEA-technique of Web Services Analysis and Dependability Ensuring - DOI: 10.1007/11916246_8.

[10] Babeshko E., Kharchenko V. and Gorbenko A. Applying F(I)MEA-technique for SCADA-Based Industrial Control Systems Dependability Assessment and Ensuring - DOI: 10.1109/DepCoS-RELCOMEX.2008.23

[11] https://www.acuitas.com/sites/default/files/standards/DStan00_40P1I7_RM_Responsibilities.pdf

[12] P. Bishop, R. Bloomfield and S. Guerra, “Safety Justification Frameworks: Integrating Rule-Based, Goal-Based and Risk-Informed Approaches”, 2007

[13] SafSec Methodology, Issue 3.1 [Report] : Standard. - 2006. - S.P1199.50.2.

[14] Jackson Dave and Dobbing Brian: Changing Regulation in Safety and Security – Implications and Opportunities [Conference] // The Relationship between Safety and Security in Software-Based Systems, SafeComp Workshop. - 2008

[15] B. Glas et al., “Automotive Safety and Security Integration Challenges”, Automotive – Safety & Security 2014 (2015) Sicherheit und Zuverlässigkeit für automobile Informationstechnik, Sttutgart, 2015

[16] Piètre-Cambacédès L, Chaudet C. The SEMA referential framework: avoiding ambiguities in the terms ‘security’and ‘safety’. Int J Crit Infrastruct Prot2010;3(2):55–66.

[17] S. Kriaa, M.Bouissou, L.P. Cambacedes, Y. Halgand, A Survey of Approaches Combining Safety and Security for Industrial Control Systems in Reliability Engineering, System Safety 139(July, 2015):156-178

[18] L. Piètre-Cambacédes, M. Bouissou, “Cross-Fertilization between safety and security engineering”, Journal Reliability Engineering and System Safety 2012

[19] Firesmith, D., Common Concepts Underlying Safety, Security and Survivability Engineering, Software Engineering Institute, Carnegie-Mellon University, report CMU/SEI-2003-TN-033, December 2003. Available for download at http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=655+

[20] Lautieri S, Cooper D, Jackson D. SafSec: Commonalities Between Safety and Security Assurance. Thirteenth Safety Critical Systems Symposium, Southampton, 2005

[21] Bock, H-H., Braband J., Milius, B. and Schäbe, H., Towards an IT Security Protection Profile for Safety-Related Communication in Railway Automation in F. Ortmeier and P. Daniel (eds), Computer Safety, Reliability and Security: Proceedings of SAFECOMP 2012, Lecture Notes in Computer Science vol 7612, Springer 2012, 137-148

[22] Simon Burton, Juergen Likkei, Priyamvadha Vembar, Marko Wolf, "Automotive Functional Safety = Safety + Security", In: 1st International Conference on Security of Internet of Things (SecurIT 2012), Kerala, India

[23] The MILS Component Integration Approach to Secure Information Sharing [Conference] // Proceedings o fthe 27th IEEE/AIAA Digital Avionics Systems Conference (DASC). - St. Paul, MN : IEEE, 2008. - DOI: 10.1109/DASC.2008.4702758

[24] Alessandro Cimatti, Rance DeLong, Davide Marcantonio, Stefano Tonetta: Combining MILS with Contract-Based Design for Safety and Security Requirements. SAFECOMP Workshops 2015: 264-276

[25] Apvrille Ludovic and Roudier Yves Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems // In Proceedings GraMSec 2014 / EPTCS 148, 2014, pp. 15-30 - DOI: 10.4204/EPTCS.148.2

[26] http://www.sp.se/en/index/research/dependable_systems/heavens/

[27] HEAVENS project deliverable D2, Security Models, v2.0, March 2016.

[28] HEAVENS project deliverable D4, Interplay between safety and security, v2.0, March 2016.

[29] F. Swiderski and W. Snyder. Threat modelling. Microsoft Press, 2004.

[30] EVITA Project. E-safety Vehicle Intrusion Protected Applications (EVITA). URL: http://www.evitaproject.org/

[31] F. Mhenni, Nga Nguyen; J.-Y. Choley, "Automatic fault tree generation from SysML system models," IEEE/ASME AIM Conference, 2014, pp.715- 720

[32] P. David, V. Idasiak, F. Kratz, Reliability study of complex physical systems using Sysml, Reliability Engineering & System Safety 95 (4) (2010) pp. 431 - 450

[33] J. Xiang, K. Yanoo, Y. Maeno, K. Tadano, Automatic synthesis of static fault trees from system models, in: Proc. of the 5th_ International SSIRI, 2011, pp. 127-136

[34] N. Yakymets, H. Jaber, and A. Lanusse, "Model-based system engineering for fault tree generation and analysis," in 1st International MODELWARDS Conference, Barcelona, Spain, 2013, pp. 210–214

[35] Geoffrey Biggs, Takeshi Sakamoto, Tetsuo Kotoku. A profile and tool for modelling safety information with design information in SysML. Software and System Modelling, Springer, 2014

[36] F. Tajarrod and G. Latif-Shabgahi, "A novel methodology for synthesis of fault trees from MATLAB-Simulink model," World Academy of Science, Engineering and Technology, vol. 17, pp. 1256–1262, 2008

[37] A. Joshi, P. Binns, and S. Vestal, "Automatic generation of static fault trees from AADL models," in IEEE/IFIP DSN Conference, Edinburgh, Scotland-UK, 2007

[38] R. Abdallah, N. Yakymets and A. Lanusse, "Towards a model-driven based security framework," *3rd International MODELSWARD Conference*, 2015, pp. 639-645

[39] Nai Fovino I, Masera M, Cian ADe. Integrating cyber attacks within fault trees. Reliab Eng Syst Saf 2009;94(9):1394–402

[40] Kornecki A, Liu M. Fault tree analysis for safety/security verification in aviation software. Electronics 2013;2(1):41–56

[41] Bezzateev S, Voloshina N, Sankin P. Joint safety and security analysis for complex systems. In: Proceedings of the 13th conference of FRUCT (Finnish– Russian University Cooperation in Telecommunications) association; 2013

[42] Steiner M, Liggesmeyer P. Combination of safety and security analysis- finding security problems that threaten the safety of a system. In: ERCIM/EWICS workshop of SAFECOMP; 2013

[43] Sindre G. A look at misuse cases for safety concerns. In: Jolita Ralyté, Sjaak Brinkkemper, Brian Henderson-Sellers, editors. Situational method engineering: fundamentals and experiences. Springer; 2007. p. 252–66

[44] Raspotnig C, Karpati P, Katta V. A combined process for elicitation and analysis of safety and security requirements. In: Ilia Bider, Terry Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Pnina Soffer, Stanisław Wrycza, editors. Enterprise, business-process and information systems modeling. Springer; 2012. p. 347–61

[45] Jürjens J. Developing safety- and security-critical systems with UML. In: DARP Workshop, Loughborough; 2003

[46] Apvrille L, Roudier Y. Towards the model-driven engineering of secure yet safe embedded systems. In: Pre-proceedings of the international workshop on graphical models for security; 2014

[47] Kornecki AJ, Subramanian N, Zalewski J. Studying interrelationships of safety and security for software assurance in cyber-physical systems: approach based on Bayesian belief networks. In: Proceedings of FedCSIS Conference; 2013. p. 1393–9

[48] Roth M, Liggesmeyer P. Modeling and analysis of safety-critical cyber physical systems using state/event fault trees. In: ERCIM/EWICS workshop of SAFECOMP; 2013

[49] Piètre-Cambacédès L, Bouissou M. Modeling safety and security interdependencies with BDMP (Boolean logic driven Markov processes). In: Proceedings of the IEEE international SMC conference; 2010. p. 2852–61

[50] CONCERTO ARTEMIS JU project: http://www.concerto-project.org/

[51] CHESS ARTEMIS JU project: www.chess-project.org/

[52] CHESS Polarsys Eclipse project: https://www.polarsys.org/chess/

[53] CHESS Modelling Language: https://www.polarsys.org/chess/publis/CHESSMLprofile.pdf

[54] MERgE Project – http://www.merge-project.eu/

[55] Julien Brunel, David Chemouil, Laurent Rioux, Mohamed Bakkali, Frédérique Vallée. A Viewpoint-Based Approach for Formal Safety & Security Assessment of System Architectures // 11th Workshop on Model-Driven Engineering, Verification and Validation, Sep 2014, Spain. 1235, pp.39-48, 2014.

[56] Deswarte, Yves, et al. "SQUALE dependability assessment criteria. "International Conference on Computer Safety, Reliability, and Security. Springer Berlin Heidelberg, 1999.

[57] Corneillie, P., et al. *SQUALE Dependability Assessment Criteria*. No. 98456. LAAS Research Report, 1999.

[58] Schoitsch Erwin - Design for safety and security of complex embedded systems: a unified approach - NATO Advanced Research Workshops, TU Gdansk, published 2005 by Springer, “Cyberspace Security and Defense: Research Issues”, p. 161-174, ISBN-10 1-4020-3380-X, Springer Dordrecht, Berlin, Heidelberg, New York - DOI: 10.1007/1-4020-3381-8_9

[59] http://www.artemis-emc2.eu/

[60] Igor KIrillov, Sergei Metcherin and Stanislav Klimenko : Towards Multi-hazard Resilience as a New Engineering Paradigm for Safety and Security Provision of Built Environment // Transactions on Computational Science XVIII - Volume 7848 of the series Lecture Notes in Computer Science pp 121-136

[61] Jean-Paul Blanquart: Similarities and dissimilarities between safety levels and security levels - http://web1.see.asso.fr/erts2012/Site/0P2RUC89/8A-2.pdf

[62] http://www.aerospace-valley.com/les-projets?keywords=seises

[63] Siwar KRIAA - Joint Safety and Security Modeling for Risk Assessment in Cyber Physical Systems - PhD Thesis // Sciences et technologies industrielles – Université Paris-Saclay / CentraleSupélec. NNT: 2016SACLC014.

[64] Förster Marc, Schwarz Reinhard and Steiner Max Integration of modular safety and security models for the analysis of the impact of security on safety - Fraunhofer IESE, 2010. - IESE-078.10/E.

[65] Jose M. Such, Antonios Gouglidis, William Knowles, Gaurav Misra, Awais Rashid, Information assurance techniques: Perceived cost effectiveness, Computers & Security, Volume 60, July 2016, Pages 117-133, ISSN 0167-4048

[66] HACMS project, http://opencatalog.darpa.mil/HACMS.html

[67] Georgios Despotou, Managing the Evolution of Dependability Cases for Systems of Systems, PhD Thesis, April 2007

[68] Georgios Despotou , John McDermid , Tim Kelly , "Using Scenarios to Identify and Trade-off Dependability Objectives in Design", January 2005

[69] G. Despotou, T. Kelly, "Design and Development of Dependability Case Architecture during System Development", 25th International System Safety Conference. System Safety Society Journal, 2007

[70] Safety Architect, ALL4TEC Tool, https://www.all4tec.net/safety-architect

[71] Apvrille, Ludovic and Saqui-Sannes, Pierre de and Mifdaoui, Ahlem *A UML framework for the dimensioning and formal verification of embedded systems.* (2009) In: SAFA Annual Workshop on Formal Methods (SAFA 2009)

[72] Apvrille, Ludovic and Mifdaoui, Ahlem and Saqui-Sannes, Pierre de *Real-time distributed systems dimensioning and validation: The TURTLE method.* (2010) Studia Informatica Universalis, vol. 3 (n° 8). pp. 47-69

[73] Apvrille, Ludovic and Saqui-Sannes, Pierre // Static analysis techniques to verify mutual exclusion situations within SysML models. (In Press: 2013) In: SDL 2013 - 16th International System Design Languages Forum, 26-28 Jun 2013, Montreal, Canada

[74] Brunel Julien et al. Formal Safety and Security Assessment of an Avionic Architecture with Alloy - 3rd International Workshop on Engineering Safety and Security Systems (ESSS). - Singapore : EPTCS, 2014 - Vol. 150 - DOI: 10.4204/EPTCS.150.2

[75] Bock, H-H., Braband J., Milius, B. and Schäbe, H., Towards an IT Security Protection Profile for Safety-Related Communication in Railway Automation in F. Ortmeier and P. Daniel (eds), Computer Safety, Reliability and Security: Proceedings of SAFECOMP 2012, Lecture Notes in Computer Science vol 7612, Springer 2012, 137-148

[76] http://www.d-mils.org/

[77] M. Born, J. Favaro, M. Winkler, L. Heidt, and A. Boulanger "Integrated Design and Evaluation of Safety and Security in Automotive System Development", Proceedings VDA SYS 2015, 15-16 July 2015, Berlin.

[78] B. Gallina, L. Fabre. Benefits of Security-informed Safety-oriented Process Line Engineering. IEEE 34th Digital Avionics Systems Conference (DASC-34), Prague, Czech Republic, September 13-17, ISBN 978-1-4799-8939-3, 2015

[79] Jose Luis de la Vara, Alejandra Ruiz, Katrina Attwood, Huáscar Espinoza, Rajwinder Kaur Panesar-Walawege, Ángel López, Idoya del Río, Tim Kelly: Model-based specification of safety compliance needs for critical systems: A holistic generic metamodel. Information & Software Technology 72: 16-30 (2016)

[80] http://www.opencossproject.eu/sites/default/files/D5.3_Compositional_Certification_Conceptual_Framework_final.pdf

[81] G. Despotou, T. Kelly. "Investigating the Use of Argument Modularity to Optimise Through-life System Safety Assurance" In proceedings of the 3rd IET International Conference on System Safety (ICSS) 2008, 20-22 October 2008, NEC, Birmingham, U.K. Proceedings by the IET.

[82] J. Fenn, R. Hawkins, P. Williams, and T. Kelly, "Safety Case Composition Using Contracts - Refinements based on Feedback from an Industrial Case Study", Proceedings of 15th Safety Critical Systems Symposium(SSS'07).

[83] A. Ruiz, "A Harmonized Compositional Assurance Approach for Safety-Critical Systems", Ph.D. Thesis, 2015.

[84] C. Schmittner, Z. Ma and P. Smith, "FMVEA for Safety and Security Analysis of Intelligent and Cooperative Vehicles",MVEA for Safety and Security Analysis of Intelligent and Cooperative Vehicles", Computer Safety, Reliability, and Security: SAFECOMP, Florence, Italy, September 8-9, , 2014

[85] G. Despotou and T.Kelly, "An Argument Based Approach for Assessing Design Alternatives and Facilitating Trade-offs in Critical Systems", Journal of System Safety, 2007

[86] Society of Automotive Engineers, http://www.sae.org/

[87] "Cybersecurity Guidebook for Cyber-Physical Vehicle Systems", http://standards.sae.org/wip/j3061/, January 2016

[88] D. Ward, "Overview Of Recommended Practice -SAE J3061™ Cybersecurity Guidebook For

[89] Cyber-Physical Vehicle Systems", A GLOBAL DISCUSSION ON SAE INTERNATIONAL J3061™, December 2015

[90] D. Ward, "Parallels Between J3061™ and Functional Safety Lifecyclein ISO 26262", A GLOBAL DISCUSSION ON SAE INTERNATIONAL J3061™, December 2015

[91] SAE, Requirements for Hardware-Protected Security for Ground Vehicle Applications, http://standards.sae.org/wip/j3101/

[92] SAE, Automotive Cybersecurity Integrity Level (ACsIL), http://standards.sae.org/wip/j3061-1/

[93] "Ground Transportation Cybersecurity White Papers" Thales Ground Transportation, September, 2016