Safety and Security Interference Analysis in the Design Stage

Jabier Martinez, Jean Godot, Alejandra Ruiz, Abel Balbis, Ricardo Ruiz Nolasco

Safety and Security Interference Analysis in the Design Stage

Jabier Martinez[1(✉)], Jean Godot[2], Alejandra Ruiz[1], Abel Balbis[3], and Ricardo Ruiz Nolasco[4]

[1] Tecnalia, BRTA (Basque Research and Technology Alliance), Derio, Spain
jabier.martinez@tecnalia.com, alejandra.ruiz@tecnalia.com
[2] All4Tec, Laval, France
jean.godot@all4tec.net
[3] Thales Alenia Space, Madrid, Spain
abel.balbis@thalesaleniaspace.com
[4] RGB Medical Devices, Madrid, Spain
rruiznolasco@rgb-medical.com

Abstract. Safety and security engineering have been traditionally separated disciplines (e.g., different required knowledge and skills, terminology, standards and life-cycles) and operated in quasi-silos of knowledge and practices. However, the co-engineering of these two critical qualities of a system is being largely investigated as it promises the removal of redundant work and the detection of trade-offs in early stages of the product development life-cycle. In this work, we enrich an existing safety-security co-analysis method in the design stage providing capabilities for interference analysis. Reports on interference analyses are crucial to trigger co-engineering meetings leading to the trade-offs analyses and system refinements. We detail our automatic approach for this interference analysis, performed through fault trees generated from safety and security local analyses. We evaluate and discuss our approach from the perspective of two industrial case studies on the space and medical domains.

Keywords: Safety · Security · Co-Engineering · Interference analysis · fault tree analysis.

1 Introduction

Several engineering disciplines are required to design and build the increasingly complex critical systems present in industrial settings and public infrastructures. Besides the different specialized disciplines related to designing and implementing the software and hardware parts for the functional capabilities of the system, there are experts on assuring the relevant non-functional properties. *Safety* is a non-functional property which considers the mitigation measures to avoid negative impact on humans or the environment, while *Security* is the combination of three criteria: confidentiality, the prevention of the unauthorized disclosure of information; integrity, the prevention of the unauthorized amendment or deletion

of information; and availability, the prevention of the unauthorized withholding of information [2]. Thus, safety and security experts aim to reduce those risks to acceptable values by integrating the needed barriers and measures within the components of the system. However, preventing both safety and security could cause contradictory situations [6] (e.g., the introduction of a security method could cause a time delay which is in contradiction with a safety requirement).

Security is usually needed to ensure safety (security-informed safety) [21] and therefore they are highly interrelated. However, current engineering practices reveal that they are mostly faced independently because safety and security teams have different highly specialized knowledge and skills. For instance, safety experts and security experts tackle the analysis of feared events in different ways [32]. Also, they are forced to show compliance to standards, jurisdictions, and regulations focusing only on one aspect [25] which usually impose the life-cycle, activities, methods, terminology conventions that they should follow, and the expected artefacts that they should produce.

Co-engineering safety and security is still a challenge [13,24] affecting several industrial scenarios such as medical devices, industrial automation, railway, air traffic management or space [25]. Safety and security separation led to redundant efforts [27] and, most importantly for this work, to late identification of conflicts and trade-offs in safety and security requirements [25]. The costs of not identifying issues related to safety and security concerns during early phases of the product life-cycle can be very significant.

In this work, we focus on how to support the co-engineering process in the design stage. We contribute with the integration of safety-security co-engineering within mainstream practices. Concretely, we extend an existing method for the combined analysis of safety and security by introducing an *interference analysis* approach. Interference analysis refers to techniques analysing the mutual influence and inter-links of different quality attributes [5]. Notably, there is a debate about what triggers trade-off meetings and interaction points [25]. They may either be scheduled interaction points or interaction points triggered when a sufficient critical mass of interference needs to be treated. It is the goal of our work to define how this may be identified and measured. We propose a solution to these measurements taking as input *fault trees* [14,17] automatically generated from *component local analyses* of the combined safety-security analysis. Thus, this interference analysis provides high-level reports on the interdependence of safety and security using artefacts from the combined analysis. The objective of these reports is to reveal and trigger the need of a co-engineering meeting and to visualise the evolution of the safety and security interdependence. We also contribute a qualitative evaluation of the presented approach from the perspective of two industrial pilot projects: earth observation and medical devices.

This paper is structured as follows. Section 2 introduces the case studies and Section 3 presents background information on relevant topics needed to better understand this work and using the case studies context. Section 4 positions our approach for the co-engineering method including the interference analysis part.

Section 5 presents the qualitative evaluation and a discussion. Finally, Section 6 presents the conclusions and future work.

2 Case Studies

The two case studies of this work stems from the AQUAS project (Aggregated Quality Assurance for Systems) [25] which objective is to provide a holistic approach to Safety/Security/Performance Co-Engineering. In the presented work, we have focused only on the design stage of two diverse domains, earth observation and medical devices, that we introduce below.

Earth observation market is growing with its main application on military settings followed by usage by civilians and enterprises. Earth observation is in many cases mission-critical and the cyber-physical systems enabling these services have strong requirements on safety, security and performance, notably because of the stringent rules specified for space equipment design. In this work, a simplified version of the AQUAS space case study [25] supports the technical description. The original case study considered the more general Space multicore case, applicable to both Telecom and earth observation payloads. Most of the safety issues are related to the fact of using dual-core architectures (complex resource sharing schemes and software design). It has been simplified because of confidentiality issues but it also provides a more comprehensible presentation. We focused on a subsystem which architecture is illustrated in Figure 1. The mission-critical responsibility of this subsystem is taking pictures (`Camera` component), packaging them before the transmission (`Data packaging` component) and transmitting them to the ground station (`Transmitter` component).

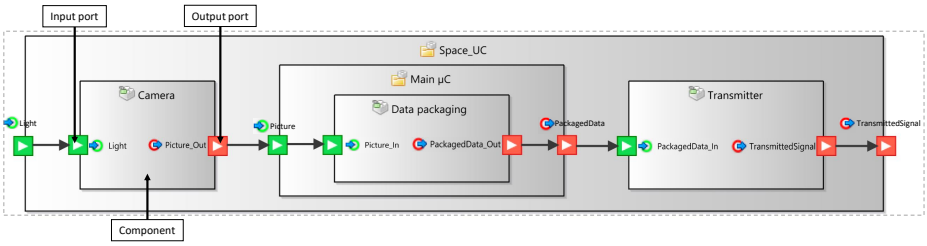

Fig. 1. Simplified system architecture of the space case study

Medical devices integrated in hospital settings and information systems have strong safety, security and performance requirements. In the AQUAS project, the development of a closed-loop controller for muscle relaxation is being investigated with respect to co-engineering challenges [25]. The device consists of a neuromuscular transmission monitor and an infusion pump system with several pumps for different patients. Because of space limitations, the architecture and more details can be found in a related publication [28].

3 Background

This section provides details on safety and security engineering (Section 3.1) as well as basic technical information on component local analysis (Section 3.3) and fault and attack trees (Section 3.2) for a better understanding of this work.

3.1 Safety and Security Engineering

As mentioned in the introduction, safety and security are usually separated processes. An industrial survey on safety and security aspects has shown that the lack of communication between engineering disciplines and their different focus and approaches are considered as a major issue [12]. However, some approaches exist trying to support combined analyses. In the STAMP analysis [18] both disciplines combine applying system engineering for accident cause analysis. Other proposals are, for instance, Failure Modes Vulnerabilities and Effects Analysis (FMVEA) [29], SAHARA (Security-aware Hazard Analysis and Risk Assessment) [20], or the combination of Fault Tree Analysis (FTA) [14,17,26], Stochastic Coloured Petri Net (SCPN) [11], Attack Trees Analysis (ATA) and FMVEA [35]. Besides the safety-security combined analyses in the concept, requirements [4], or risk analyses stages, other techniques are proposed for the subsequent phases. Extensive list of examples of those techniques are available [5,24] and a mapping of safety and security processes have been presented in several application domains such as medical [28] or industrial automation systems [27]. In this work, we focus on the design stage where the architecture and the components involved in the system are defined. This is a crucial stage before the actual software, hardware and communications implementation. Thus, preventing the late identification of issues that might have been avoided through co-engineering in the design stage is of high interest.

3.2 Fault Trees and Attack Trees

Fault Tree Analysis, a widely adopted practice on reliability and safety engineering [14,17,26], proposes a hierarchical structure of events, where the top event is an undesired event or system state, and the rest of the tree are events or gates describing the Boolean conditions which are sufficient to reach the top event. Traditional usage includes probability analysis to asses if the risk is under an acceptable threshold, and identification and analysis of the shortest or more probable paths to reach the top event with the objective to refine the system with the appropriate safety barriers. Besides that, diverse extensions to FTA were proposed [26]. Fault trees quickly get complex in terms of size complicating their human visualization and comprehension [19]. It is even more complicated to reason on high-level concepts (e.g., safety, security) and their interactions by just looking at the fault tree.

In the security engineering domain, a similar approach was used named attack trees [30] for modelling security threats where the top events represent an attack goal. Following the trend of safety-security co-engineering, approaches to

conciliate safety-related fault trees with security attack trees are proposed [32] and industrial experiences of this mix are reported (e.g., railways domain [36]).

3.3 Component Local Analysis

In safety engineering, methods such as Failure Propagation and Transformation Notation (FPTN) [9] or Interface Focused-FMEA (IF-FMEA) [23] have been established to cope with the analysis of complex and large systems based on abstraction and decomposition. Thus, the analysis is conducted locally on components to identify and describe their failure behaviour. Technically, a component local analysis can be represented with specific equation syntax [9], table [23] or diagram [15]. The diagram solution is used in Safety Architect tool[5] with a notation based on Component Fault Trees [15] as depicted in Figure 2. In the earth observation case study, the `Transmitter` component design, with its input and output ports (Figure 1), is enriched with information about failure modes, feared events, and how system or local events can lead to the latter through logic expressed with Boolean gates. For instance, in the `Transmitter` component, perturbations, internal errors of the transmission, or errors from upstream components via the input port can lead to the feared event `Erroneous transmitted signal`.

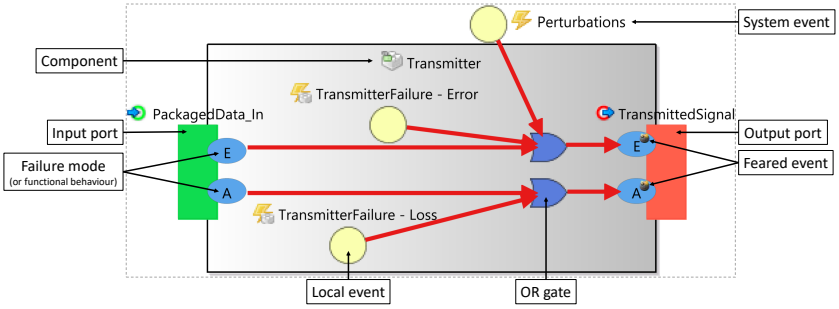

Fig. 2. Example of a safety component local analysis in Safety Architect

This local analysis is performed for single components of the system. However, given that this analysis is performed for each component in the system, the potential of components triggering feared events are captured for the whole system (i.e., the analysis span over components). Then, the enriched information introduced through the system components architecture, can be used to generate a Failure Modes and Effects Analysis (FMEA); taking as input the local analyses of all system components, or, more importantly for this work, a global system safety analysis such as, fault trees for feared events. Generated fault trees for the whole system, gathering the local analyses, present even more challenges

[5] Safety Architect: `https://www.riskoversee.com/en/safety-architect-en/`

for visualization and comprehension given their size and the technical details included directly from the system design.

4 Enriching safety-security co-analysis in the design stage with interference analysis

We propose a reusable building block for safety-security co-engineering in the design stage trying to integrate co-engineering into mainstream practices. Figure 3 is a UML activity diagram representing the enriched safety-security co-analysis approach that we propose. The parts tagged with ① and ② are advanced but established techniques dealing separately with safety and security aspects, and ③ is an advanced co-engineering technique. Then, ④ adds more co-engineering support through advanced interference analysis.

The proposed method falls within the design stage of the development life-cycle (see that the start and end UML symbols at Figure 3 are within the design stage). The main goal is to define the system architecture with the chosen technological solutions to cover the requirements. Then, several engineering domains are involved such as hardware, software, safety, security or performance. These specialist teams for each domain receive inputs including the results of the concept phase: requirements and specification; and they are responsible for evolving the initial system architecture under design. Each of these domains work with their own processes, methods and tools, and progress in parallel during the development life-cycle (e.g., ① and ②).

As mentioned in Section 3.1, recent approaches propose to cross the result of different engineering domains. Our goal is to reduce the number of iterations for designing the system architecture that are usually required to tune the technical solutions and to find and solve potential trade-offs. The proposed method is dedicated to the co-engineering between safety and security domains based on a combined local analysis (③ and more explanations in Section 4.1). The co-engineering interference analysis is supported by an automatic tagging method applied on the fault trees and by high level reports that help to identify and set the scope of the issues to be analysed (④ and details in Section 4.2). Co-engineering meetings are triggered by issues or by an increase on the interference that should be discussed. These moments in the product life cycle where experts from the different disciplines met are called interaction points [25]. In case of trade-offs and where design decisions need to be made, rationale representations [7,31] (e.g., decision reports) are recommended.

The proposed method is associated with a fully integrated toolchain. Regarding tool-support, all parts are supported by Safety Architect and Cyber Architect, and we add the interference analysis with a seamless integration of the Concept-aware analysis library. The safety-security co-analysis and interference analysis are explained in details in the following sub-sections.

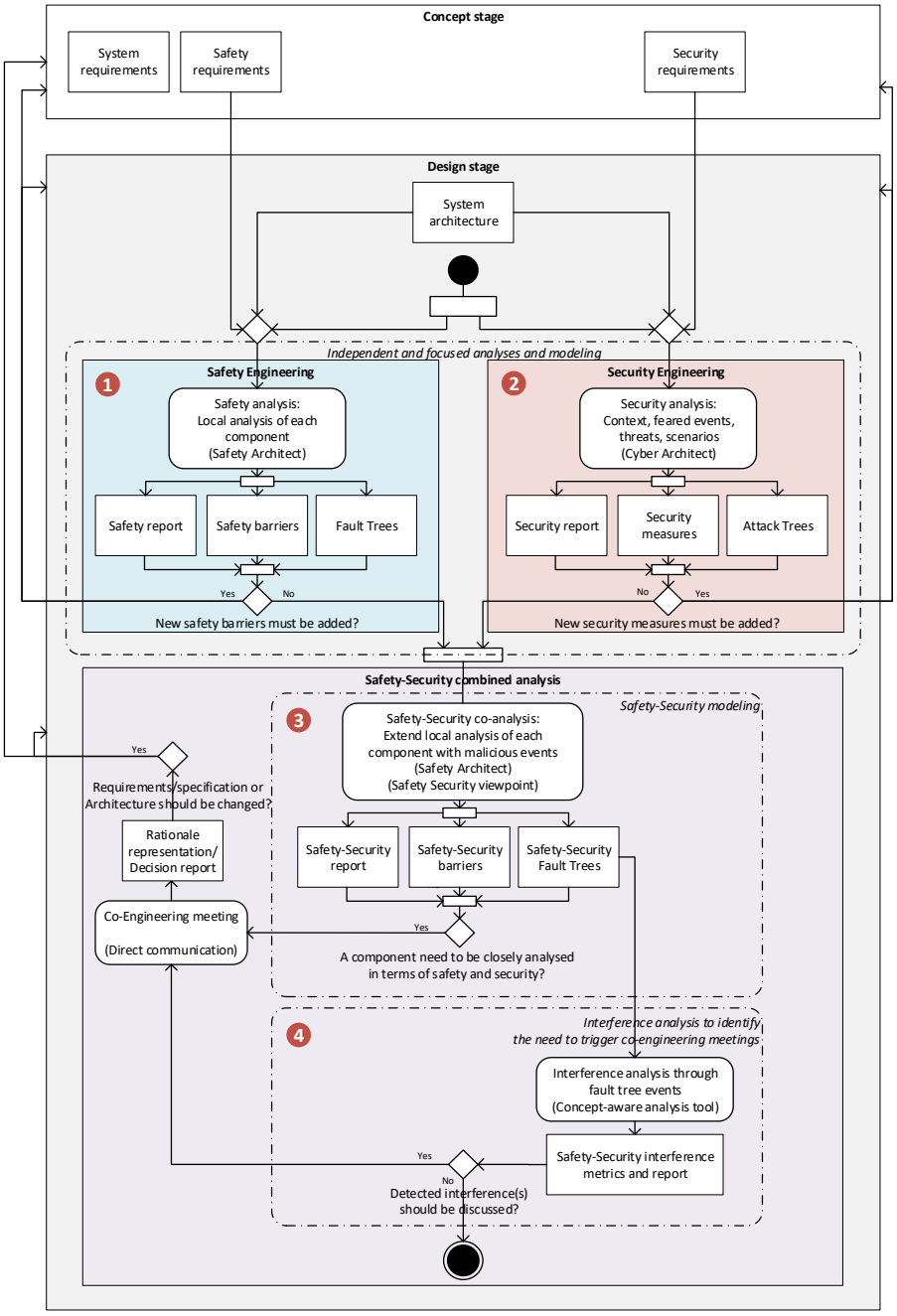

Fig. 3. UML Activity Diagram representing the enriched safety-security co-analysis in the design stage with interference analysis

4.1 Safety-Security co-analysis

As described in the background Section 3, the safety analysis can be decomposed by local analysis of the system components to automatically generate fault trees, FMEA or reports for the whole system. Security analysis has its own concepts and methodology such as Ebios2010 [3] or ISO/IEC 27005 [1]. To propose a co-engineering method, a shared conceptual framework should be defined. Figure 4 presents the mapping proposal between safety and security concepts enabling the safety-security combined analysis (③ in Figure 3).

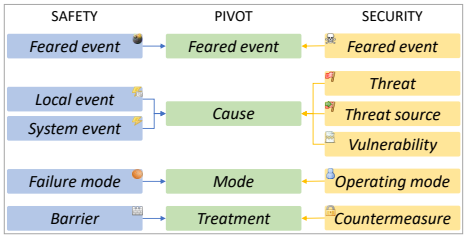

Fig. 4. Mapping between safety and security concepts

Thanks to this mapping, it is possible to bring safety elements into security analysis and vice-versa. Thus, in Safety Architect, a security threat scenario can be displayed with the component local analysis syntax. Figure 5 shows an example. In the `Transmitter` component, the external threat source of malevolent people and internal vulnerabilities and threats can led to the feared event of `Spying`.

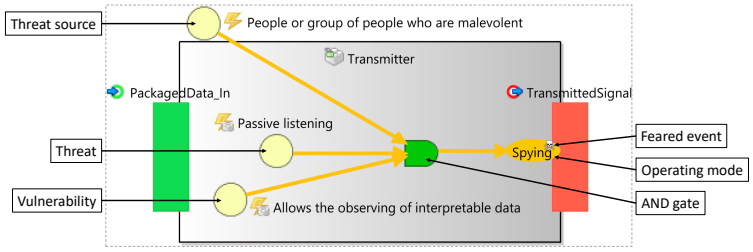

Fig. 5. Example of a security component local analysis

Then a safety-security local analysis can be conducted on each of the components that integrate safety and security concerns to represent their mutual impact. For instance, if safety engineers require a new barrier to comply with the safety goals, or conversely, the security engineers require a new countermeasure to reach the desired security level, an automatic analysis can be conducted to identify if the addition of this element could impact or interfere on other engineering domains.

In the first version of the earth observation case study, no security protection was implemented as, traditionally, with the exception of commercial telecommunications missions, security mechanisms have not been widely employed on civilian space missions. However, in recognition of increased threat, there has been a steady migration towards the integration of security services and mechanisms [33]. Then, in the case study, security engineers require to integrate an encryption module in the telecommunication space system. Thus, the previous security analysis (Figure 5) is updated with the proposal to add a `Cipher` to protect the transmitted message as shown in Figure 6.

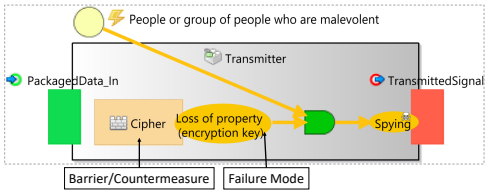

Fig. 6. Example of a security barrier in a component local analysis

The safety-security view allows to overlay, in the same local analysis, the safety and security one. Then, it is possible to analyse how the cipher module impacts the safety-related elements. One of the problems common to all forms of satellite encryption relates to signal degradation caused by different perturbations: terrestrial weather, solar and cosmic radiation or many other forms of electromagnetic noise. Depending on the encryption algorithm chosen, this situation can be particularly problematic because the entire encrypted message may be lost if even a single bit of data is out of place [34]. Then, new propagation links related to safety are added as part of the design of the solution to describe that perturbations can conduct to the failure mode Absent (A) where the feared event "Loss of the message" is associated. Thanks to the combined safety-security local analysis, the safety analysis is updated with new links involving the cipher and the perturbations, as depicted in Figure 7. In this co-modelling

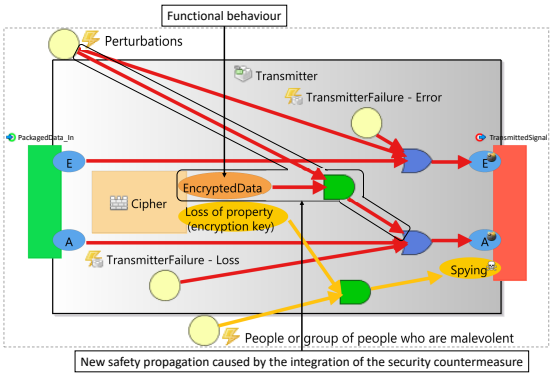

Fig. 7. Example of a safety-security component local analysis

step of the component local analysis, the interference can be easily identified and treated, however, we do not get system-level quantification of the interference.

4.2 Interference analysis

From the combined safety-security local analyses, fault trees are automatically generated. This is the input for ④ in Figure 3. The combined fault trees describe the combination of safety and security events (failure mode, vulnerability, threat) that conduct to a safety or a security feared event. Figure 8 presents an illustrative excerpt of a safety-security fault tree generated from the earth observation example. The events are annotated with `Safety` and `Security`. However, this is not visible in the tools. We added it for illustration purposes. In the example, because of its small size, the identification of the interference of safety and security is easy but in real projects it requires to deeply explore the generated safety-security trees which can count hundreds of events making it time-consuming or impossible to comprehend the safety-security interference.

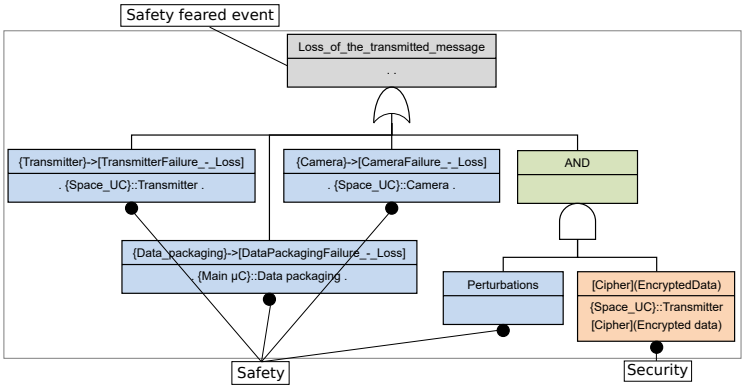

Fig. 8. Excerpt of a Safety-Security tree example

In the use of the CyberArchitect and SafetyArchitect tools, events belong to either safety or security concepts, we were able to export the fault tree models using Open-PSA exchange format [22] with attributes to add this information for each event. This information is seamlessly consumed by the tool named Concept-aware which takes these attributes in the events and performs a propagation mechanism of these tags. When an event is annotated with a tag (e.g., Safety), this event is a Safety-related event but, at the same time, through the propagation, *all ancestors and descendants* are events where this tag is potentially involved. For example, `Cipher,EncryptedData` is a Security event and given that `Loss of the transmitted message` is its ancestor, Security is involved in `Loss of transmitted message`. This information of events and tags (direct and propagated) are used to identify the interference. We automatically create a

formal context to perform a Formal Concept Analysis [10] (a wide-spread technique to create concept hierarchies and groups) where the objects are the events and the properties are the tags. The Concept-aware tool uses the Galatea library to perform this analysis [8]. The information of the obtained concept lattice [10] is then used to create high-level reports and visualisations.

We provided a report consisting on a snapshot of the interference at a given point in time, usually the latest version of the design, and another report on the evolution of the interference given that the design evolve during the design process and also during refinements caused by issues or improvements found in later product life-cycle stages. Figure 9 shows an example of the reports generated from the illustrative example of the fault tree from Figure 8. In the Concept size part on the left, we can observe how all events are involved in Safety and two on Security. In the right side, we can observe the level of the interference of Safety and Security in the system. In the evolution report we can compare the increase of the interference from zero to two (the latter corresponds to the latest version of the fault tree shown in Figure 8).

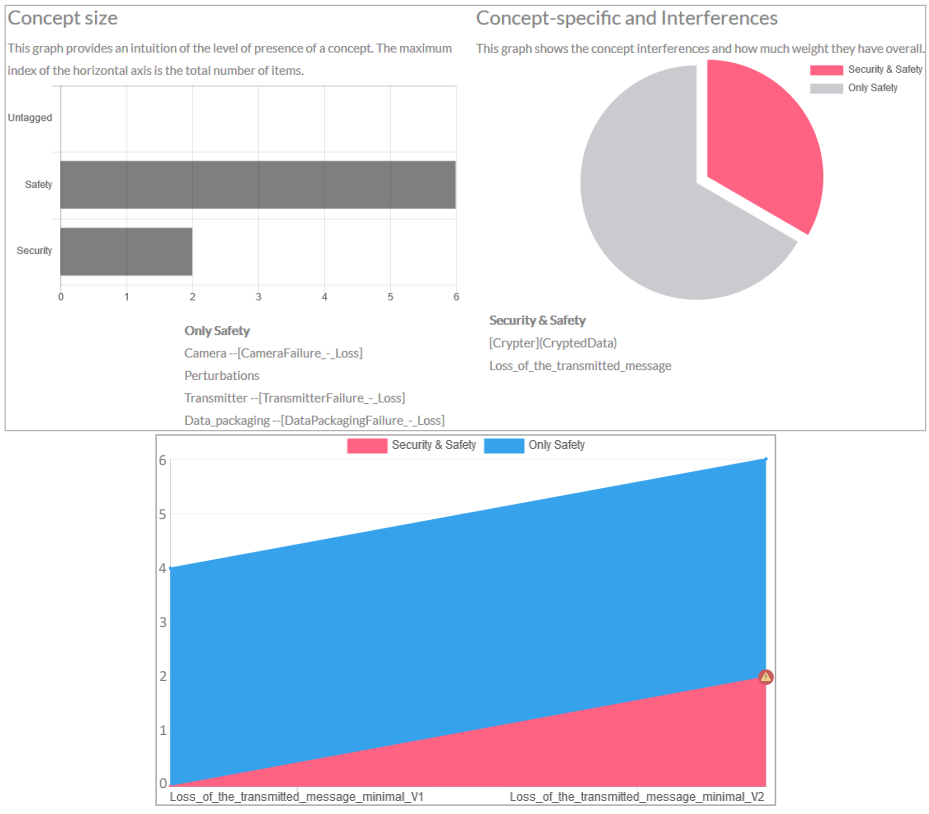

Fig. 9. Concept report and Evolution report of the illustrative example

5 Qualitative Evaluation and Discussion

Given the confidentiality of the case studies, we rely on a qualitative discussion of the practitioners from the industrial companies involved in the pilots. To show the characteristics of the pilots, Table 1 reports on the size of the modelled sub-systems and Table 2 the size of the generated fault trees. As mentioned in Section 3.2, we highlight the difficulty on visualising such large fault trees to infer interferences between safety and security concerns. The following paragraphs are based on feedback from the persons who applied the approach.

Table 1. Number of components (HW: Hardware, SW: Software) for the two pilots

Case study	HW components	SW components	**Total**
Earth observation	2	8	**10**
Medical devices	17	30	**47**

Table 2. Elements in the fault trees (Tmtc: Tele-Metrics to TeleCommunication)

	Feared event	Events	Gates	**Total**
Earth observation	Absent Tmtc Out	24	67	**91**
	Erroneous Tmtc Out	17	49	**66**
	Data Spying	6	17	**23**
Medical devices	Erroneous Drug Dose Rate	43	188	**231**
	Loss of integrity drug dose rate	2	16	**18**

Earth observation feedback: The high level reports on safety-security interference created through the proposed tool-supported process can help to make "trade-offs" decisions at the design stage, specially in large projects, where integration of complex systems and the involvement of different teams make system design decisions more difficult to be evaluated because of the lack of visibility of the fine-grained details. Figure 10 shows the evolution report for a feared event. We can observe how the design evolved taking only Safety into account and then Security was integrated creating a significant interference. The interference analysis report and the design should be analysed to check whether the elements in the interference requires a decision, an action, or introduces a trade-off. As mentioned before, the final objective of this interference analysis technique is to potentially trigger a co-engineering meeting to discuss the implications of the refinements of the design.

Medical devices feedback: The proposed co-engineering method is a structured method that can help refining the design and may led to improve significantly the detection of interferences between safety and security requirements at early stages of the design. This improvement will have a positive impact on the reduction of cost and time required for designing a medical device. The cybersecurity is an increasingly important factor to consider for the design of medical

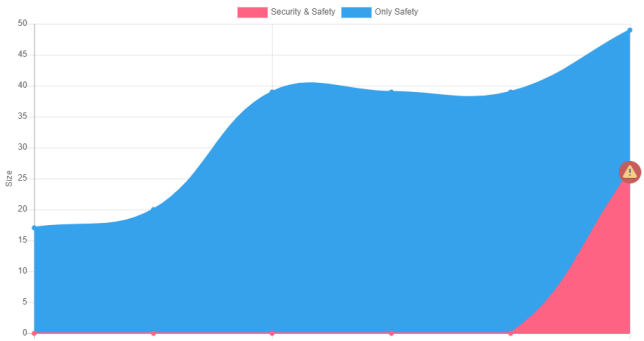

Fig. 10. Evolution report for a feared event of the earth observation case study

devices, so it is becoming highly regulated. Given the interlinks between safety and security, that we already acknowledged in the product-life cycles of RGB products [16], the proposed independent safety and security analyses followed by the combined analysis, can provide evidences that issues related to this interference were considered, and eventually, discussed and treated. As a drawback, RGB has experience on safety and security analysis using fault tree analysis, but integrating these new methods and tools can represent a significant learning curve.

6 Conclusion

We proposed a method for co-engineering in the design stage based on enriching components' local analyses and enabling interference analysis to avoid the later identification of issues and conflicts between safety and security aspects. The system-level reports on safety-security interference are possible through generated fault tree models. These high-level reports can help quantifying the interference at a given point in time as well as from the historic of changes. We used our approach in two pilot projects. As further work, we aim to provide more support for the interference analysis to rank or prioritize the interference events identified in fault tree analysis, and by supporting an integrated interference analysis approach using other artefacts such as requirements.

Acknowledgments

The research leading to this paper has received funding from the AQUAS project (H2020-ECSEL grant agreement 737475). The ECSEL Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme. It is a collaboration between Spain, France, United Kingdom, Austria, Italy, Czech Republic and Germany.

References

1. ISO/IEC 27005:2018 - Information security risk management
2. Abdulkhaleq, A., Wagner, S., Lammering, D., Boehmert, H., Blueher, P.: Using STPA in compliance with ISO 26262 for developing a safe architecture for fully automated vehicles. In: Automotive - Safety & Security 2017, Stuttgart (2017)
3. ANSSI: Expression of Needs and Identification of Security Objectives - EBIOS Security knowledge Base (2010), `https://tinyurl.com/ebios2010`
4. Apvrille, L., Li, L.W.: Harmonizing safety, security and performance requirements in embedded systems. In: DATE 2019. pp. 1631–1636. IEEE (2019)
5. AQUAS project: D3: Combined Safety, Security and Performance Analysis and Assessment Techniques (2019), `https://aquas-project.eu/documents/`
6. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE TDSC **1**(1), 11–33 (2004)
7. Dutoit, A.H., McCall, R., Mistrik, I., Paech, B.: Rationale Management in Software Engineering. Springer-Verlag, Berlin, Heidelberg (2006)
8. Falleri, J.R.: Automatic Refactoring and Alignment of Class Models (Contributions à l'IDM : reconstruction et alignement de modèles de classes). Ph.D. thesis (2009), `http://www.theses.fr/2009MON20103` and `https://github.com/jrfaller/galatea`
9. Fenelon, P., McDermid, J.A., Nicolson, M., Pumfrey, D.J.: Towards integrated safety analysis and design. SIGAPP Appl. Comput. Rev. **2**(1), 21–32 (1994)
10. Ganter, B., Wille, R., Franzke, C.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, Berlin, Heidelberg, 1st edn. (1997)
11. Gehlot, V., Nigro, C.: An introduction to systems modeling and simulation with colored petri nets. In: Winter Simulation Conference. pp. 104–118 (2010)
12. ARC advisory group, Kaspersky, T.M.: The state of industrial cybersecurity (2019), `https://ics.kaspersky.com/media/2019_Kaspersky_ARC_ICS_report.pdf`
13. Gruber, T., Schmittner, C., Matschnig, M., Fischer, B.: Co-engineering-in-the-loop. In: Computer Safety, Reliability, and Security. pp. 151–163 (2018)
14. IEC 61025: Fault tree analysis, second edition (2006)
15. Kaiser, B., Schneider, D., Adler, R., Domis, D., Möhrle, F.S., Berres, A., Zeller, M., Höfig, K., Rothfelder, M.: Advances in component fault trees. In: ESREL (2018)
16. Larrucea, X., Nanclares, F., Santamaria, I., Nolasco, R.R.: Approach for enabling security across PLC phases: An industrial use case. In: EuroSPI (2018)
17. Lee, W.S., Grosh, D.L., Tillman, F.A., Lie, C.H.: Fault tree analysis, methods, and applications - a review. IEEE Transactions on Reliability **R-34**(3), 194–203 (1985)
18. Leveson, N.G.: Engineering a Safer World: Systems Thinking Applied to Safety. The MIT Press, Cambridge, Mass. (Jan 2012)
19. Maaskant, R.: Interactive visualization of fault trees (2016), `https://fmt.ewi.utwente.nl/media/169.pdf`
20. Macher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C.: Sahara: A security-aware hazard and risk analysis method. In: DATE (2015)
21. Netkachova, K., Bloomfield, R.E.: Security-informed safety. IEEE Computer **49**(6), 98–102 (2016)
22. Open-PSA: Fault tree exchange format, `https://open-psa.github.io`
23. Papadopoulos, Y., McDermid, J.A.: Hierarchically performed hazard origin and propagation studies. In: SafeComp. pp. 139–152 (1999)

24. Paul, S., Brunel, J., Rioux, L., Vallée, F., Oliveira, J., Gailliard, G., Gilbert, J.L., Wiander, T., El Bakkali, M., Faucogney, A., Chemouil, D.: Recommendations for Security and Safety Co-engineering - Part A. MERGE project (2016)
25. Pomante, L., Muttillo, V., Křena, B., Vojnar, T., Veljković, F., Magnin, P., Matschnig, M., Fischer, B., Martinez, J., Gruber, T.: The AQUAS ECSEL Project Aggregated Quality Assurance for Systems: Co-Engineering Inside and Across the Product Life Cycle. Microprocessors and Microsystems (2019)
26. Ruijters, E., Stoelinga, M.: Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. Computer Science Review **15**, 29–62 (2015)
27. Ruiz, A., Puelles, J., Martinez, J., Gruber, T., Matschnig, M., Fischer, B.: Preliminary safety-security co-engineering process in the industrial automation sector. In: ERTS 2020, 10th European Congress on Embedded Real Time Systems (2020)
28. Sango, M., Godot, J., Gonzalez, A., Nolasco, R.R.: Model-Based System, Safety and Security Co-Engineering Method and Toolchain for Medical Devices Design. vol. 2019 Design of Medical Devices Conference (DMDC) (2019)
29. Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E.: Security Application of Failure Mode and Effect Analysis (FMEA). In: Computer Safety, Reliability, and Security. pp. 310–325 (2014)
30. Schneier, B.: Attack Trees. Dr. Dobb's Journal (Dec 1999)
31. Shipman, F.M., McCall, R.J.: Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication. AI for Engineering Design, Analysis and Manufacturing **11**(2), 141–154 (1997)
32. Steiner, M.: Integrating Security Concerns into Safety Analysis of Embedded Systems Using Component Fault Trees. Ph.D. thesis, TU Kaiserslautern (2016)
33. The Consultative Committee for Space Data Systems: CCSDS Cryptographic Algorithms (12 2014)
34. Vacca, J.R.: Computer and Information Security Handbook (Third Edition). Morgan Kaufmann Publishers Inc. (2017)
35. Verma, S., Gruber, T., Schmittner, C., Puschner, P.: Combined approach for safety and security. In: Computer Safety, Reliability, and Security. pp. 87–101 (2019)
36. Yi, S., Wang, H., Ma, Y., Xie, F., Zhang, P., Di, L.: A Safety-Security Assessment Approach for Communication-Based Train Control (CBTC) Systems Based on the Extended Fault Tree. In: ICCCN (2018)