20º Congresso Brasileiro de Sistemas

Perspectivas Sistêmicas para a Gestão da Inovação e Proteção Intelectual

Instituto Federal de Mato Grosso do Sul

de 01 a 04 de Outubro

ARCADIA-STPA Integration for Cybersecurity in the Problem Domain: An Aerospace Case Study

Filipe de Paulo Oliveira[1], Vitor C. F. Gomes[2], André F. M. Caetano[2], Christopher S. Cerqueira[3]

[1]Instituto Tecnológico de Aeronáutica (ITA) CEP 12228-900 – São José dos Campos – SP – Brazil

[2]Instituto de Estudos Avançados (IEAv) CEP 12245-021 – São José dos Campos – SP – Brazil

[3]Embry-Riddle Aeronautical University – ZIP 32114 – Daytona Beach – FL – United States of America

paulofpo@ita.br, {vitorvcfg, andreafmc}@fab.mil.br, cerqueic@erau.edu

Abstract. *Complex systems face increasing cybersecurity threats that traditional analysis methods fail to address. This paper proposes a methodological framework that integrates the Architecture Analysis & Design Integrated Approach (ARCADIA) method with the System-Theoretic Process Analysis (STPA) for systemic security analysis. We demonstrate the framework's application through a case study in a real-world system for aerospace simulation. The Loss Scenario Table shows that this synergistic approach effectively identifies four emergent vulnerabilities arising from only one unsafe interactions between components, enabling the implementation of Security by Design principles by linking identified Safety Constraints to architectural elements.*

Keywords: *System-Theoretic Process Analysis, Model-Based Systems Engineering, MBSE, STPA, ARCADIA, Capella, Cybersecurity.*

Thematic Area: *Complex Systems, Systemic Thinking and System Theories*

1. Introduction

The growing sophistication of cyber threats in the aerospace sector has escalated significantly, making it a high-value strategic target and exposing not only operational systems but also complex development and training ecosystems to significant risks. Reports from agencies like ENISA (The European Union Agency for Cybersecurity) consistently warn about the escalation of attacks on critical infrastructure sectors, including

aviation (European Union Agency for Cybersecurity, 2025). This risk is formally recognized by security standards such as RTCA DO-326A/ED-202A (airworthiness security process), DO-356A/ED-203A (continuing airworthiness guidance), ISO/IEC 27005 (information security risk management), and SAE ARP 4754 (aircraft systems development and certification), which extend the concern with security beyond the aircraft in flight to encompass its entire life cycle (Siddiqui et al., 2023).

The Aerospace Simulation Environment (ASA) (DANTAS *et al*., 2022) serves as a strategic decision-support tool designed to simulate complex scenarios within aviation, facilitating the testing and validation of systems and operations. However, its security is of the utmost importance, as a breach could compromise simulated scenarios, extract sensitive tactical information, and act as an attack vector for operational networks, effectively transforming the training environment into a direct threat. Given the increasing sophistication of cyber threats, it becomes clear that the security of ecosystems like ASA can no longer rely on outdated methodologies that fail to adapt to the evolving threat landscape. This highlights a significant gap: there is an urgent need to integrate robust and systemic security approaches that recognize and mitigate emerging vulnerabilities, ensuring proactive defense strategies that protect both simulated systems and actual operations.

The existing gap demands a deeper analysis of the fundamental problem affecting current security engineering practices. Traditional approaches like Failure Mode and Effects Analysis (FMEA), focused on individual component failures, are inherently inadequate for uncovering vulnerabilities that arise from the functional, yet logically flawed, interaction between multiple subsystems (LEVESON, 2012 and SUN; LI; ZIO, 2022). In complex systems, accidents often result from interactions between components that are, individually, functioning as specified (LEVESON, 2012). The direct consequence is that cybersecurity is often treated as a late bolt-on, which is a reactive, expensive, and ineffective approach, as the architectural decisions that originate the risks have already been consolidated. The central problem, therefore, is the inability of conventional methods to mitigate emergent risks, vulnerabilities that are not a property of isolated components but of the system as a whole, born from the complexity of its interactions.

It is in this context that Model-Based Systems Engineering (MBSE) and associated methodologies emerge as the relevant and justifiable foundation for this proposal (SINGAM; CARTER, 2024). MBSE offers the necessary paradigm to overcome this disconnection, establishing a single source of truth that allows for the rigorous and integrated analysis of multiple disciplines from the beginning of the life cycle (ROQUES, 2018). Within this paradigm, the Architecture Analysis and Design Integrated Approach (ARCADIA) methodology provides the formal, function-oriented framework for modeling the system architecture, following the guidelines for a deep security analysis. Complementarily, the System-Theoretic Process Analysis (STPA), based on control theory, is the tool that was used to analyze this architectural "map," as it was designed to identify the emergent risks that traditional methods ignore (LEVESON, 2012).

The innovation and justification of this work lie in the synergistic integration of ARCADIA and STPA to perform aerospace security analysis in the problem domain, creating a method to embed security analysis into the architectural design, considering it an emergent property from conception. This paper proposes a methodological framework that integrates the architectural modeling of ARCADIA with the systemic security analysis of STPA using the Capella tool. The central hypothesis is that this integration results in a

more robust and earlier cybersecurity analysis, while also improving the alignment between the system architecture and security requirements. The main objective is, therefore, to analyze this framework through a case study.

To meet this objective in a structured and logical manner, this article is organized as follows: Section 2 presents the theoretical background on MBSE, ARCADIA, and STPA. Section 3 details the methodology for developing and applying the integrated framework. Section 4 presents the results obtained in the case study. Section 5 discusses the implications of these results. Finally, Section 6 concludes the work, points out its limitations, and suggests directions for future research.

2. Literature Review

2.1. Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE) represents a paradigm change from the traditional document-centric approach to a model-centric one, using formal modeling languages like SysML or methodologies such as ARCADIA. The first reason supporting this transition is overcoming the inherent limitations of document-based engineering. As pointed out by Roques (2018), the goal of an approach like ARCADIA is "to contribute to the transformation of engineering by providing an environment that offers a model-based, rather than a document-based, process-driven procedure, and that delivers effective co-engineering by construction". Instead of text repositories prone to ambiguities and inconsistencies, MBSE establishes a digital, integrated, and authoritative system model.

The second reason lies in the practical benefits that emerge from this change. The adoption of MBSE facilitates the management of systemic complexity, improves communication among multidisciplinary teams by providing a common language, and allows for the early detection of flaws through continuous validations. The ability to verify the solution's adequacy from the initial design phases is vital, as it minimizes the risk of encountering architectural limitations late in the life cycle, when corrections are increasingly more costly (Roques, 2018). By fostering collaborative work around a centralized artifact, MBSE intrinsically improves communication and coherence among the various project stakeholders.

In the context of this work, MBSE serves as a foundation that enables the rigorous integration of various engineering analyses. The formal structure of the model makes it possible to connect security, safety, performance, and other analyses directly to the architecture development. Roques (2018) emphasizes that within the framework of MBSE, various specialized engineering domains (such as safety, performance, cost, and mass) are formalized as distinct viewpoints in relation to the requirements, which subsequently facilitates the verification of the proposed architecture. It is this integration capability that establishes MBSE as the single source of truth and the prerequisite for a cybersecurity analysis that is cohesive, traceable, and conducted early in the system's life cycle.

It becomes clear, therefore, that MBSE is not just an incremental improvement but a a paradigm shift for the engineering of complex systems, to enhance the development, analysis, and management of systems, such as Cyber-Physical Systems (CPS) and Systems of Systems (SoS) (SINGAM; CARTER, 2024). Among the various methodologies that implement this transformation, the ARCADIA approach stands out for its tool-aligned

methodological structure, which is particularly suitable for the purpose of this work.

2.2. The ARCADIA Methodology and the Capella Tool

The ARChitecture Analysis and Design Integrated Approach (ARCADIA) methodology provides the specific MBSE methodology on which this framework is built. Its main feature is the structuring of the engineering process into consecutive and well-defined levels of abstraction. As described by Roques (2018), the methodology guides the engineer from the **problem domain**, through Operational Analysis to capture what the users of the system need to accomplish and System Analysis to define what the system has to accomplish for the users. It then transitions to the **solution domain**, where the Logical Architecture specifies how the system will work to fulfill expectations and the Physical Architecture details how the system will be developed and built. This top-down approach ensures that the solution design is always traceable and justified by the user and system needs captured in the Operational and System Analyses.

The second aspect to consider is ARCADIA's focus on functional analysis. Unlike purely structural approaches, the methodology is "based on functional analysis familiar to all system engineers, and on the allocation of functions to architecture components" (Roques, 2018, p. 77). This means that ARCADIA directs the engineer to first understand "what" the system needs to do (the functions) before deciding "how" it will be built (the components). This principle ensures that the architecture is a direct reflection of the functional and operational requirements, creating a logical and cohesive basis for subsequent analyses.

Finally, Capella[1] is an open-source tool developed to support the practical application of the ARCADIA methodology. It provides modeling capabilities aligned with ARCADIA's levels of abstraction, offering diagrams such as Functional Chains and Data Flow diagrams that formalize the system architecture. In this way, the tool serves as a practical environment for producing the architectural models used in this study, that serves as the ideal input for a systemic security analysis, like STPA.

2.3. The STPA Analysis

System-Theoretic Process Analysis (STPA) is a hazard identification technique that moves away from the traditional analysis of component failures to focus on unsafe interactions within a systemic control structure. Its approach is primarily defined by its theoretical basis. STPA derives directly from the System-Theoretic Accident Model and Processes (STAMP), which, as Leveson and Thomas (2018) state, is "the new accident causality model based on systems theory[...] that provides the theoretical foundation for STPA". This foundation in Systems Theory, which emphasizes emergent properties and complex interactions rather than the analytical decomposition of components (LEVESON, 2012), allows STPA to go beyond traditional failure analysis.

The central innovation of this technique lies in its focus on unsafe control, not failures. Instead of asking "What can fail?", STPA asks "What control actions, even if perfectly executed, can lead to a hazardous state?". This core technique is the identification of Unsafe Control Actions (UCAs). A UCA is, therefore, a control action that, in a particular context, violates constraints and leads to a hazard (LEVESON; THOMAS, 2018).

[1] Available at https://mbse-capella.org

Therefore, the applicability of STPA to cybersecurity is noteworthy. Although originally conceived for functional safety, the STPA control model is directly applicable to the security domain. As STAMP applies to any emergent property of a system, STPA can be used to analyze cybersecurity (LEVESON; THOMAS, 2018). In this context, malicious actors can be modeled as sources of disturbances or as manipulators of the control structure to cause losses. For example, an adversary can inject, falsify, tamper with, or intercept a control action or a feedback channel (LEVESON; THOMAS, 2018), thus making STPA a powerful tool for analyzing emergent vulnerabilities in cyber-physical systems.

STPA, with its foundation in systems theory and its focus on unsafe control, provides the necessary analytical lens to examine the architecture modeled in ARCADIA and identify vulnerabilities that traditional methods would ignore. With the conceptual foundations of MBSE, ARCADIA, and STPA established, it is now crucial to consider how existing literature has addressed the integration of these tools to position the original contribution of this work.

2.4. Related Work

The academic literature recognizes the need to integrate security analyses into the MBSE process. Systematic reviews uncover several proposals that seek to unite STPA analysis with modeling languages like SysML. Works such as those by Span et al. (2024) and Gkoktsis and Peters (2024) highlight the research community's interest in formalizing this integration. However, a critical analysis of these proposals reveals that they often lack a prescriptive methodological guide and a step-by-step process. The integration is often presented at a conceptual level, failing to translate the principles into a practical and replicable workflow.

Studies that specifically unite the ARCADIA/Capella methodology with STPA are scarcer. Additionally, existing works tend to focus primarily on functional safety, leaving a gap in its application to security. A targeted search for works that integrate ARCADIA, STPA, and cybersecurity reveals a scarcity of formalized approaches, confirming that the proposal of an explicit framework for this purpose is an original and necessary contribution.

The literature review confirms that, although the need for integration between MBSE and security analysis is recognized, there is a clear and significant gap: the absence of a prescriptive and validated framework for the specific integration of ARCADIA and STPA with a focus on cybersecurity. To the best of our knowledge, no prior work provides a prescriptive and replicable workflow that integrates ARCADIA/Capella with STPA specifically for cybersecurity analysis. This paper addresses that gap.

Having established the theoretical basis and justified the originality and necessity of the research, the next step is to detail the methodology that was developed to precisely fill the gap.

3. Methodology

3.1. Development of the ARCADIA-STPA Integration Framework

The development of the methodological framework was based on a systematic mapping between the artifacts generated by the ARCADIA methodology and the steps of the STPA analysis. The process connects the formal artifacts of ARCADIA, which describe the

architecture in progressive levels of abstraction (ROQUES, 2018), with the analytical steps of STPA, which include Defining the Purpose, Modeling the Control Structure, Identifying UCAs, and Identifying Loss Scenarios (LEVESON; THOMAS, 2018). This mapping translates, for example, the Entities and Functions of ARCADIA's System Analysis into the Controllers and Controlled Processes of the STPA control structure, creating a bridge between the two approaches.

The central innovation of the framework lies in leveraging the artifacts from the Operational Analysis and System Analysis modeled in Capella as a formal basis for Defining the Purpose of the System, which is Step 1 of STPA. As Leveson and Thomas (2018, p. 15) highlight, "defining the purpose of the analysis is the first step with any analysis method". The initial phases of ARCADIA, which represents the problem domain, define "what the users need to accomplish" and "what the system must do," provide direct, formal, and unambiguous answers to these fundamental questions, ensuring an early and robust alignment between the architecture and security objectives.

At last, the framework's feasibility was realized through the development of a support tool by the Capella community[2], made available as open source to ensure replicability and free use. The existence of this tool demonstrates that the proposed integration is not just a theoretical exercise; it has been implemented in a way that operationalizes systematic modeling, guiding the engineer through a cohesive workflow. The add-on facilitates the creation of the STPA Control Structure from ARCADIA diagrams and a model element in an STPA analysis can be related to a Capella element, serving as empirical evidence of the framework's practicality.

Having detailed the construction of the framework and its tool support, the next step is to demonstrate its application and assess its effectiveness through a case study.

3.2. Instrumental Case Study

To validate the applicability of the framework, an instrumental case study was conducted on the Aerospace Simulation Environment[3], a complex system developed by the Institute for Advanced Studies (IEAv), part of the Brazilian Air Force. The choice of an instrumental case study as a validation method is justified by its ability to allow an in-depth investigation of a phenomenon, in this case, methodological integration, in its real and complex context, which is a consolidated practice for validating new approaches in systems engineering (YIN, 2014).

The object of study, is a real computational system used to support decision-making in air combat scenarios, which lends high credibility and relevance to the results. As a system that integrates multiple software and hardware subsystems to simulate complex tactical scenarios (DANTAS *et al.*, 2022), it represents a non-trivial example of the engineering challenges the framework aims to solve. ASA's complexity, therefore, serves as a test for the scalability and robustness of the ARCADIA-STPA approach.

The selection of ASA was strategic, as its intrinsic characteristics, namely, its aerospace domain and computationally intensive nature, make it an ideal scenario to

[2] STPA Add-On for Capella, available at https://github.com/labs4capella/stpa-capella

[3] Ambiente de Simulação Aeroespacial (ASA) in Portuguese, available at https://www.asa.dcta.mil.br

demonstrate the framework's ability to elicit and visualize cybersecurity requirements. The aerospace sector has particularly strict security requirements due to the potential consequences of a failure (SIDDIQUI *et al*., 2023).

However, beyond the application, it is essential to present the criteria used to evaluate the effectiveness of this approach and measure its contribution.

3.3. Framework evaluation

The evaluation of the integrated framework's effectiveness was qualitative analysis. The choice of this approach was based on the premise that the evaluation of complex engineering methodologies is more robust when not restricted to a single dimension of quantitative analysis, allowing for the capture of procedural benefits and technical outcomes (Wohlin et al., 2012).

Hence, success was measured by the framework's ability to promote a shared and cohesive understanding of cybersecurity risks among stakeholders. This criterion aligns with a central benefit of model-based approaches, where all engineering stakeholders utilize a common methodology, share the same information, and maintain a unified description of the need and the product through a shared model (ROQUES, 2018) . The qualitative evaluation, therefore, focused on observing how the integrated model served as a central artifact for communication and ambiguity resolution during security reviews, this observation was captured during remote meeting sections[4] with computer engineers specialized in simulation.

With the development, application, and evaluation methodology now elucidated, it is possible to present the concrete results obtained through the application of this process to the case study.

4. Results

4.1. Architectural Modeling in Capella

The application of the ARCADIA methodology, as outlined in the framework, resulted in a detailed architectural model of the ASA subsystem. The first modeling step, the Operational Analysis, was crucial for capturing the activities and interactions of the stakeholders. This process established a realistic and well-founded context of use for the architecture, as illustrated in the generated artifacts, such as the Operational Capability diagram (OCB) shown in Figure 1, which identifies external actors (e.g., "Analyst," "Developer") and their main capabilities (e.g., "Simulate Scenarios").

[4] Organized and recorded by Google Meet©

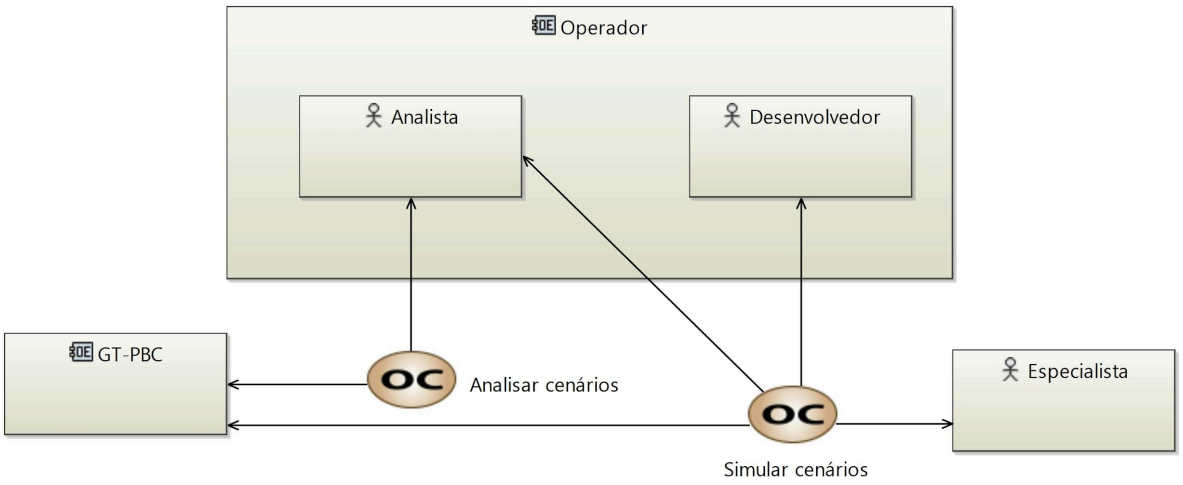


Figure 1. Operational Capability Diagram (OCB)

From this context, the resulting architectural model was developed, covering the Operational and System perspectives. This representation details the primary functions of the system, its constituent components, and the essential data flows, creating a precise and unambiguous representation of the design. Diagrams such as the Operational Architecture (OAB), in Figure 2, and the System Architecture (SAB), in Figure 3, were produced, showing the decomposition of the system into components and the allocation of functions, while Functional Chains, such as "Batch Simulation," allowed for the visualization of critical workflows through the architecture.

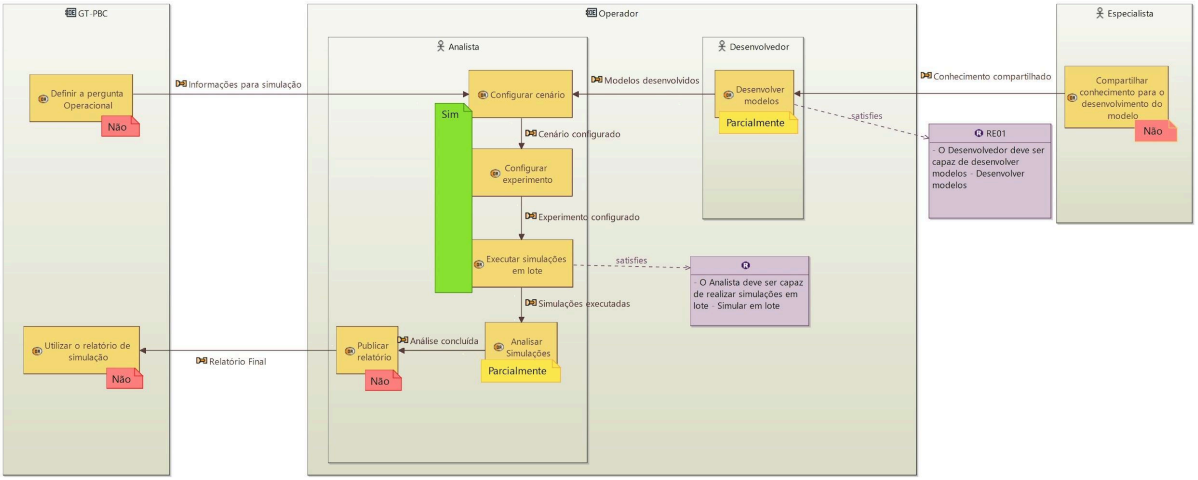


Figure 2. Operational Architecture Diagram (OAB)

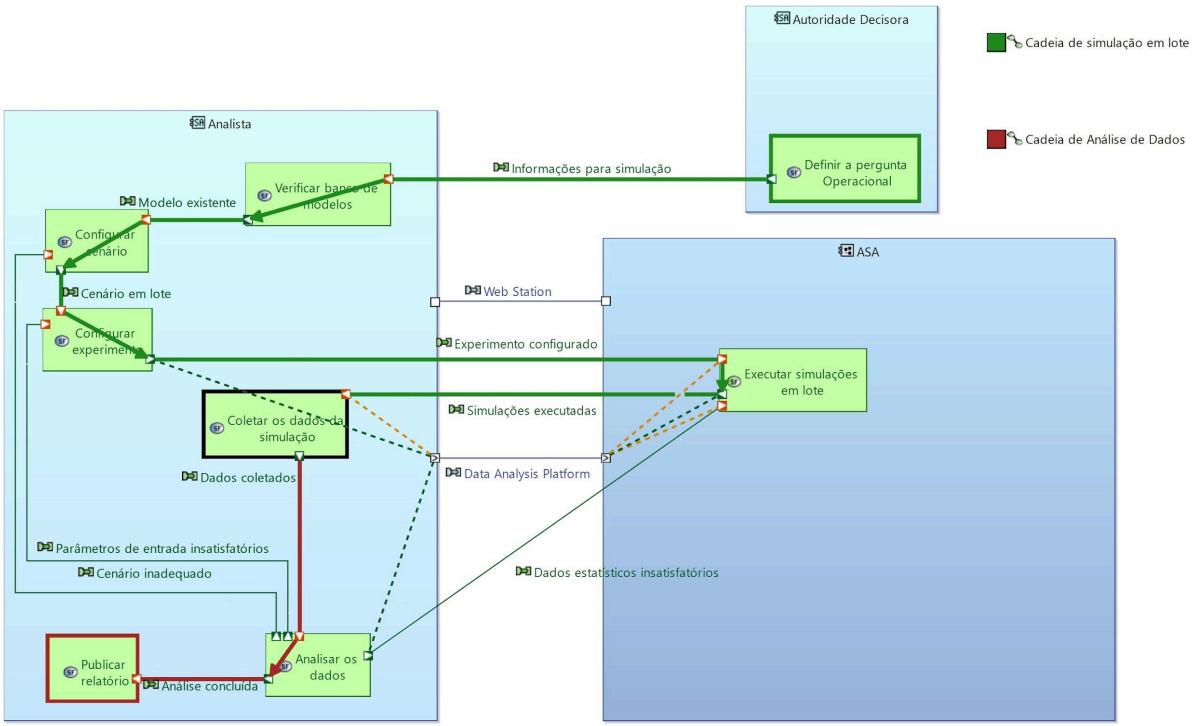


Figure 3. System Architecture Diagram (SAB)

The central artifact of this phase is, therefore, a formal and integrated model in Capella, which serves as the "single source of truth" for the architecture. This MBSE principle, where all stakeholders share the same product description in the form of a model, was achieved. The integrated nature of the model is evidenced by the tool's "Semantic Browser," presented in Figure 4, which demonstrates how a single architectural element is connected to multiple diagrams, functions, and requirements. It is this rigorous and traceable foundation that becomes indispensable for the subsequent systemic security analysis.

The modeling in Capella, therefore, produced a formal and detailed architectural artifact, which serves as the "single source of truth" for the subsystem design. This precise and unambiguous map is the indispensable foundation upon which a rigorous security analysis can be built, the results of which we will present next.

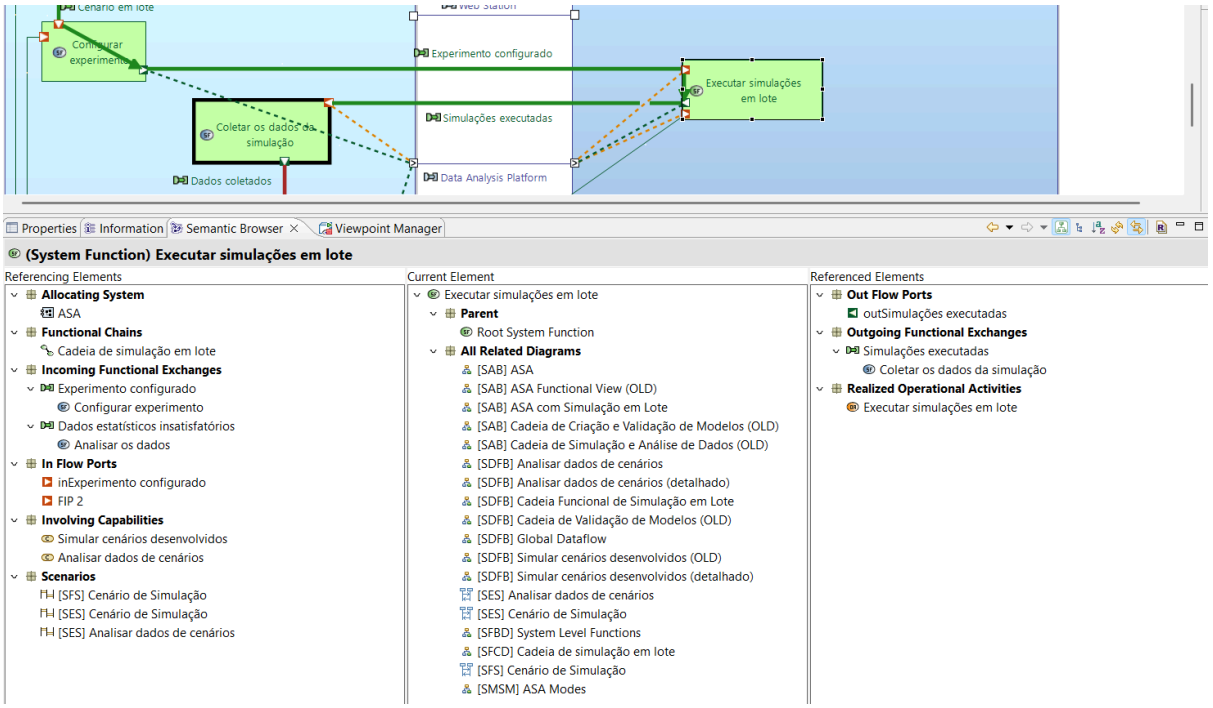


Figure 4. Semantic Browser

4.2. Security Analysis with STPA

The execution of the STPA analysis, using the add-on for Capella and based on the architectural model, produced a set of security artifacts that reveal the systemic vulnerabilities of the ASA subsystem. The first fundamental result, presented in Figure 5, was the construction of tables identifying the items from Step 1: Define the Purpose of the Analysis, namely, Stakes, Losses, Hazards, and System-Level Constraints.

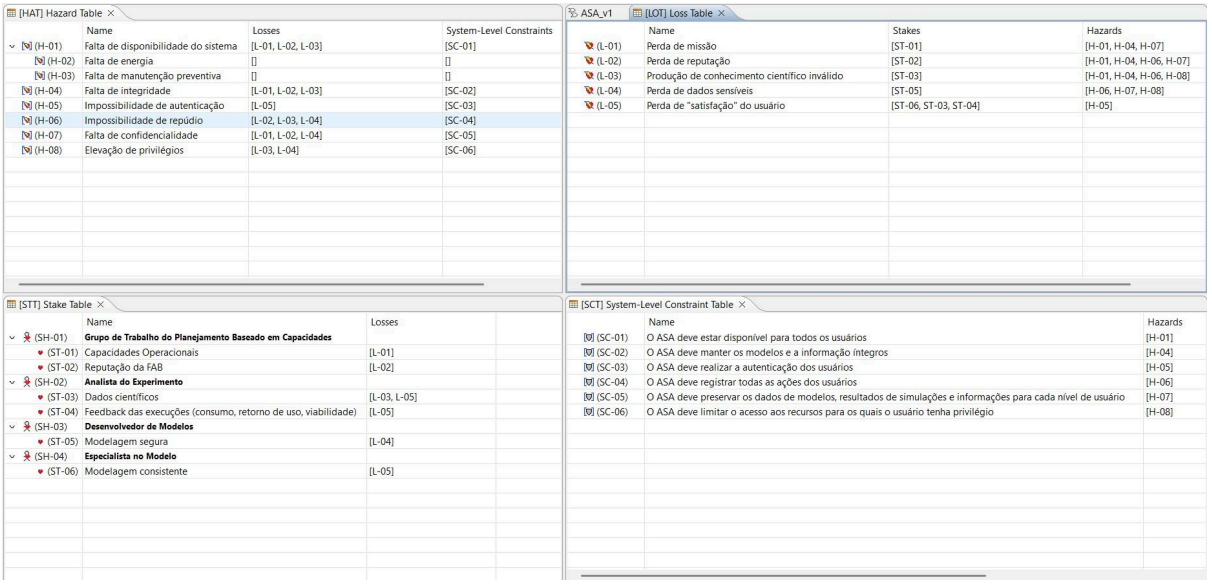


Figura 5. Estrutura de Controle Hierárquica (HCS)

In Step 2 of STPA: Model the Control Structure, the artifact produced was the Hierarchical Control Structure (HCS), which models the control and feedback relationships

between the system and the actors. Figure 6 illustrates this structure, where the model elements were systematically mapped to the controllers and controlled processes of STPA. Highlighted is the "Analyst" actor, and its reference in the model is emphasized in the Semantic Browser. This step ensured a direct and traceable alignment between the system architecture and the risk analysis, serving as the basis for identifying unsafe behaviors.

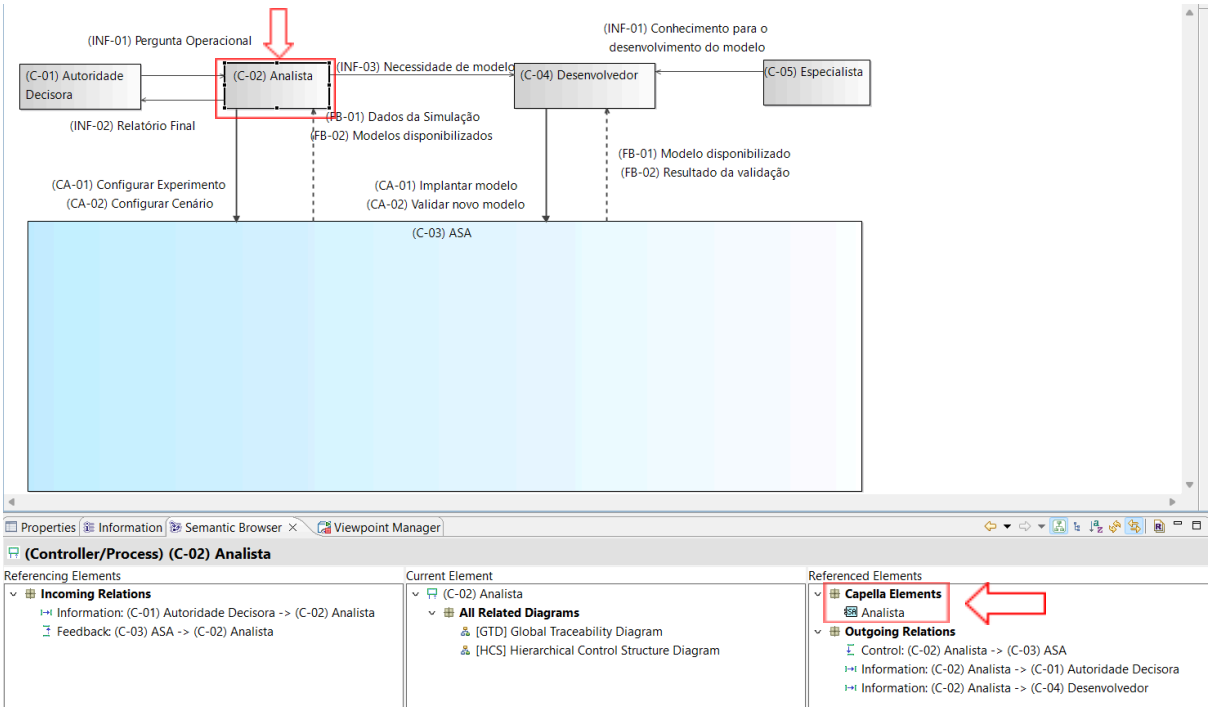
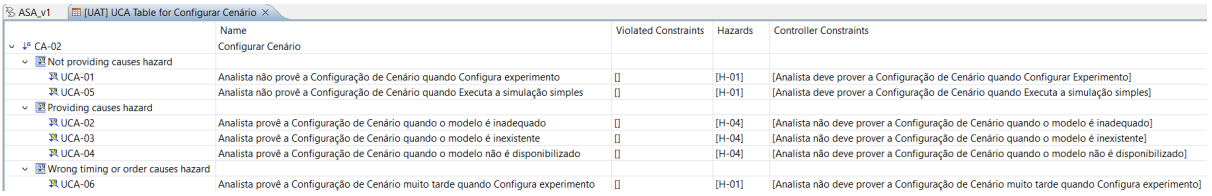


Figure 6. Hierarchical Control Structure (HCS)

The core of the analysis revealed a set of critical Unsafe Control Actions (UCAs) for the subsystem's controllers. These UCAs expose scenarios where the nominal behavior of the system, under certain contextual conditions, could directly lead to a hazard. Figure 7 presents a selection of the most significant UCAs identified for the "Analyst" controller, demonstrating how seemingly correct actions, such as "Configurate Scenario," can become hazardous if provided in an inappropriate context, such as when a previous simulation is still running.

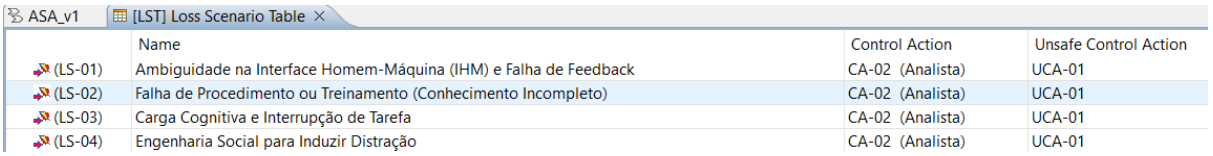

ASA_v1 [UAT] UCA Table for Configurar Cenário

	Name	Violated Constraints	Hazards	Controller Constraints
CA-02	Configurar Cenário			
Not providing causes hazard				
UCA-01	Analista não provê a Configuração de Cenário quando Configura experimento	[]	[H-01]	[Analista deve prover a Configuração de Cenário quando Configurar Experimento]
UCA-05	Analista não provê a Configuração de Cenário quando Executa a simulação simples	[]	[H-01]	[Analista deve prover a Configuração de Cenário quando Executa a simulação simples]
Providing causes hazard				
UCA-02	Analista provê a Configuração de Cenário quando o modelo é inadequado	[]	[H-04]	[Analista não deve prover a Configuração de Cenário quando o modelo é inadequado]
UCA-03	Analista provê a Configuração de Cenário quando o modelo é inexistente	[]	[H-04]	[Analista não deve prover a Configuração de Cenário quando o modelo é inexistente]
UCA-04	Analista provê a Configuração de Cenário quando o modelo não é disponibilizado	[]	[H-04]	[Analista não deve prover a Configuração de Cenário quando o modelo não é disponibilizado]
Wrong timing or order causes hazard				
UCA-06	Analista provê a Configuração de Cenário muito tarde quando Configura experimento	[]	[H-01]	[Analista não deve prover a Configuração de Cenário muito tarde quando Configura experimento]

Figure 7. Unsafe Control Actions (UCA)

For each UCA, detailed loss scenarios were developed that identified not only component failures but, crucially, systemic vulnerabilities emerging from complex interactions, ambiguous requirements, and human-machine interfaces. For example, in Figure 8, an identified loss scenario showed how a sequence of valid commands from the instructor, combined with ambiguous state feedback from the graphical interface, could lead the "Scenario Manager" to start a new scenario over an existing one, corrupting the simulation state. This is an interaction design vulnerability, invisible to traditional analysis

methods focused on component failures.


ASA_v1 [LST] Loss Scenario Table

	Name	Control Action	Unsafe Control Action
(LS-01)	Ambiguidade na Interface Homem-Máquina (IHM) e Falha de Feedback	CA-02 (Analista)	UCA-01
(LS-02)	Falha de Procedimento ou Treinamento (Conhecimento Incompleto)	CA-02 (Analista)	UCA-01
(LS-03)	Carga Cognitiva e Interrupção de Tarefa	CA-02 (Analista)	UCA-01
(LS-04)	Engenharia Social para Induzir Distração	CA-02 (Analista)	UCA-01

Figure 8. Loss Scenario (LS)

The STPA analysis, based on the architectural model, enabled the identification of a set of UCAs and their systemic causal scenarios, revealing vulnerabilities that transcend simple component failures. Identifying these risks is fundamental for a resilient system; thus, the strength of the framework lies in translating these findings into concrete improvements in the architecture, as will be presented in the next subsection.

4.3. Evaluation of the Framework's Effectiveness

The evaluation of the results was conducted based on criteria defined in the methodology, that is qualitative analysis to assess the framework's effectiveness.

The application of the framework demonstrated a clear benefit in promoting a shared understanding of cybersecurity risks. The integrated model in Capella served as a central and unequivocal artifact during security reviews, allowing engineers from different specialties to discuss complex vulnerabilities based on a common and formal representation. This result empirically validates one of the central benefits of MBSE approaches, where the model becomes the "single source of truth" (ROQUES, 2018), reducing ambiguities and facilitating communication.

The evaluation, therefore, confirms that the framework is effective both at the process level (improving communication and understanding) and at the technical level (identifying a more critical and higher-impact class of vulnerabilities).

5. Discussion

The implications of this study for the practice of systems engineering are significant, as the proposed framework offers a methodological path for the effective implementation of the Security by Design principle. The main implication is that the framework transforms this often-abstract principle into a practical, methodological, and replicable engineering process. By providing a model-based and process-driven procedure that unites ARCADIA and STPA, it offers a concrete "how-to," rather than just a "what-to-do," allowing engineering teams to systematically incorporate security into their workflow.

This leads us to the second implication: the ability to shift security left in the development life cycle. By integrating security analysis from the initial phases, the framework allows security to inform and shape the architecture design. This is a fundamental shift from a reactive security posture (finding and fixing flaws in a finished design) to a proactive one (designing a system to be inherently secure). As Leveson (2012, p. 173) states, "the key to having a cost-effective safety effort is to incorporate it into a

systems engineering process from the initial concept development and then design safety into the system as design decisions are made". The ARCADIA/Capella approach, which emphasizes the importance of the initial design phases, is a suitable vehicle for this proactivity.

The third practical implication is the generation of actionable security requirements. The framework generates requirements that are specific, contextual, and directly traceable to the architectural elements that must implement them, a process derived from the transformation of UCAs into controller constraints (LEVESON; THOMAS, 2018). This surpasses the common practice of using generic security checklists. For example, instead of a vague requirement like "The system must be secure", the framework produces a precise requirement such as "The Scenario Manager component **shall not** provide the 'Start Scenario' command **if** the state of the Simulation Process is 'Running'". This type of requirement is directly implementable, verifiable, and validatable, increasing the rigor of the engineering process.

The implications of the results demonstrate that the framework offers a concrete methodological path for implementing Security by Design, shifting the security paradigm from reactive to proactive. With the contribution and impact of this work now clear, the next step in the critical analysis is to synthesize the general conclusions of the study, acknowledge its limitations, and point out directions for future research, ensuring a balanced and complete perspective.

6. Conclusion

This work responded to the challenge of integrating cybersecurity analysis from the initial design phases of complex systems. A methodological framework was proposed that unites the architectural modeling of ARCADIA/Capella with the systemic risk analysis of STPA. The application of the framework in a case study on the Aerospace Simulation Environment (ASA) demonstrated that the integration is not only feasible but also results in a more complete identification of vulnerabilities, capturing both component failures and, crucially, emergent risks from systemic interactions.

To achieve this objective, the article began with a literature review that established the theoretical foundations of MBSE, ARCADIA, and STPA, and identified the gap in the integration of these methodologies with a focus on security. Next, the methodology detailed the development of the framework and its application in a case study. The results presented the concrete artifacts generated, from architectural models to derived security requirements. Finally, the discussion interpreted these results, corroborating the hypothesis that the integrated approach is robust in detecting systemic risks and operationalizes the principle of Security by Design.

We acknowledge the inherent limitations of this study. The validation of the framework was conducted through a single case study, which, although in-depth, demands caution in generalizing the results to other domains or larger-scope systems. The effectiveness of the application also depends on the analyst's expertise in both ARCADIA and STPA, and the analysis presented reflects a specific application that may vary in other teams or organizational contexts.

The conclusions and limitations of this work point to promising future developments. A natural path would be the application of the framework in multiple case

studies in different domains (e.g., automotive, healthcare), to refine and validate its generality. Another important line of research would be the investigation of mechanisms for automating steps of the framework, such as the semi-automatic generation of STPA control structures from more detailed Capella models, or the automatic translation of UCAs into formal security constraints, aiming to increase efficiency and reduce dependence on analyst expertise.

By providing a rigorous and traceable bridge between architecture design and systems theory-based security analysis, the ARCADIA-STPA framework empowers development teams to build security proactively, rather than reactively. We believe that this integrated approach is essential for the development of increasingly complex and interconnected cyber-physical systems, ensuring their resilience and reliability in the face of emerging threats.

7. References

DANTAS, Joao P. A. *et al.* **ASA: A Simulation Environment for Evaluating Military Operational Scenarios**. arXiv, , 23 jun. 2022. Disponível em: <http://arxiv.org/abs/2207.12084>. Acesso em: 29 ago. 2025

EUROPEAN UNION AGENCY FOR CYBERSECURITY. Space threat landscape. 2025.

GKOKTSIS, Georgios; PETERS, Ludger. **The Cyber Safe Position: An STPA for Safety, Security, and Resilience Co-Engineering Approach.** *In*: New York, NY, USA: ACM, 30 jul. 2024. Disponível em: <https://dl.acm.org/doi/10.1145/3664476.3671011>. Acesso em: 14 mar. 2025

LEVESON, Nancy. **Engineering a safer world : systems thinking applied to safety**. *[S.l.]*: The MIT Press, 2012.

LEVESON, Nancy G.; THOMAS, John P. **STPA Handbook**. *[S.l.: S.n.]*.

ROQUES, Pascal. **Systems Architecture Modeling with the Arcadia Method**. *[S.l.]*: Elsevier, 2018.

SIDDIQUI, Fahad *et al.* **Cybersecurity Engineering: Bridging the Security Gaps in Avionics Architectures and DO-326A/ED-202A.** *In*: IEEE, 1 out. 2023. Disponível em: <https://ieeexplore.ieee.org/document/10311187/>. Acesso em: 14 mar. 2025

SPAN, Martin Trae *et al.* **Security Requirements Engineering: A Survey for the Systems Engineer**. *In*: IEEE, 16 out. 2024. Disponível em: <https://ieeexplore.ieee.org/document/10741103/>. Acesso em: 14 mar. 2025

SUN, Liangliang; LI, Yan-Fu; ZIO, Enrico. **Comparison of the HAZOP, FMEA, FRAM, and STPA Methods for the Hazard Analysis of Automatic Emergency Brake Systems.** ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, v. 8, n. 3, 1 set. 2022.

SINGAM, Caitlyn; CARTER, Jeffrey. **Model-Based Systems Engineering (MBSE)** in SEBoK Editorial Board. 2024. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.12, N. Hutchison (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens

Institute of Technology. Disponível em: <https://www.sebokwiki.org>. Acesso em: 27 jun. 2025.

WOHLIN, Claes *et al.* **Experimentation in Software Engineering**. *[S.l.]*: Springer Science & Business Media, 2012.

YIN, Robert K. **Case study researchdesign and methods**. 5. ed. *[S.l.: S.n.]*.