Republic Democratic Algeria of People

Ministry of Higher Education and Scientific Research

University of Saad Dahleb – BLIDA 1

Faculty of Sciences

Department of Informatics

End of Study Memory

Academic Master

Domain: Mathematics and Computer Science

Stream: Computer Science

Specialty: Information systems security

THEME

Toward an integrated approach for security risk assessment in mission oriented System of Systems

Directed by: EL-HAOUARI Amira
ZAOUI Fadhila Sirine

Promotor : Mme Cherfa Imène

Before the jury: LAHIANI Nesrine
CHIKHI Imène

Academic year : 2022/2023

Acknowledgments

Above all, we thank ALLAH the Almighty for giving us the health, strength, and courage to carry out this work.

We would like to express our deepest gratitude to our promoter, madame CHERFA Imene for proposing this initiative and for guiding and helping us throughout the process. She has been a great support and a great skill.

We would like to express our sincere appreciation to the jury members for their interest in this report and for taking the time to review and evaluate our work. We are truly grateful for their expertise and for recognizing the importance of our research.

We would like to express our gratitude to all the teachers who taught us throughout our academic journey. You have inspired us, challenged us, and supported us in our learning and growth. We appreciate your dedication and passion for education.

We would like to express our sincere gratitude to our family especially to our dear parents for their unwavering support and encouragement throughout our academic journey. They have always been there for us, cheering us on, listening to our struggles, and celebrating our achievements. They have taught us the value of hard work, perseverance, and passion. They have also sacrificed a lot for us to pursue my dreams and goals. We dedicate this thesis to them, with love and appreciation.

We would like to thank all the friends who have supported us throughout our thesis journey. They have been a source of help, inspiration, encouragement, and comfort in times of difficulty and stress.

Abstract

This thesis explores the need for an integrated approach to security risk assessment in mission-oriented systems of systems (SoS). It highlights the shortcomings of traditional approaches that focus on individual system components and overlook the interconnected nature of SoS. To address this gap, the thesis advocates for a holistic perspective that considers the interdependencies and interconnectivity within the SoS.

The proposed approach integrates various disciplines and methodologies, such as system engineering, cybersecurity, risk management, and system-of-systems engineering. It employs techniques like system modeling, vulnerability analysis, and impact assessment to thoroughly evaluate security risks within the SoS context.

By embracing this integrated approach, organizations can strengthen the resilience, robustness, and security of their mission-oriented systems of systems. The study emphasizes the importance of a comprehensive comprehension of security risks and the necessity for well-informed decision-making to ensure the successful execution of critical missions in a dynamic and challenging environment.

Overall, this thesis provides a valuable contribution to the field by underscoring the significance of an integrated approach for security risk assessment in mission-oriented systems of systems. It offers insights and recommendations tailored to practitioners and decision-makers in this domain.

Keywords:

Integrated approach, Security risk assessment, Mission-oriented systems of systems (SoS),System engineering, System modeling, Vulnerability analysis, Impact assessment, Mission-critical aspects.

Résumé

Cette thèse explore la nécessité d'une approche intégrée pour l'évaluation des risques de sécurité dans les systèmes de systèmes orientés vers la mission (SoS). Elle met en évidence les limites des approches traditionnelles qui se concentrent sur les composants individuels du système et négligent la nature interconnectée des SoS. Pour combler cette lacune, la thèse préconise une perspective holistique qui tient compte des interdépendances et de l'interconnectivité au sein des SoS.

L'approche proposée intègre différentes disciplines et méthodologies, telles que l'ingénierie des systèmes, la cybersécurité, la gestion des risques et l'ingénierie des systèmes de systèmes. Elle utilise des techniques telles que la modélisation des systèmes, l'analyse des vulnérabilités et l'évaluation de l'impact pour évaluer de manière approfondie les risques de sécurité dans le contexte des SoS.

En adoptant cette approche intégrée, les organisations peuvent renforcer la résilience, la robustesse et la sécurité de leurs systèmes de systèmes orientés vers la mission. L'étude souligne l'importance d'une compréhension globale des risques de sécurité et de la nécessité de prises de décision éclairées pour assurer l'exécution réussie de missions critiques dans un environnement dynamique et difficile.

Dans l'ensemble, cette thèse apporte une contribution précieuse au domaine en soulignant l'importance d'une approche intégrée pour l'évaluation des risques de sécurité dans les systèmes de systèmes orientés vers la mission. Elle offre des perspectives et des recommandations adaptées aux praticiens et aux décideurs de ce domaine.

Mots-clés :
Approche intégrée,Systèmes de systèmes orientés vers la mission (SoS), Ingénierie des système, Modélisation des systèmes, Analyse des vulnérabilités, Évaluation de l'impact, Aspects critiques de la mission.

ملخص

تناولت هذه الأطروحة الجامعية الحاجة إلى نهج متكامل لتقييم المخاطر الأمنية في أنظمة النظم الموجهة نحو المهمات. وسلطت الضوء على نقائص الأساليب التقليدية التي تركز على مكونات النظام الفردية وتغفل الطبيعة المترابطة للنظم المنحى نحو المهمات. لمعالجة هذا الفجوة، تدعو الأطروحة إلى منظور شامل يأخذ في الاعتبار الترابط والتداخل داخل النظم الموجهة نحو المهمات.

يضم النهج المقترح تخصصات ومنهجيات متنوعة، مثل هندسة النظم، وأمن المعلومات، وإدارة المخاطر، وهندسة النظم الموجهة نحو المهمات. ويستخدم تقنيات مثل نمذجة النظم، وتحليل الضعف، وتقييم التأثير لتقييم المخاطر الأمنية بدقة في سياق النظم الموجهة نحو المهمات.

من خلال اعتماد هذا النهج المتكامل، يمكن للمؤسسات تعزيز المرونة والمتانة والأمان في أنظمتها. تؤكد الأطروحة على أهمية الفهم الشامل للمخاطر الأمنية وضرورة اتخاذ القرارات المستنيرة لضمان تنفيذ المهمات الحرجة بنجاح في بيئة ديناميكية وتحديات.

بصفة عامة، تقدم إسهاما قيما للمجال من خلال التأكيد على أهمية النهج المتكامل لتقييم المخاطر الأمنية في أنظمة النظم الموجه نحو المهمات. وتقدم نصائح وتوصيات متخصصة للممارسين ولصناع القرار في هذا المجال.

الكلمات المفتاحية:
هندسة النظم ، نمذجة الأنظمة ، تحليل الضعف ، تقييم الأثر ، الجوانب (SoS) في أنظمة النظم ، أنظمة موجهة نحو المهام الحرجة للمهمة.

Contents

List of abbreviations

ADE: Application Domain Expert.

ATHENA-IP: Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications Integrated Project.

BPM: Business process management.

CS: constituent system.

DODAF: Department of Defense Architecture Framework.

DSL: domain specific language.

EMF: Eclipse modeling framework.

EIF: European Interoperability Framework.

FDNA: Functional Dependency Node Analysis.

GMF: graphical modeling framework.

IEEE: Institute of Electrical and Electronics Engineers.

INCOSE: International Council on Systems Engineering.

INFOSEC: information security management system.

ISMS: information security management system.

ISO: International Organization for Standardization.

MITRE: Massachusetts Institute of Technology Research and Engineering.

MoE: Measurements of Effectiveness.

MOP: mission-oriented process.

MOPSE: mission-oriented process system engineering.

MoP-SoSE: Mission Oriented Process for System of Systems Engineering.

NIST: National Institute of Standards and Technology.

OASOSIS: OCTAVE Allegro for System of Systems Information Security.

OCTAVE: Operationally Critical Threat, Asset, and Vulnerability Evaluation.

OMGEE: operational managerial geographical evolutionary emergent.

OT: Operational technology.

SE: system engineering.

SIMEX: Simulation Exercise.

SSE: system security engineering.

SysML: system modeling language.

SoS: System of Systems.

SoSE: System of Systems Engineering.

UML: Unified Modeling Language.

XML: Extensible Markup Language.

List of figures

List of tables

General Introduction

This research explores the necessity of a comprehensive approach to evaluating security risks in mission-oriented systems of systems (SoS). It emphasizes the limitations of traditional methods that focus on individual system components while neglecting the interconnected nature of SoS. This study advocates for a holistic perspective that acknowledges the interdependencies and interconnectedness within the SoS.

The problem addressed in this memory is the need for an integrated approach to security risk assessment in mission-oriented SoS. Traditional risk assessment methods often focus on individual systems and may not adequately capture the intricate dependencies and interactions present in an SoS. This can result in incomplete risk identification, insufficient risk analysis, and inadequate risk mitigation strategies. Therefore, there is a requirement for a comprehensive framework that considers the unique security risks associated with the interconnections and interdependencies within an SoS.

This memory aims to propose an integrated approach for security risk assessment in mission-oriented SoS engineering. The approach aims to provide a structured framework that takes into account the interconnected nature of systems within an SoS, the operational context, and the mission objectives. By integrating various methodologies, standards, and best practices, the objective is to enable organizations to effectively assess and mitigate security risks throughout the entire life cycle of the SoS. The goal is to enhance the overall security posture, resilience, and mission success of mission-oriented SoS by addressing the specific security challenges inherent in these complex systems.

By embracing this integrated approach, organizations can bolster the resilience, robustness, and security of their mission-oriented systems of systems. The research underlines the significance of a comprehensive understanding of security risks and the importance of informed decision-making to ensure the successful execution of critical missions in a dynamic and challenging environment.

The rest of the report is organized as follows:

General Introduction

- Chapter 1: Focuses on the System of Systems definition, the SoSs characteristics, the typology of SoSs, and will describe the engineering activities of the mission-oriented process for System of Systems. This chapter ends with the description of the challenges of security engineering in context of SoSs.
- Chapter 2: Explains what is risk assessment and some of important approaches: Risk Assessment Methodology provided by the National Institute of Standards and Technology (NIST) Guide, o the System of Systems Security Engineering (SSE) Framework and the OCTAVE Allegro for System of Systems (OASoSIS).
- Chapter 3: explores the integration of NIST risk assessment steps within a mission-oriented process in Systems of Systems (SoS) engineering. It presents the case study and a detailed description of each step within the SoS crowd management scenario. The chapter also introduces the proposed Metamodel. Additionally, it presents a concrete syntax and the final user of the application.
- Chapter 4: Presents the implementation of the modeling. We begin by defining the work environment, as well as the development tools used. We conclude this chapter with the presentation of our application through a case study.
- The general conclusion of the project serves as a summary of the findings and restates the original objectives. It emphasizes the valuable contributions and insights derived from the research study. Moreover, the conclusion explores the implications of the research and offers recommendations for future work in the field.

1. Introduction

The mission-oriented process for System of Systems (SoS) engineering provides a structured approach to achieving mission objectives within the context of complex, interconnected systems. In this chapter, we will explore the key elements of this process, including the definition of SoS, the SoSs characteristics, the typology of SoSs, and will describe the engineering activities of the mission-oriented process for System of Systems. This chapter ends with the description of the challenges of security engineering in context of SoSs.

2. SoS Definition

The ISO/IEC/IEEE 21839 standard defines the System of Systems (SoS) as "a set of systems or system elements that interact to provide a unique capability that none of the constituent systems can accomplish on its own. Note: Systems elements can be necessary to facilitate the interaction of the constituent systems in the system of systems" [1].

Systems-of-systems are unique systems that are made up of other systems that can function on their own and have their own advantages and values. The element system retains its independence once it is incorporated into the system-of-systems, and interactions between the systems are common [2].

3. Characteristics of SoSs

In a response to a growing acceptance of SoSs associated to the lack of a common consensus on an SoS definition, Maier [4][5] identified five principal characteristics to distinguish between systems of systems and complex monolithic systems, often known by the acronym "OMGEE":

- **Operational independence of the constituent systems**, which means that the constituent systems must be able to achieve their own mission if the SoS is disassembled.

- **Managerial independence of the constituent systems**, which means that the constituent systems must be managed individually, they are acquired individually, and have their own life cycle and organization.

- **Geographical distribution of the constituent systems** means that the constituent systems are distributed over a large geographic extent. This geographical expansion is relative and relies on the available communication means and technologies. The constituent systems can exchange information and not considerable quantities of mass or energy.

- **Evolutionary development of the SoS**, which means that the SoS's objectives can change constantly and its development can be gradual. Over time, some features can be removed, modified or added, likewise, some constituent systems may be disassembled from the SoS.

- **Emergent Behaviors**, which means that the SoS capabilities could not be achieved by any of its constituent systems. Therefore, these emergent behaviors are unpredictable which lead to difficulties in validating the SoS.

4. Typology of SoSs

The degree of complexity of a SoS is high since it comprises several connected systems that are managerially and operationally autonomous. In order to meet the issues brought on by complexity, the discipline of SoSE attempted to develop many forms of SoS; this classification provides a framework for understanding SoS.

Four kinds of SoSs have been discovered by more SoSs research, and standardization by the ISO/IEC/IEEE 21841 [6]. These categories, as shown in Figure 1, are primarily dependent on the level of control and accountability over the SoS and its development.

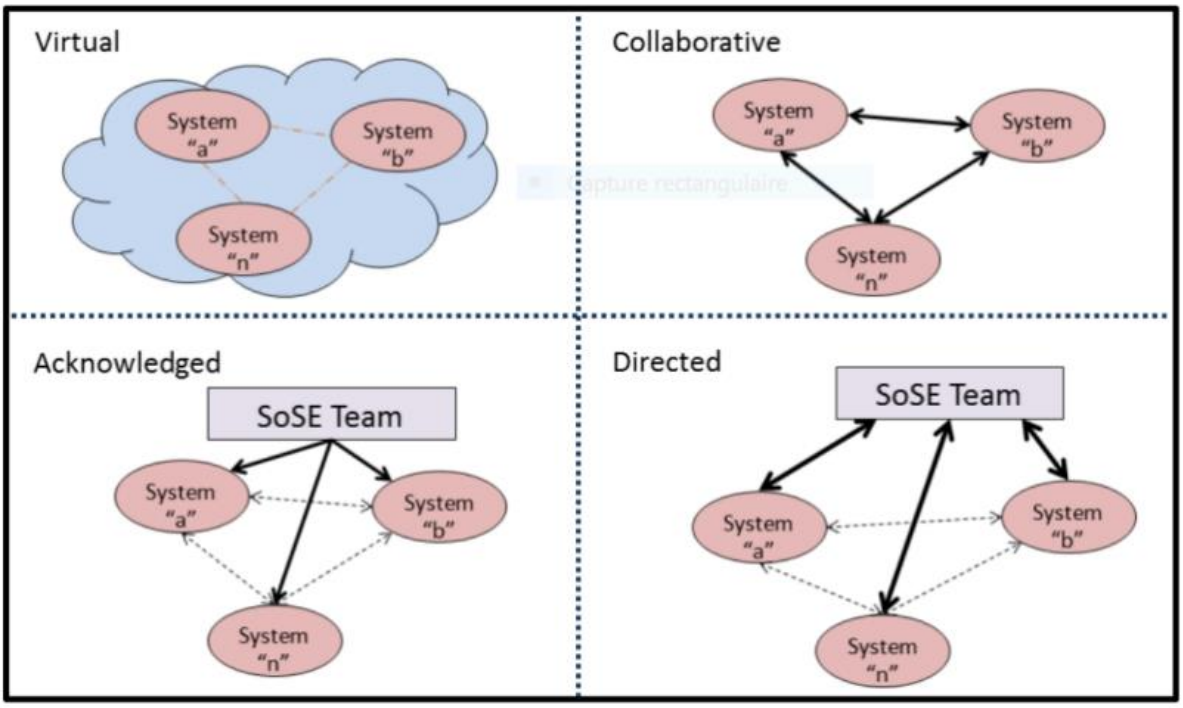


Figure 1: SoS Categories [7]

4.1. Virtual:

Virtual SoS is distinguished by the lack of centralized control and a unified objective. It is usually ad hoc, and the underlying mechanisms aren't always recognized [8]. made the suggestion of a virtual SoS, citing the Internet and other services that can be created or merged on-demand as examples.

4.2. Collaborative:

By [8], collaborative SoS was defined. The system engineering teams that make up a collaborative SoS cooperate with one another more or less willingly to accomplish the key shared objectives. There is no central authority in a collaborative SoS. [7] used the regional area disaster management system as an example of a collaborative SoS, in which each entity involved in first response circumstances is in charge of its own systems.

4.3.Acknowledged:

Although the SoS has a clear mission, its component systems continue to maintain their own ownership, goals, and development. This SoS type's modifications are based on agreements reached in cooperation between the system and the SoS.

4.4.Directed:

Directed SoS is created and managed to achieve specific goals. During long-term service, it is centrally regulated to continue achieving those objectives as well as any further ones the system owners may want to take on. Although they are regulated to achieve the SoS objectives, constituent systems can function independently. The SoSE team has the power to demand these component systems to create and support SoS capabilities, which is frequently done through some type of contract, as shown in Figure 1 by the bi-directional arrows connecting the SoSE team and important constituent systems (but not necessarily all) [7]. We use the integrated air defense network, which is often centrally managed to defend an area against adversary systems, as an example of directed SoS [9]. Its constituent systems may function independently.

5. System of Systems Engineering definition, issues and perspectives

To develop SoS and address the various new difficulties, SoSE is an essential extension and evolution of traditional systems engineering. SoSE was described by Keating et al. [10] as “design, deployment, operation and transformation of meta-systems that must function as an integrated complex system to produce desirable results”. The Systems Engineering Guide for System of Systems [11] suggested that SoSE deals with “planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts”. Several surveys confirmed that at this stage of SoSE development, there is no precise and unified definition of SoSE.

The INCOSE Systems Engineering Handbook (1) does not define SoSE; instead, it views system engineering (SE) as a field that works with all types of systems, despite the fact that each kind uses distinct procedures and techniques. Additionally, the manual makes the assumption that "SoS is itself a system, and the systems engineer may choose to address it as either a system or as a SoS, depending on which perspective is better suited to a particular problem." In this article, we'll assume that SoSE is a branch of SE that focuses on SoS. The following topics might be categorized as SoSE's primary study areas:

5.1. Modeling and Architecting:

the creation of models that utilize current systems as SoS components and optimize design while taking the SoS dimensions into consideration. Additionally, the analysis of SoS mission and capabilities objectives using Domain Specific Languages (DSL) and the definition of operational development ideas.

5.2. Simulation:

the idea of using simulation tools to examine and comprehend how complicated SoS behavior.

5.3. Testing:

the usage of testing methods in situations where there are several stakeholders, complicated and huge SoS, different standards are being used, and so on.

5.4. Verification:

the creation of verification tools to assist testing and simulation and to analyze various characteristics.

6. Mission-Oriented Process in Systems of Systems Engineering

In order to link SoS objectives to the individual functionalities undertaken by the CSs (Constituent Systems) in SoSs context. [12] considers that SoSs are acquired to satisfy new capabilities in a mission context. The latter is a key element to assist SoSs engineers to determine the systems that must be involved and the functions they must perform. In this perspective, the authors of [12] proposed a process to build and evolve SoSs, that is called MOP-SoSE (Mission Oriented Process for System of Systems Engineering).

Figure 2 illustrates the process. It consists of different engineering activities, and involve several stakeholders. The process offers a disciplined procedure for explicitly specifying the SoS end-to-end mission and generating the appropriate architecture. It is composed of top-down planning and decision making, and bottom-up adjustments. The process aims to refine the mission, until the architecture is reached, while preserving the mission traceability. Therefore, the refinement activities are as follows:

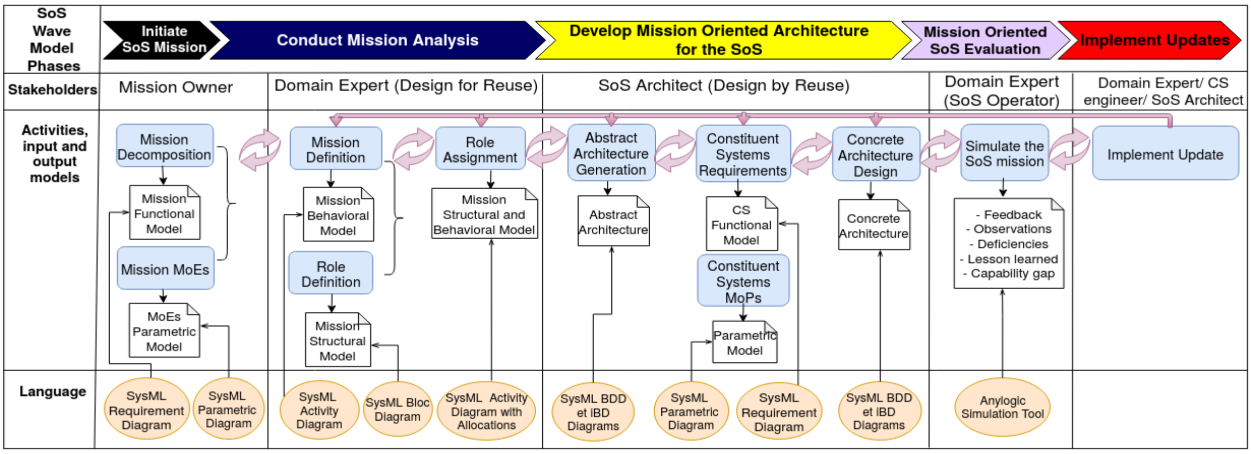


Figure 2: Actors and Responsibilities of MOP-SoSE [12]

- **Mission decomposition:** this activity is intended to provide a functional coarse grain view of the mission. This aspect is achieved through an analysis of the general mission objectives to recursively identify more precise sub-mission objectives. The criterion for stopping the mission decomposition is the identification of a process that can perform a given sub-mission. Therefore, this step results in a mission functional model of the SoS.

- **Mission Measurements of Effectiveness (MoEs) definition:** in this activity, the mission owner defines effectiveness measures, using the system modeling language (SysML) parametric diagram. These measures will be used as metrics to assess the overall performance of the SoS mission.

- **Mission definition:** the aim of this activity is the design of the operational view of the mission. It consists on the definition of mission threads and activities. It results on a fine-grained behavioral view of sub-missions using activities. The view is elaborated using the SysML activity diagram.

- **Role definition**: the role is used to provide an abstract representation of hierarchy of entities having capabilities that enable the achievement of the mission. The produced model for role definition is based on a SysML profile extending the block definition diagram.

- **Role assignment:** this step is intended to designate the role that must be associated with each action of an activity. This association creates a link with the constituent system through the assigned role.

- **Abstract architecture generation**: the architecture is a structural view that describes the constituent systems of the SoS and their connections. However, all the above-mentioned definitions refer only to roles instead of constituent systems. Therefore, the first generated architecture from the given definitions corresponds to the abstract architecture of the SoS. It is described using both of the SysML internal block diagram and SysML block definition diagram.

- **Concrete systems requirements:** before replacing roles with concrete CSs. The architect can identify new requirements at the CSs level, necessary for their integration into the SoS. These new requirements may require negotiation with CSs engineers, and are described using the SysML requirement diagram.

- **Concrete systems Measurements of Performance (MoPs):** MoPs are described for each service in the architecture, to determine the capabilities and limitations of all relevant CSs. This helps to choose the best CS to handle a given action. The SysML parametric diagram is used to capture MoPs.

- **Concrete architecture design:** the abstract architecture is progressively refined during the architecture analysis to get the concrete architecture. For this activity, both the SysML internal block diagram and SysML block definition diagram are employed.

- **Simulate the SoS:** the simulation of models is necessary to evaluate the ability of an architecture configuration, to accomplish the specified SoS mission. It allows also to confirm performance, and to discover errors.

- **Implement update**: updates can be done at the level of the SoS models, or at the level of the CSs. Individual updates within a CS, follow system engineering life-cycle. The ADE (Application Domain Expert) has only to influence those changes with the CS engineers.

7. Security Engineering of Systems-of-Systems

[13] highlights the cyber-security issues in SoS, that could result from SoS characteristics, and that may differ from monolithic systems. We present them in the following [13]:

7.1.Operational independence:

In an SoS, the component systems may be operated separately, under different policies, using different implementations. This can lead to potential incompatibilities and conflict between each system's security, including different security requirements, protocols, procedures, technologies and culture. Additionally, some systems may be more vulnerable to attack than others, and compromise of such systems may lead to compromise of the entire SoS.

7.2.Managerial independence:

Component systems may be managed by completely different organizations, each with their own agendas. In the cyber security context, activities of one system may produce difficulties for the security of another system. What rights should one system have to specify the security of another system for SoS activities and independent activities? How can systems protect themselves within the SoS from other component systems and from SoS emerging activities? Does greater fulfilment require a component system to allow other component systems to access it?

7.3.Evolutionary development:

An SoS typically evolves over time, and this can introduce security problems that the SoS or its components do not address, or are not aware of. Therefore, the security mitigations in place for an evolving SoS will be difficult to completely specify at design time, and will need to evolve as the SoS evolves.

7.4.Emergent behavior:

SoS are typically characterized by emerging behaviors and functions that occur after the SoS has been deployed. These could clearly introduce security issues for the SoS or for its component systems, and therefore the security of the SoS will again need to evolve as the SoS evolves.

7.5.Geographic distribution:

An SoS is often geographically dispersed, which may cause difficulties in trying to secure the SoS as a whole if national regulations differ. These may restrict what can be done at different locations, and how the component systems may work together to respond to a changing security situation.

8. Challenges in Security Engineering of Systems-of-Systems

Starting from the challenges related to characteristics specific to SoS,identifies and describes challenges to security engineering of SoS. The authors organize them according to the activity in which they have the most impact. The activities considered are: requirements, design, implementation, verification, release/response activities. In our work we are interested by the requirements activity. Indeed, we present in the following the challenges related to the requirements activity [14]:

8.1.Identifying SoS security requirements:

Because requirements are taken on by the constituent systems to meet the SoS objectives, identifying the security requirements for the overarching SoS provides a framework for assessing the adequacy of the system security engineering actions on the part of the constituent systems for security for the SoS and its mission [15]. How to identify these overarching security requirements?

8.2.Security requirements modeling SoS security engineering:

It involves a tension between near-term risk mitigation and long-term evolution to a more secure SoS architecture. In the near term, risks can usually be mitigated effectively by controls at policy domain boundaries and at interfaces between individual systems. In the long term, uniform enforcement mechanisms within and between policy domains not only mitigate risks more effectively but also improve interoperability and maintainability. How can security be integrated into requirements modeling [15]? How can a balance between near-term and long-term security requirements be achieved?

8.3. Ownership:

Who should have the ultimate ownership responsibility for the SoS? Who will be responsible for dealing with issues arising from the SoS, for example if the system was used for malicious purposes, who would be legally culpable? Who will be responsible for testing and proving the system is running as expected and fulfilling its security requirements [16]?

8.4.Risk management:

This is concerned with management and control for the assessment, updating and mitigating of risks [16]. Security-related risks would be part of SoS risk identification and mitigation. They include new security risks resulting from new SoS capabilities composed from interacting constituent systems, as well as any residual security risk of constituent systems [16]. How to identify and mitigate risks associated with end-to-end flow of information and control, without, if possible, focusing on risks internal to individual systems [15]? While there are standards for risk management of standalone systems [17], there are not for SoS. To what extent do they apply to SoS; should such standards be extended to SoS?

8.5. Security of interoperability:

Several important aspects of enterprise interoperability have been the focus of European research programs and initiatives such as European Interoperability Framework (EIF), INTEROP-Vlab and ATHENA-IP. How should these interoperability approaches consider organizational and human factors, such as personal responsibilities (policies and best practice for system security) from the earliest stage of the analysis? How should they address information protection, trust and security [18]?

9. Conclusion

In conclusion, this chapter has provided an overview of the mission-oriented process for System of Systems (SoS) engineering. We have covered the key elements such as SoS definition, characteristics, typology, and engineering activities. Additionally, we have highlighted the challenges of security engineering in the context of SoSs. By understanding these aspects, we can navigate the complexities of SoS engineering more effectively and address security concerns for successful system development and deployment.

Chapter 2 : Risk Assessment

1. Introduction

This chapter explores the critical topic of assessing security risks and requirements within System of Systems (SoS) architectures. As SoS becomes increasingly prevalent in various domains, it is crucial to understand and effectively manage the unique risks associated with these complex, interconnected systems. We will examine the risk assessment process, the state-of-the-art risk assessment methodology provided by the National Institute of Standards and Technology (NIST) Guide, and the significance of risk assessment within the context of SoS. Additionally, we will delve into the System of Systems Security Engineering (SSE) Framework as valuable tools for assessing and addressing security risks in SoS architectures.

2. Risk assessment

Before defining what risk assessment is, it is important to first understand what risk management is given that risk assessment is a sub-process of risk management. Both notions are defined in the following:

2.1.Risk management

Referring to ISO 31000, the risk management process is a “systematic application of management policies, procedures, and practices to the tasks of communication, consultation, establishing the context, identifying, analyzing, evaluating, treating, monitoring and reviewing risk” [19].

2.2.Security risk assessment:

The ISO 27001 describes risk assessment as: “the process of identifying, analyzing, and evaluating risks to the availability, integrity, and confidentiality of information assets, in order to determine the level of risk and make decisions on whether to accept, mitigate, transfer, or avoid those risks and to establish the appropriate level of information security controls”. [20]

A security risk assessment identifies, assesses, and implements key security controls in applications. It also focuses on preventing application security defects and vulnerabilities. Carrying out a risk assessment allows an organization to view the application portfolio holistically—from an attacker’s perspective. It supports managers in making informed resource allocation, tooling, and security control implementation decisions. Thus, conducting an assessment is crucial to establishing and maintaining an effective information security management system (ISMS) based on ISO 27001 requirements.

3. Risk assessment process, a state-of-the-art

The risk assessment methodology outlined in the NIST Guide comprises nine steps. After completing the first step, steps 2, 3, 4, and 6 can be conducted concurrently [21].

3.1. Step 1: System Characterization

It is necessary to gather data that is pertinent to the IT system. An IT system has specific hardware, software, system interfaces, executed processes, data and information, and system and data criticality and data sensitivity. System-related data can be gathered using a variety of methods, including surveys, interviews, document inspections, and automated scanning software [21].

3.2. Step 2: Threat Identification

The second step is to identify potential threats to the identified assets. Threats may be intentional or unintentional and may come from internal or external sources. Some common threats to information systems include malware, phishing attacks, social engineering attacks, natural disasters, power outages, and equipment failure. It is essential to identify all possible threats to ensure that the risk assessment is comprehensive [21].

3.3. Step 3: Vulnerability Identification

Finding the weaknesses that the threats that have been discovered potentially exploit is the third step. Technical or non-technical vulnerabilities can include things like software flaws, configuration mistakes, weak passwords, and a lack of personnel training. To assess the risk that a threat would exploit the flaws and cause harm, it is crucial to understand the vulnerabilities [21].

3.4. Step 4: Control Analysis

In order to decrease the risk that a threat may exploit system vulnerabilities, an organization must analyze the controls it has already put in place or is planning to put in place. The application of controls must be taken into account when calculating the likelihood rating of a potential vulnerability being exploited in the related threat environment. If there is a low degree of threat interest or capacity, or if there are efficient security mechanisms in place that can mitigate the impact, a vulnerability is less likely to be exploited.

Output from Step 4: A list of the IT system’s present and future controls that are used to lessen the risk of a vulnerability being exploited and the effects of such a negative event [21].

3.5. Step 5: Likelihood Determination

The likelihood it is the threat occurrence probability, and this probability of a potential vulnerability being exercised by a particular threat source can be described as:

3.5.1. High: The source of the threat is sufficiently skilled and has strong motivation, and the controls designed to prevent the vulnerability from being exploited are ineffective.

3.5.2. Medium: Although there is a risk from a capable and motivated threat source, there are safeguards in place to mitigate the potential impact of an attempted exploitation of the vulnerability.

3.5.3. Low: The vulnerability is either not attractive enough for the source of the threat to pursue or the attacker lacks the ability to exploit it, or preventative measures have been established to deter or significantly hamper any potential exploitation [21].

3.6. Step 6: Impact Analysis

Determine the negative impact that would result from a successful threat exercise of a vulnerability, and it is divided into three degrees as shown in the following table 1: [21]

Degree of impact	Impact definition
High	Exploiting the vulnerability could: 1-cause the highly expensive loss of significant tangible assets or resources; 2-gravely infringe upon, harm, or obstruct the goals, reputation, or interests of an organization; 3- cause serious harm or death to people
Medium	Exploiting the vulnerability could: 1-may cause the expensive loss of material resources or assets; 2- may damage, or obstruct an organization's mission, reputation, or interest; 3- may cause human injury.
Low	Exploiting the vulnerability could: 1-cause the loss of some material resources or assets; or 2- have an obvious impact on the goals, standing, or interests of an organization.

Table1: Degree of impact and its definition

3.7. Step 7: Risk Determination

This step's goal is to assign a risk score based on the possibility that the threat will materialize, taking into account the controls currently in place and the potential effects on the organization if the threat was successful in exploiting a vulnerability.

Healthcare organizations can prioritize resources and concentrate on the regions with the most risk by assessing the hazards [21].

3.8. Step 8: Control Recommendations

It involves putting in place controls that could reduce or completely get rid of identified risks to a reasonable level. The recommended controls are determined by a number of variables, including their efficiency, organizational policies, operational impact, safety, and reliability. The risk mitigation process, which involves the evaluation, prioritization, and implementation of procedural and technical security controls, uses these control recommendations. However, an organization might not be able to implement all recommended controls. Consequently, a cost-benefit analysis should be carried out to support the costs of putting controls in place with the decrease in risk level. The viability and operational effects of implementing the suggested controls should also be assessed. The result of this step is a recommendation for controls and other risk-mitigation strategies [21].

3.9.Step 9: Results Documentation

It consists of documenting all the results of the risk assessment that are completed (Threat-sources and vulnerabilities identified, risks assessed, and recommended controls provided) [21].

4. Risk Assessment in context of SoS

Several papers were proposed to deal with risk assessment in the context of SoSE, in what follows, we present the most important ones:

- SSE Risk-Based System Analysis Methodology.
- OASoSIS.

4.1.SOS SSE Framework

A five-stage continuous Security Engineering (SE) process is done to examine SoS from a mission viewpoint, identify essential components for security risk to mission objectives, and address those componentsSoS from a mission viewpoint, identify essential components for security risk to mission objectives, and address those components. It places SSE in a context with SE and SoS. Figure 3 presents a picture of the framework.

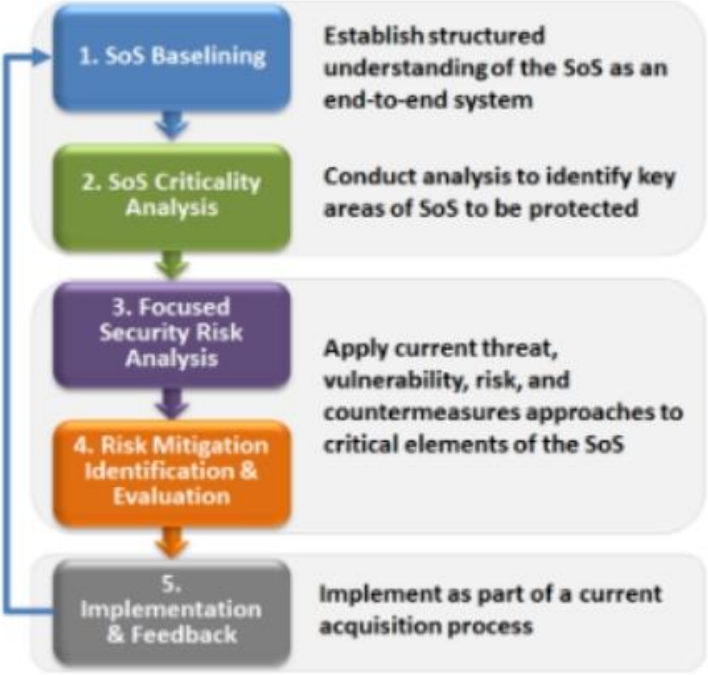

1. SoS Baselining
Establish structured understanding of the SoS as an end-to-end system
2. SoS Criticality Analysis
Conduct analysis to identify key areas of SoS to be protected
3. Focused Security Risk Analysis
4. Risk Mitigation Identification & Evaluation
Apply current threat, vulnerability, risk, and countermeasures approaches to critical elements of the SoS
5. Implementation & Feedback
Implement as part of a current acquisition process

Figure 3: System of Systems Security Engineering Framework [22]

4.1.1. SoS baselining

Baselining is a crucial technique for forensics, audit, incident response, and troubleshooting. It involves recording the characteristics of a system's recognized. Ideal state. This may be used in the present system state to do a comparison study to identify what has changed and how it has changed. Its goal is to recognize the existing configuration of SoS components and their function in supporting infrastructure, linkages, and interfaces for mission execution.

Mission resilience against persistent threats is being addressed by an increasing range of strategies. An awareness of the existing "brownfield" mission scenario is necessary in order to effectively use them. When SoSE is present, it may be simple; if not, investments might be necessary.

Their procedure is to understand the concepts of the task and its outcomes, including the end jobs and performance measures and descriptions of existing systems, linkages, and relations; The dynamics of SoS environments that underpin mission outcomes. And from that, we conclude that the result or product is a technological framework for study of crucial factors, security threats, and mediations [22].

Various methods of establishing and displaying SoS/Mission are:

- Baselining tools based on OT system data and mission threads

- Activities and sequential linkages are represented using BPM approaches based on standards.
- Architecture tools for representing systems and interactions, such as DoDAF
- Model-based ways to model SoS elements, actions, and connections (e.g., UML, SysML).

4.1.2. SoS Criticality Analysis

The goal is to identify the key SoS components needed for mission success, regardless of potential risks. Mission systems, links and interfaces, and supporting infrastructure, Helps align mission objectives and protection priorities. Since end-to-end SoS cannot be completely protected, need a method to recognize crucial SoS components and control complexity. Explain existing systems, linkages, and their relationships; describe SoS dynamics; describe settings that support mission results, among other representation and analytic techniques [23].

SoS Criticality Analysis consists of three interacting activities as presented in figure 4:

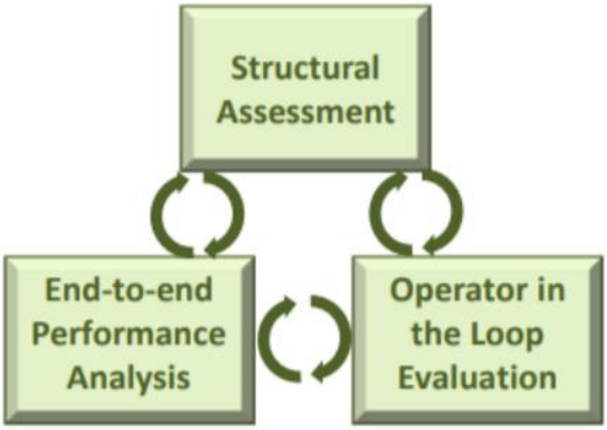

Structural Assessment
End-to-end Performance Analysis
Operator in the Loop Evaluation

Figure 4: Methods Supporting SoS Criticality Analysis [24]

- **Structural assessment:** to recognize critical elements and how they relate to one another.
- **End-to-end performance analysis:** to comprehend SoS behavior and the impact of loss, invasions, or interruptions to essential elements on mission results
- **Operator in the loop:** Evaluation for obtaining a real perspective on important factors

4.1.2.1. Structural assessment:

As a starting point, determine SoS components that are obviously essential or not to the purpose.

Their procedure is to define the overall system flows and dependencies needed to perform the task (based on SoS rules), clearly identify the elements on the critical path for mission success based on the SoS architecture review, and identify the components that can be excluded from a critical path based on minimal dependencies, redundancy, etc... [24].

As a result of the initial identification of SoS components needed for mission success

*Several methods/tools appropriate for structural assessment:

- Learning from operations and user inputs; alternative techniques of validation are needed.
- Analysis of flows and pathways across nodes and relationships is supported by BPMs
- DoDAF data regarding linkages between and among mission elements.

FDNA (Functional Dependency Node Analysis) or other approaches to model and analyze the operational performance of a mission network if one or more entities deteriorate or fail, such as tools like System Architect for the study of SoS components.

4.1.2.2. End End-to-end performance analysis

Understand SoS behavior and the impact of critical element loss, intrusions, or interruptions on mission results.

Identifying an acceptable model or simulation to depict missions, a set of scenarios that accurately reflect the mission environment, and metrics for mission success and effectiveness are the first steps in this research. Agent-based models, discrete event simulations, and other operations/systems analysis settings that are utilized to address additional mission-level concerns in specific mission domains may be the environments used for this study [24].

The study starts with the mission objectives, including performance and effectiveness metrics. In order to simulate the mission in a chosen set of scenarios, the analyst represents the entire mission thread, including systems and their behaviors, in a realistic operational context. The analyst then runs a series of excursions, starting with a base case, to evaluate the nominal performance and effectiveness of the mission. The research also includes a number of excursions where crucial SoS components are altered to assess the effects on mission effectiveness and performance. Instead of simulating the threat to the elements, the objective is to presume that they are in danger and examine how that will affect mission outcomes [24].

Where there are a significant number of crucial SoS components, an analysis of experiments may be carried out to determine the number of outings required to pinpoint the crucial components based on the outcomes of the mission. To facilitate these investigations, facilities like the MITRE Elastic Goal-Directed Simulation Framework tool (is a tool to enhance already-existing simulation applications by giving them access to grid- and cloud-based execution, sophisticated Design of Experiments techniques like simulation-based optimization, and reliable data processing and visualization.) may be used to support these analyses [23].

Results of the initial end-to-end performance analysis may point to the necessity of additional structural analysis or offer the information required for specific structural analysis approaches (e.g. FDNA). A proposed set of priority SoS components with a knowledge of the mission consequences of impacts to these elements is the end result of the completed study and will be evaluated for the security risk to the mission [23].

4.1.2.3. Operator in the Loop Evaluation

Evaluate the essential components of the SoS in an operational setting and collect knowledge that can only be acquired by interacting directly with system users.

The analyst gathers and analyzes data on the essential aspects indicated in the structural and performance studies, taking into account the human components of the operation, in a SIMEX, operational exercise, or even by observations from operations. Other analytical methods could miss important operational environment components that are crucial to comprehending the dynamics surrounding potential vital SoS aspects. This strategy may involve anything from simple observations to formal human-in-the-loop investigations. The findings offer information and insights to evaluate the potential components of the SoS and may point to the necessity for additional structural or performance assessments [25].

4.1.3. Focused Security Risk Analysis

Its goal is to evaluate whether or if mission-critical components are actually in danger or are sufficiently safeguarded. Infrastructure systems, linkages, and interfaces that support the mission.

Its approach is:

- ➢ Use current threat, vulnerability, and effect analysis approaches at the system level.
- ➢ Threat assessment identifies risks to a crucial element in the context of the specific mission.
- ➢ Vulnerability assessment assesses an element's level of threat protection utilizing PPP findings from the tests.

Results/Products: Determination of the type and severity of security threats for all major system elements and the basis for selecting priority areas to further ensure mission/results.

4.1.4. Risk Mitigation Identification and Evaluation

Its goal is to determine, assess, and suggest a number of improvements to the SoS for risk reduction.

Its approach is: Analyze risk mitigation choices and assess their potential effects on mission results, technical viability, cost, and other factors, such as dependencies between composite solution possibilities.

Results/Products: Develop a composite set of system improvements to enhance SoS security and fulfill mission objectives [25].

4.1.5. Implementation and Feedback

Its goal is to implement system improvements derived from the earlier phases to enhance mission results. Incorporates designing, implementing, integrating, and testing modifications and their effects on mission assurance and the SoS. Feedback is a continuous process that is often completed as part of system development, upgrade, or technology refresh [25].

Its approach is:

- Implementation is a standard aspect of the activities involved in purchasing a system.
- Monitoring implementation for problems that might affect SoS is the second SoS-level action.

A revised SoS baseline reflects system changes.

4.2. Assessing System of Systems Security Risk and Requirements with OASoSIS:

OCTAVE Allegro for System of Systems (OASoSIS) is an information security risk assessment and modeling process, to assist risk-based decision making in SoS Requirements Engineering, the steps of OASoSIS are:

4.2.1. Identify SoS context, structure, stakeholders, roles, goals, and dependencies:

The first step in the OASIS version of OCTAVE Allegro is to identify the context, structure, stakeholders, roles, goals, and dependencies of the system of systems. This includes identifying the components of the system of systems, the stakeholders who are involved, their roles and responsibilities, and the goals that the system of systems is intended to achieve. This step also involves identifying the dependencies between the components of the system of systems [26].

4.2.2. Establish risk measurement criteria

The second step is to establish the risk measurement criteria that will be used to assess the risks associated with the system of systems. This includes identifying the types of risks that are relevant to the system of systems, such as cyber-attacks, natural disasters, or human error. It also involves defining the risk measurements criteria, such as likelihood and impact, that will be used to assess the risks [26].

4.2.3. Develop information asset profile

The third step is to develop an information asset profile for the system of systems. This involves identifying the critical assets and information that are stored and processed by the system of systems. It also involves categorizing the assets and information based on their sensitivity and criticality [26].

4.2.4. Identify information asset containers

The fourth step is to identify the information asset containers that are used to store and process the information assets. This includes identifying the hardware and software components that are used to manage the information assets, as well as the networks and systems that are used to communicate and transfer the information [26].

4.2.5. Identify areas of concern with threat scenarios, and identify vulnerabilities:

The fifth step is to identify the areas of concern that are associated with the system of systems. This includes identifying the potential threat scenarios that could impact the system of systems, such as cyber-attacks or natural disasters. It also involves identifying the vulnerabilities that exist within the system of systems, such as outdated software or weak passwords [26].

4.2.6. Identify risks

The sixth step is to identify the risks that are associated with the system of systems. This involves analyzing the potential consequences of the threat scenarios and vulnerabilities that have been identified. It also involves estimating the likelihood of the risk events occurring [26].

4.2.7. Analyze risks

The seventh step is to analyze the risks that have been identified. This involves assessing the likelihood and impact of the risks, as well as identifying the controls and countermeasures that can be implemented to mitigate the risks [26].

4.2.8. Prioritize critical risks, Model and visualize SoS risks, and Select mitigation approach to risks:

The final step is to prioritize the critical risks that have been identified. This involves selecting the risks that pose the greatest threat to the system of systems and developing mitigation strategies to address those risks. It also involves modeling and visualizing the risks associated with the system of systems to help stakeholders understand the risks and the potential consequences. Finally, it involves selecting the mitigation approach to the risks, which may include implementing technical controls, developing policies and procedures, or providing training and awareness to personnel [26].

5. Discussion

Assessing security risks and requirements within System of Systems (SoS) architectures is of paramount importance in today's interconnected world. The risk assessment process here's a comparative table of the three risk assessment approaches discussed in this chapter:

Aspect	*NIST Guide Risk Assessment*	*SoS SSE Framework*	*OASoSIS*
Scope and Focus	Individual IT systems	Complex System of Systems (SoS)	SoS Requirements Engineering
Steps and Approach	9 sequential steps	5 continuous stages	8 steps
Application Domain	IT systems (hardware, software)	Complex SoS environments	System of Systems (SoS)
Outputs and Results	Risk assessment report	Insights into mission resilience, critical SoS components, and risks	Prioritized critical risks with risk mitigation recommendations
Complexity and Scope	Individual systems, straightforward scope	Complex SoS environments involving interactions & dependencies	SoS-specific risk assessment, considering information assets

Table 2: The comparison of NIST Guide Risk Assessment, SoS SSE Framework and OASoSIS

6. Conclusion

Throughout this chapter, we have examined the importance of assessing security risks and have explored various processes that have been proposed for both individual systems and complex systems of systems. The NIST recommends the use of several steps for systems.

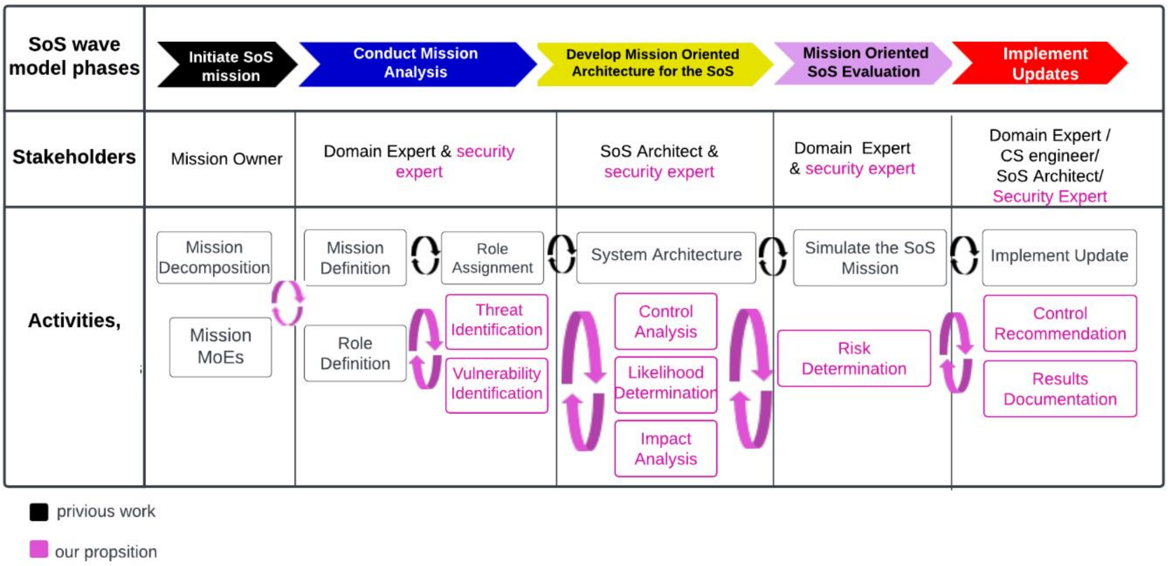
When it comes to systems of systems, the complexity increases as multiple interconnected systems are involved. Therefore, a different approach is required. That on what we will focus in the next chapter.

1. Introduction:

This chapter explores the integration of NIST risk assessment steps within a mission-oriented process in Systems of Systems (SoS) engineering, focusing on the context of crowd management. It presents a case study that demonstrates the practical implementation of these steps and provides a detailed description of each step within the SoS crowd management scenario. The chapter also introduces a proposed Metamodel that enhances risk assessment capabilities in the context of crowd management. Additionally, it presents a concrete syntax, utilizing graphical notation, for identifying vulnerabilities and threats and also presents the final user of the application. The chapter aims to provide valuable insights into applying NIST risk assessment steps in the field of SoS engineering for crowd management.

2. Integrating NIST Risk Assessment Steps for Mission-Oriented Process in Systems of Systems Engineering:

To propose a complete process that aligns the SoS mission objectives and the security risk assessment, we combine the MoP-SoSE engineering activities with the NIST risk assessment steps, as shown in Figure 4. Each activity is described in what follows:



SoS wave model phases	Initiate SoS mission	Conduct Mission Analysis	Develop Mission Oriented Architecture for the SoS	Mission Oriented SoS Evaluation	Implement Updates
Stakeholders	Mission Owner	Domain Expert & security expert	SoS Architect & security expert	Domain Expert & security expert	Domain Expert / CS engineer/ SoS Architect/ Security Expert
Activities,	Mission Decomposition Mission MoEs	Mission Definition Role Assignment Role Definition Threat Identification Vulnerability Identification	System Architecture Control Analysis Likelihood Determination Impact Analysis	Simulate the SoS Mission Risk Determination	Implement Update Control Recommendation Results Documentation

Figure 4: Integrating NIST Risk Assessment Steps for Mission-Oriented Process in Systems of Systems Engineering

2.1.Mission decomposition:

This task aims to give a practical, coarse perspective of the mission. This is accomplished by analyzing the overarching mission goals in order to recursively pinpoint more specific sub-mission goals. Finding a procedure that can complete a certain sub-mission is the requirement for terminating the mission decomposition. In order to break down the primary mission into submissions, establish context-dependent variation points, and identify mission risks, we have designed a profile that extends the SysML requirement diagram. As a result, a mission-functional model of the SoS is produced in this step [12].

2.2.Mission Measurements of Effectiveness (MoEs):

Using the SysML parametric diagram, the mission owner defines efficacy measures in this activity. These indicators will be employed to evaluate the overall effectiveness of the SoS mission [12].

2.3.Mission definition:

The design of the operational view of the mission is what this activity aims to accomplish. The definition of mission threads and activities makes up this part. By employing activities, it produces a fine-grained behavioral perspective of sub-missions. The SysML activity diagram is used to expand the view. The mission functional model's refine relationship links each sub-mission to a specific activity. The criteria for terminating activity decomposition is when a sub-activity corresponds to a role capacity that we call action. Complex activities can be broken down into sub-activities [12].

2.4.Role assignment:

Activities are made up of actions that match role capabilities, as is well known. Multiple capabilities make up the role, and the same capacity might be found in multiple roles. Therefore, the purpose of this phase is to specify the role that must be connected to each of an activity's actions. Through the designated role, this association establishes a connection with the component system [12].

2.5.Role definition:

The position serves as an abstract depiction of the hierarchy of entities with capacities that make it possible to accomplish the mission. Roles may offer or demand certain capabilities, allowing for the composition of roles. A SysML profile that extends the block definition diagram serves as the foundation for the role definition model that was created [12].

2.6.Threat identification:

Threat identification entails finding possible threats that might take advantage of weaknesses and affect the system.

The input for threat identification includes system characterization information, system boundaries, critical assets, and functions. Additionally, documentation regarding the system's architecture, design, and operational context is taken into account. Input from stakeholders, subject matter experts, and security professionals is also considered. Furthermore, information about historical or industry-specific threats and attack patterns is taken into consideration during the process.

Here are some important factors to think about:

- **Internal and External Threats**: Take into account both internal and external threats. Insiders with malicious intent, displeased workers, or unintentional activities by authorized people are all examples of internal dangers. Hackers, attackers, natural disasters, and other external entities can all pose a threat from the outside.
- **Threat Sources**: Identify the potential sources of threats. It can be made up of individuals, groups, states, or machines.
- **Understanding the motivations behind threats** is important. They can be doing it for personal benefit, business gain, political gain, competitive advantage, retaliation, or just general disturbance.
- **Threat Capabilities**: Evaluate the prospective threats capacities. Consider the technological expertise, abilities, tools, and resources at their disposal.
- **Threat Methods:** Recognize the strategies and tactics that threats may use. This can involve illegal access, denial of service, virus attacks, social engineering, etc.

The output of threat identification includes a list of identified threats and their characteristics. It provides an understanding of the motives, capabilities, and methods of potential threat sources. This output serves as input for further analysis, including vulnerability assessment and determining the likelihood of occurrence.

2.7.Vulnerability identification:

Vulnerability assessment focuses on identifying and assessing vulnerabilities or weaknesses within the system.

The input for vulnerability identification includes system characterization information, critical assets, resources, and functions. It also involves technical documentation such as system architecture, design specifications, and configuration details. Additionally, knowledge of system components, software, networks, and infrastructure is considered. Input from subject matter experts, security professionals, and stakeholders is also taken into account during the process.

Here are some key aspects to consider:

Technical and Non-Technical Vulnerabilities: Identify both technical vulnerabilities (e.g., software vulnerabilities, misconfigurations, weak passwords) and non-technical vulnerabilities (e.g., lack of policies, inadequate training, physical security weaknesses).

Vulnerability Scanning: Use automated tools or manual techniques to scan the system and identify known vulnerabilities. This can include vulnerability scanners, penetration testing, code reviews, or configuration reviews.

Vulnerability Rating: Assess the severity and potential impact of each vulnerability. Assign a rating or score to prioritize vulnerabilities based on their potential risk.

Likelihood of Exploitation: Determine the likelihood of each vulnerability being exploited. Consider factors such as the presence of active threats, ease of exploitation, accessibility, and existing safeguards.

Vulnerability Prioritization: Prioritize vulnerabilities based on their severity, exploitability, and potential impact on the system's assets and functions.

Emerging Vulnerabilities: Stay updated with the latest security advisories, vulnerability databases, and industry alerts to identify emerging vulnerabilities that may not have known patches or mitigation strategies.

The output of vulnerability identification includes a list of identified vulnerabilities and their characteristics. It also involves assessing the likelihood of each vulnerability being exploited and the potential impact if exploited. This output serves as input for further analysis, including control analysis and determining the overall risk.

2.8.System architecture:

The Systems Architect is responsible for designing the comprehensive system architecture that integrates individual subsystems and components. They define the structure, interfaces, and interactions between subsystems to ensure seamless operation and interoperability. Additionally, they are involved in implementing updates or changes to the system architecture as the project progresses. They assess the impact of any changes and make necessary adjustments to ensure the system remains aligned with the mission requirements [12].

2.9.Control analysis:

Control analysis involves the evaluation of existing security controls and safeguards within the system. It considers inputs such as system documentation and a control inventory. The process includes assessing the adequacy and effectiveness of each control, analyzing their efficiency in reducing risks, and identifying any control gaps or deficiencies. The outputs of control analysis consist of a report detailing the findings, including the evaluation of each control and recommendations for additional controls or enhancements.

2.10. Likelihood determination:

Likelihood determination assesses the probability of a threat exploiting a vulnerability. It takes into account inputs from the threat assessment and vulnerability assessment. The process involves evaluating likelihood factors such as historical data, expert judgment, and environmental conditions. A likelihood rating or score is assigned to each threat-vulnerability pair, leading to outputs such as a likelihood assessment and a summary of likelihood ratings. These outputs provide insights into the potential frequency of risk occurrences and help prioritize risks for further analysis and decision-making.

2.11. Impact analysis:

Impact analysis involves assessing the potential consequences of a realized threat on the system. Key aspects to consider include identifying critical assets, determining impact factors such as confidentiality, integrity, availability, financial, reputational, and safety impacts, and evaluating the magnitude of these impacts. Additionally, considering dependencies between assets and potential chain reactions or indirect impacts within the system is crucial. The goal is to understand the extent of harm or damage that could occur to guide risk mitigation strategies and decision-making.

2.12. Risk determination:

risk determination involves assessing the likelihood and impact of threats to determine overall risk levels. This is done by evaluating factors such as threat sources, motives, capabilities, historical data, and environmental conditions. The assessments are combined using a risk matrix or similar tool to categorize risks based on their likelihood and impact ratings. Risks are then prioritized, with high likelihood and high impact risks receiving the most attention. The identified risks and their associated information are documented to guide risk management activities and decision-making processes.

2.13. Control recommendation:

Control recommendation involves developing appropriate measures to address identified risks and mitigate them to an acceptable level. Key considerations include evaluating risk treatment options, selecting controls based on industry standards and guidelines, assessing control effectiveness, conducting cost-benefit analysis, creating an implementation plan, and establishing monitoring and review mechanisms. The goal is to propose effective controls that can reduce risks and ensure their continued effectiveness over time.

2.14. Results documentation:

Results documentation involves documenting the outcomes of the risk assessment process, including identified risks, risk levels, and control recommendations. Key considerations include maintaining a risk register or database, providing clear risk descriptions with relevant context, documenting risk levels using rating scales, detailing recommended controls and their rationale, describing risk treatment plans, including supporting documentation and regularly reviewing and updating the documentation. This documentation serves as a valuable reference for risk management activities and decision-making processes, ensuring accuracy and accessibility for stakeholders.

2.15. Simulate the SoS:

To determine whether a configuration of an architecture is capable of carrying out the defined SoS mission, simulation of models is required. Additionally, it enables mistake detection and performance confirmation [12].

2.16. Implement update:

Updates can be made either at the level of the CSs or the level of the SoS models. A CS's individual updates adhere to the system engineering life cycle. The ADE merely needs to work with the CS engineers to change those things [12].

3. Case study presentation:

The use case study of football crowd management examines the strategies and practices implemented to ensure the safety, security, and satisfaction of fans at football events. This study analyzes crowd control techniques, infrastructure planning, communication protocols, and emergency response systems. By studying real-world scenarios, the use case aims to inform the development of effective crowd management strategies that enhance safety, incorporate technology, promote stakeholder collaboration, and prioritize the fan experience [27].

In the figure 5 we present demonstrative images of the crwd management use case study:

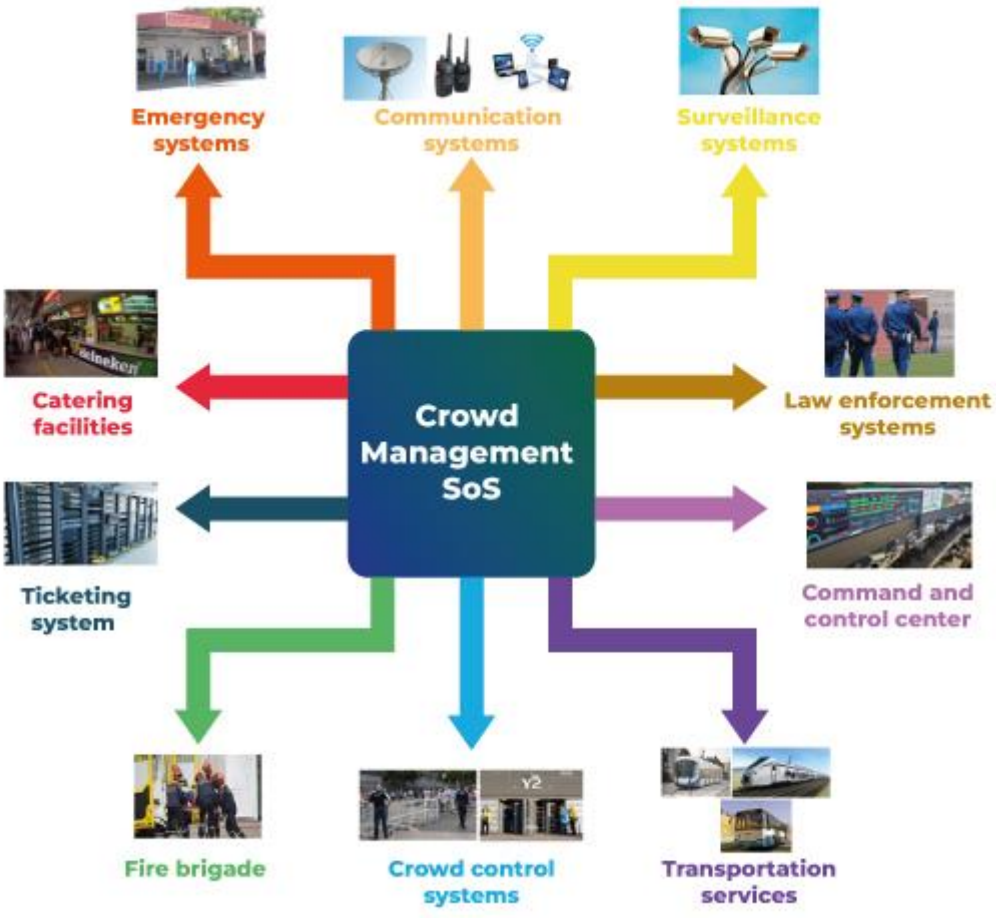


Figure 5: Crowd Management Case Study

Football crowd management presents a complex and mission-oriented System of Systems (SoS) challenge, requiring an integrated approach to ensure the safety, security, and smooth operation of football events. Within this context, the primary mission is to manage and control the movement, behavior, and well-being of a large number of spectators in a stadium environment. The SoS perspective recognizes that crowd management extends beyond a single entity and involves multiple interconnected systems, including law enforcement systems, command and control center, infrastructure, ticketing system, communication systems, emergency response mechanisms, and transportation services. By integrating these systems and aligning them with the overarching mission of maintaining crowd safety and enjoyment, stakeholders can develop comprehensive strategies for risk assessment, incident prevention, and rapid response coordination.

4. Steps description in context of SoS crowd management case study:

In this section, we delve into a comprehensive description of the Systems of Systems (SoS) crowd management case study:

- **Mission Decomposition**: The mission of football crowd management can be decomposed into various sub-missions, such as crowd control, behavior monitoring, emergency response, and infrastructure management. Each sub-mission focuses on specific aspects of managing the crowd to ensure safety, security, and smooth operation during football events.

- **Mission Measurements of Effectiveness (MoEs):** To assess the effectiveness of the crowd management mission, several MoEs can be defined. These may include metrics such as the average response time to incidents, the percentage of successfully controlled crowd disturbances, the number of injuries or accidents, and the overall satisfaction level of spectators. MoEs provide measurable criteria to evaluate the success and efficiency of the mission.

- **Mission Definition:** The mission of football crowd management is to manage and control the movement, behavior, and well-being of a large number of spectators in a stadium environment. It encompasses activities related to crowd control, safety, security, and incident response during football events.

- **Role Assignment:** Various stakeholders and entities are assigned specific roles in the football crowd management mission. These roles may include stadium security personnel, law enforcement agencies, event organizers, medical staff, communication teams, and transportation service providers. Each role is responsible for specific tasks and functions within the overall mission.

- **Role Definition:** Each assigned role in the football crowd management mission has specific responsibilities and duties. For example, stadium security personnel ensure access control, monitor crowd behavior, and respond to incidents. Law enforcement agencies maintain public order, handle potential threats, and enforce relevant laws. Clear role definitions help streamline coordination and ensure effective execution of the mission.

- **Threat Identification:** Threat identification is a critical aspect of football crowd management missions, as it involves recognizing potential risks and hazards that could pose a threat to the safety, security, and smooth operation of the event. These threats encompass a range of possibilities, including unruly behavior, overcrowding, unauthorized access, terrorism, medical emergencies, and natural disasters. By proactively identifying these threats, appropriate preventive and response measures can be implemented to ensure the effective management of the crowd and the overall safety of the event. This comprehensive approach covers both internal threats, such as unruly behavior and overcrowding, as well as external threats like terrorism and hooliganism, which may disrupt crowd management efforts.

- **Vulnerability Identification:** Vulnerabilities within the football crowd management system are identified to assess potential weaknesses that could be exploited by threats. These vulnerabilities may include inadequate communication systems, insufficient security measures, poorly designed infrastructure, or lack of coordination between entities. Identifying vulnerabilities enables the development of mitigation strategies to strengthen the overall system.

- **System Architecture:** The football crowd management system consists of multiple interconnected systems, such as law enforcement systems, command and control centers, infrastructure, ticketing systems, communication systems, emergency response mechanisms, and transportation services. The system architecture defines how these components interact and work together to achieve the mission objectives.

- **Control Analysis:** Control analysis involves evaluating the effectiveness of existing controls and measures in mitigating risks and addressing threats. It assesses whether the current control mechanisms are sufficient or require improvements to ensure the safety, security, and smooth operation of football events.

- **Likelihood Determination:** Likelihood determination assesses the probability of specific threats occurring during football events. It considers historical data, intelligence reports, and expert opinions to estimate the likelihood of various threats materializing. This information helps prioritize risks and allocate resources accordingly.

- **Impact Analysis:** Impact analysis examines the potential consequences of threats and incidents on the crowd management mission. It evaluates the severity of possible disruptions, such as injuries, property damage, public panic, or reputational harm. Understanding the potential impacts aids in developing response strategies and allocating resources effectively.

- **Risk Determination:** Risk determination combines the likelihood and impact assessments to determine the overall risk level associated with specific threats. It helps prioritize risks and allocate resources based on their significance and potential consequences. High-risk threats require greater attention and mitigation efforts.

- **Control Recommendation:** Based on the identified risks, vulnerabilities, and control analysis, recommendations for new or enhanced controls are proposed. These recommendations aim to mitigate risks, strengthen the system, and improve the overall effectiveness of the football crowd management mission.

- **Results Documentation:** Throughout the risk assessment process, documentation is crucial. This includes recording the identified threats, vulnerabilities, risk levels, control recommendations, and the rationale behind decision-making. Results documentation ensures transparency, facilitates communication among stakeholders, and provides a reference for future assessments and improvements.

- **Simulate the SoS:** Simulation techniques can be employed to model the football crowd management SoS and assess its behavior under different scenarios. Simulations help evaluate the effectiveness of control measures, identify potential bottlenecks or vulnerabilities, and test the responsiveness of the system to various incidents. By simulating the SoS, stakeholders can gain insights into system performance and make informed decisions.

- **Implement and Update:** The risk assessment findings and control recommendations are implemented within the football crowd management lifecycle. This includes updating processes, protocols, and technologies, as well as training personnel on new control measures. The risk assessment process should be ongoing, with regular updates and adjustments to adapt to evolving threats and changing circumstances in order to ensure continuous improvement and effective risk management.

5. The proposed Metamodel:

As part of our six-month academic curriculum, we have to carry out a risk assessment project in context of SoS. However, we know that this process is complex and requires several steps. Therefore, we decided to focus on the two most important steps: threat identification and vulnerability identification. These two steps will allow us to determine the potential risks to which our system is exposed and the measures to take to reduce or avoid them. In the figure 6 we present our proposed metamodel the Abstract syntax for vulnerability and threats identification:

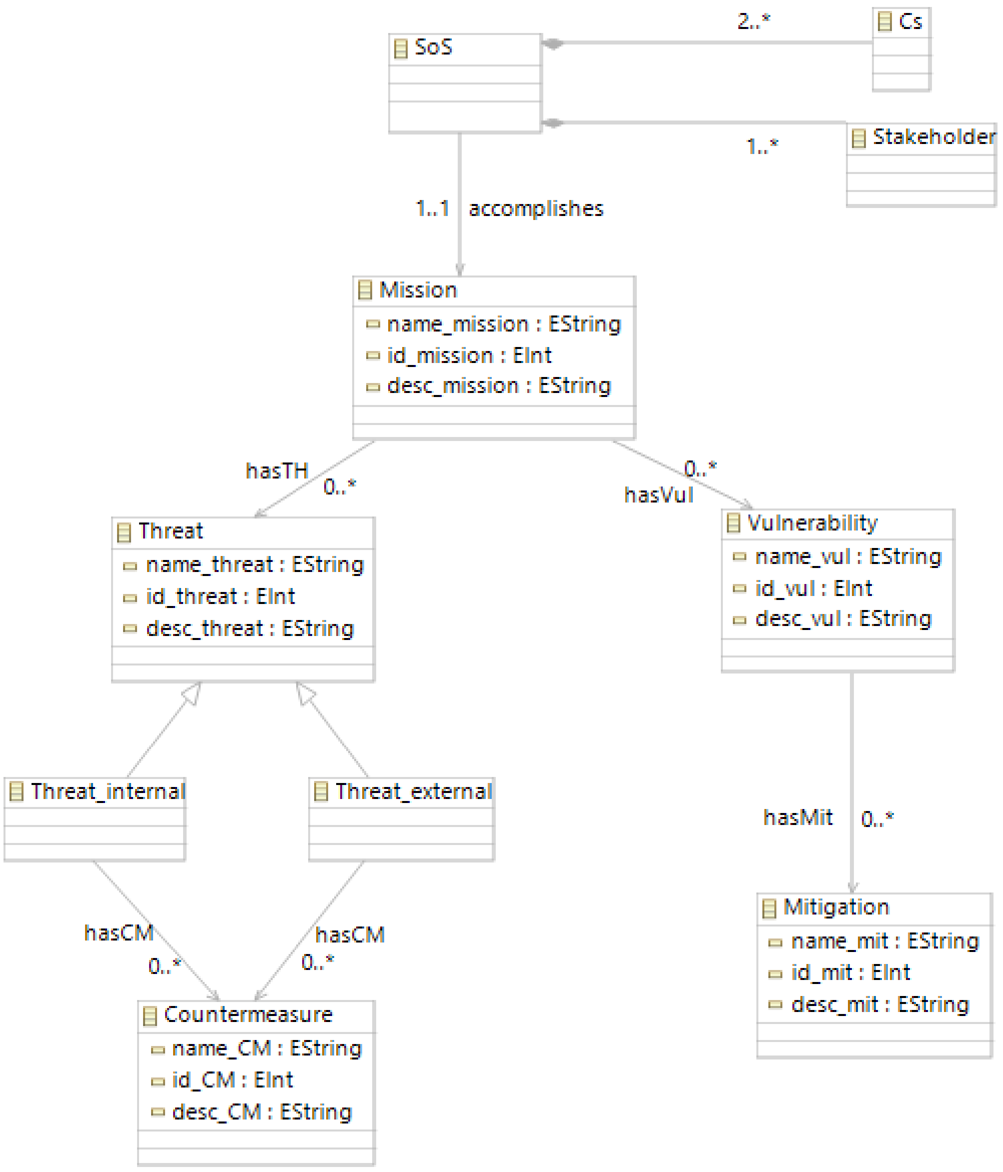


Figure 6: The proposed Metamodel of threat and vulnerability identification.

We will present in the next table each meta class attributes, type of each attribute and its designation, role of the meta class and the links between the meta classes.

MetaClass	Attribute	Type	Designation	Roles	Links
SOS (System of Systems)				Represents a specific objective or task to be accomplished by the System of Systems (SoS)	* Association from SOS: Indicates that a mission is associated with a System of Systems (SoS). * Associations to Threat External, Threat Internal, and Vulnerability: Represents the dependencies between missions and potential threats or vulnerabilities.
CS (Constituent System)				Refers to a system that is part of a larger system or System of Systems (SoS)	* Aggregation from SoS: Represents that a constituent system is a part of or associated with the System of Systems (SoS).
Stakeholder				Represents an individual, group, or organization that has an interest or concern in the system of interest.	* Aggregation from SoS: Indicates that a stakeholder is associated with or part of a System of Systems (SoS)
Mission	Name_mission	String	Name of mission	Represents a specific objective or task to be accomplished by the System of Systems (SoS) or its constituent systems.	* Association from SoS: Indicates that a mission is associated with a System of Systems (SoS). * Associations to Threat and Vulnerability: Represents the dependencies between missions and potential threats or vulnerabilities.
	Id_mission	Int	ID of mission		
	Desc_mission	String	Description of mission		
Threat	Name_threat	String	Name of threat	Represents a potential danger or harm that can affect the Mission	* Inheritance from Threat External and Threat Internal: Indicates that Threat is a superclass and has two subclasses: Threat External and Threat Internal. * Associations to Countermeasure: Represents the countermeasures or actions taken to mitigate the identified threats.
	Id_threat	Int	ID of threat		
	Desc_threat	String	Description of threat		
Threat internal				A threat internal represents an internal	* Inheritance from Threat: Indicates that Threat

				factor or element that poses a risk or potential harm to the System of Systems (SoS) or its constituent systems.	Internal is a subclass of Threat. * Association to Countermeasure: Represents the countermeasures or actions taken to mitigate the identified internal threats.
Threat external				Represents an external factor or entity that poses a risk or potential harm to the System of Systems (SoS) or its constituent systems.	* Inheritance from Threat: Indicates that Threat External is a subclass of Threat. * Association to Countermeasure: Represents the countermeasures or actions taken to mitigate the identified external threats.
Countermeasure	Name_CM	String	Name of countermeasure	Refers to a specific action, process, or mechanism implemented to prevent, mitigate, or reduce the impact of a threat internal or external.	* Associations from Threat External and Threat Internal: Represents the countermeasures associated with mitigating specific threats.
	Id_CM	Int	ID of countermeasure		
	Desc_CM	String	Description of countermeasure		
Vulnerability	Name_vul	String	Name of vulnerability	Represents a weakness or flaw in the System of Systems (SoS) or its constituent systems that could be exploited by threats.	* Association from Mission: Indicates the vulnerabilities associated with a particular mission. * Association to Mitigation: Represents the mitigation strategies or actions taken to address the identified vulnerability.
	Id_vul	Int	ID of vulnerability		
	Desc_vul	String	Description of vulnerability		
Mitigation	Name_mit	String	Name of mitigation	Refers to the process of reducing, minimizing, or eliminating the potential impact of a vulnerability or risk	* Association from Vulnerability: Indicates the mitigation strategies or actions associated with a specific vulnerability.
	Id_mit	Int	ID of mitigation		
	Desc_mit	String	Description of mitigation		

Table 2: Descriptive table of Meta Classes, their attributes, roles, and links.

6. Concrete syntax (graphical notation) for vulnerability and threats identification:

In the next figure 7 we will represent a graphical notation of threat and vulnerability determination of two missions, the first one is Emergency response and the second is Ticketing:

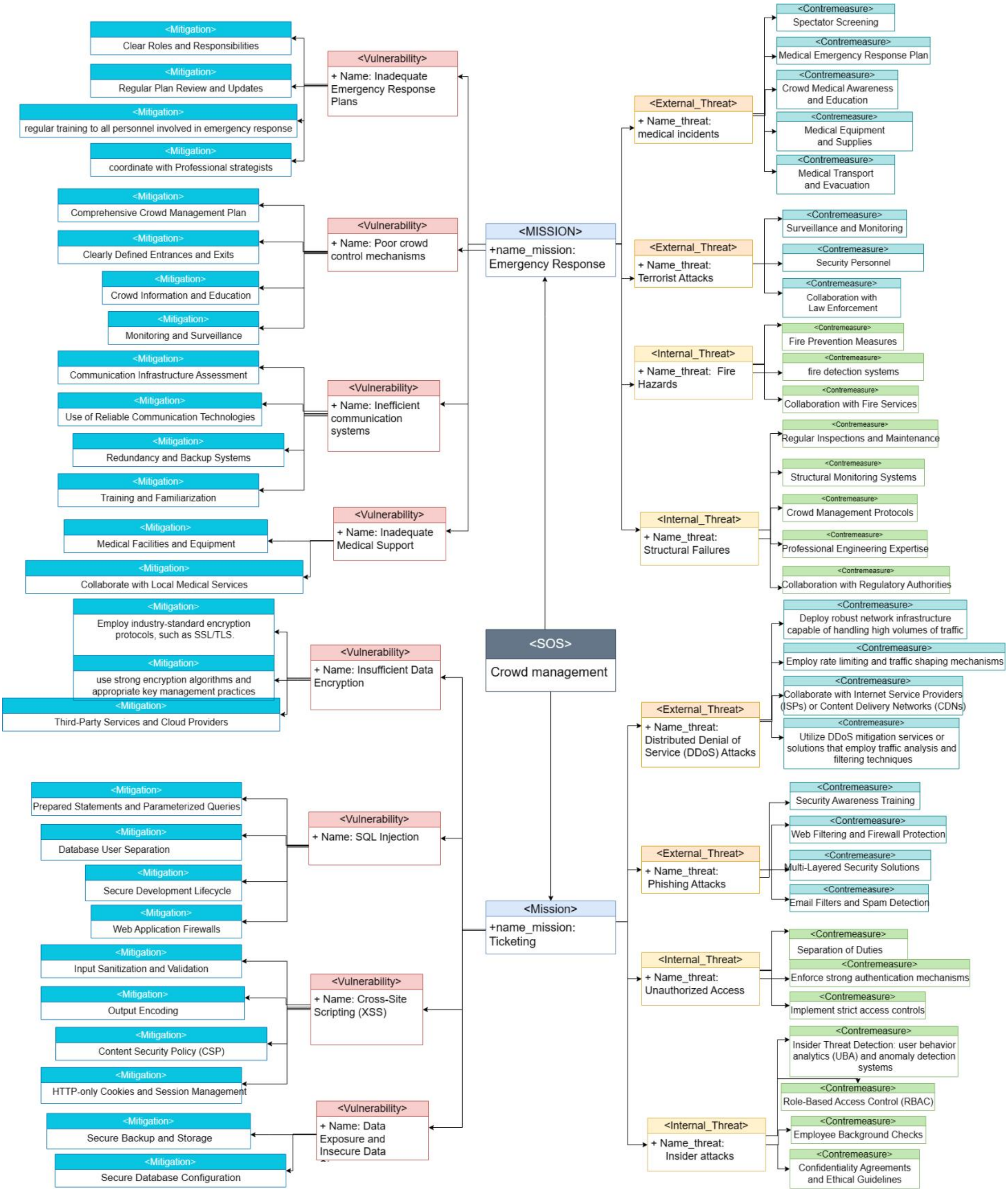

<Mitigation>
Clear Roles and Responsibilities
<Mitigation>
Regular Plan Review and Updates
<Mitigation>
regular training to all personnel involved in emergency response
<Mitigation>
coordinate with Professional strategists
<Vulnerability>
+ Name: Inadequate Emergency Response Plans
<Mitigation>
Comprehensive Crowd Management Plan
<Mitigation>
Clearly Defined Entrances and Exits
<Mitigation>
Crowd Information and Education
<Mitigation>
Monitoring and Surveillance
<Vulnerability>
+ Name: Poor crowd control mechanisms
<Mitigation>
Communication Infrastructure Assessment
<Mitigation>
Use of Reliable Communication Technologies
<Mitigation>
Redundancy and Backup Systems
<Mitigation>
Training and Familiarization
<Vulnerability>
+ Name: Inefficient communication systems
<Mitigation>
Medical Facilities and Equipment
<Mitigation>
Collaborate with Local Medical Services
<Vulnerability>
+ Name: Inadequate Medical Support
<MISSION>
+name_mission: Emergency Response
<External_Threat>
+ Name_threat: medical incidents
<Contremeasure>
Spectator Screening
<Contremeasure>
Medical Emergency Response Plan
<Contremeasure>
Crowd Medical Awareness and Education
<Contremeasure>
Medical Equipment and Supplies
<Contremeasure>
Medical Transport and Evacuation
<External_Threat>
+ Name_threat: Terrorist Attacks
<Contremeasure>
Surveillance and Monitoring
<Contremeasure>
Security Personnel
<Contremeasure>
Collaboration with Law Enforcement
<Internal_Threat>
+ Name_threat: Fire Hazards
<Contremeasure>
Fire Prevention Measures
<Contremeasure>
fire detection systems
<Contremeasure>
Collaboration with Fire Services
<Internal_Threat>
+ Name_threat: Structural Failures
<Contremeasure>
Regular Inspections and Maintenance
<Contremeasure>
Structural Monitoring Systems
<Contremeasure>
Crowd Management Protocols
<Contremeasure>
Professional Engineering Expertise
<Contremeasure>
Collaboration with Regulatory Authorities
<SOS>
Crowd management
<Mitigation>
Employ industry-standard encryption protocols, such as SSL/TLS.
<Mitigation>
use strong encryption algorithms and appropriate key management practices
<Mitigation>
Third-Party Services and Cloud Providers
<Vulnerability>
+ Name: Insufficient Data Encryption
<Mitigation>
Prepared Statements and Parameterized Queries
<Mitigation>
Database User Separation
<Mitigation>
Secure Development Lifecycle
<Mitigation>
Web Application Firewalls
<Vulnerability>
+ Name: SQL Injection
<Mitigation>
Input Sanitization and Validation
<Mitigation>
Output Encoding
<Mitigation>
Content Security Policy (CSP)
<Mitigation>
HTTP-only Cookies and Session Management
<Vulnerability>
+ Name: Cross-Site Scripting (XSS)
<Mitigation>
Secure Backup and Storage
<Mitigation>
Secure Database Configuration
<Vulnerability>
+ Name: Data Exposure and Insecure Data
<Mission>
+name_mission: Ticketing
<External_Threat>
+ Name_threat: Distributed Denial of Service (DDoS) Attacks
<Contremeasure>
Deploy robust network infrastructure capable of handling high volumes of traffic
<Contremeasure>
Employ rate limiting and traffic shaping mechanisms
<Contremeasure>
Collaborate with Internet Service Providers (ISPs) or Content Delivery Networks (CDNs)
<Contremeasure>
Utilize DDoS mitigation services or solutions that employ traffic analysis and filtering techniques
<External_Threat>
+ Name_threat: Phishing Attacks
<Contremeasure>
Security Awareness Training
<Contremeasure>
Web Filtering and Firewall Protection
<Contremeasure>
Multi-Layered Security Solutions
<Contremeasure>
Email Filters and Spam Detection
<Internal_Threat>
+ Name_threat: Unauthorized Access
<Contremeasure>
Separation of Duties
<Contremeasure>
Enforce strong authentication mechanisms
<Contremeasure>
Implement strict access controls
<Internal_Threat>
+ Name_threat: Insider attacks
<Contremeasure>
Insider Threat Detection: user behavior analytics (UBA) and anomaly detection systems
<Contremeasure>
Role-Based Access Control (RBAC)
<Contremeasure>
Employee Background Checks
<Contremeasure>
Confidentiality Agreements and Ethical Guidelines

Figure 7: Graphical notation for threat and vulnerability determination

7. The final user of the application:

To determine the threats and the vulnerabilities in a SoS, we have proposed an application allowing the security expert of the system to do so.

To describe what the security expert can do using our application, we present in Figure (8) the use case diagram of our application:

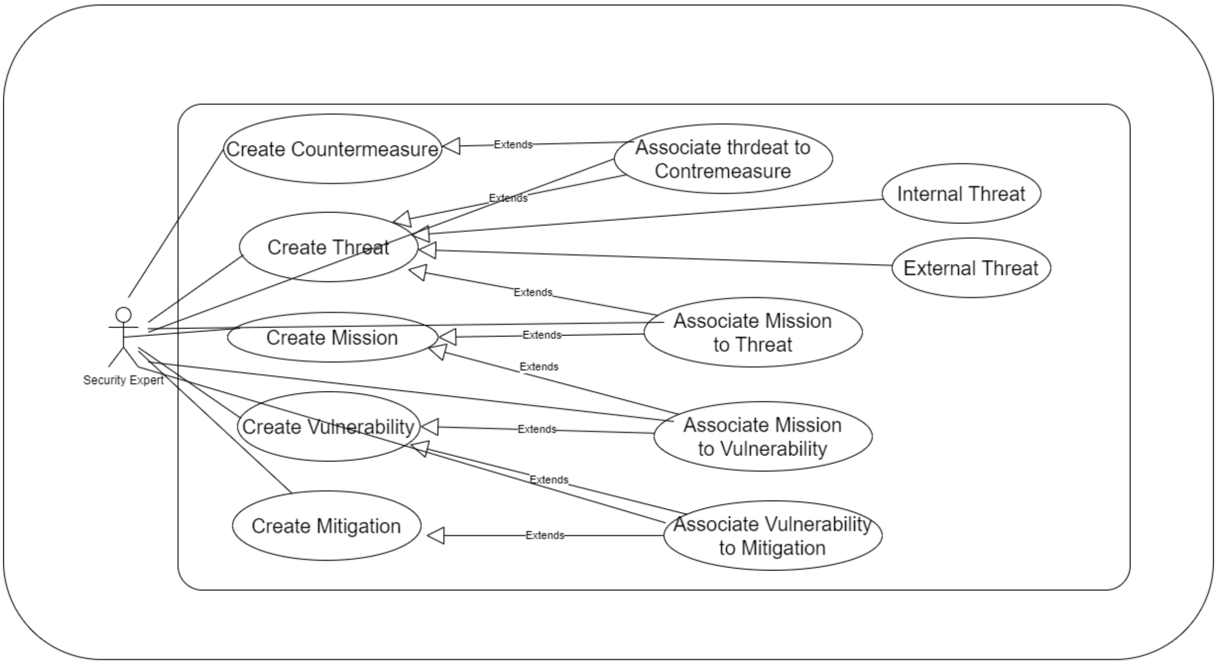


Figure 8: Use case of the application

8. Conclusion:

In conclusion, this chapter has examined the integration of NIST risk assessment steps within a mission-oriented process in Systems of Systems engineering, specifically in the context of crowd management. Through the case study presentation, the practical application of these steps within a crowd management scenario has been demonstrated, showcasing their effectiveness in assessing and mitigating risks. The detailed description of each step within the context of crowd management provides a comprehensive understanding of their implementation and their relevance to managing complex crowd dynamics. The proposed Meta-model offers a valuable framework for enhancing risk assessment capabilities specifically tailored to the unique challenges of crowd management. Additionally, the introduction of a concrete syntax utilizing graphical notation facilitates the identification of vulnerabilities and threats in crowd management scenarios. The final user of this application, the security expert, will benefit from the insights and tools provided to effectively assess risks, ensure public safety, and enhance the overall crowd management process in Systems of Systems engineering.

Chapter 4: Implementation

1. Introduction:

After proposing our meta-model, as well as the corresponding modeling, in this chapter we will present the implementation of the modeling. We begin by defining the work environment, as well as the development tools used. We conclude this chapter with the presentation of our application through a case study.

2. Work environment and the used tools:

In our application, we used the Java 11 platform as the development and execution environment, along with the Eclipse Modeling Tools package, which includes essential components such as the Eclipse Modeling Framework (EMF) and the Graphical Modeling Framework (GMF). We will now introduce these elements below:

2.1. Java:

Java 11, released in September 2018, brought numerous enhancements and features to the Java programming language. One of the most prominent additions was the introduction of a long-term support (LTS) release, providing stability and extended support for enterprises and developers [28].

2.2. The Eclipse Modeling Framework (EMF):

EMF is an open-source framework designed to simplify the process of building domain-specific modeling tools and applications. It provides a comprehensive set of tools and libraries that allow developers to define metamodels, generate models based on those metamodels, and perform various operations on the models. EMF simplifies the creation, manipulation, and persistence of models by providing features such as model validation, serialization, and code generation. With EMF, developers can quickly build modeling tools that enable the efficient creation, editing, and transformation of models, making it a popular choice for model-driven development in various domains. [29]

2.3. The Graphical Modeling Framework (GMF):

GMF is a powerful tool for creating domain-specific graphical editors within the Eclipse platform. It provides a comprehensive framework that allows developers to define graphical notations and generate customizable diagram editors for their specific domain models. [30]

The process of generating a GMF graphical editor involves six distinct steps [31]:

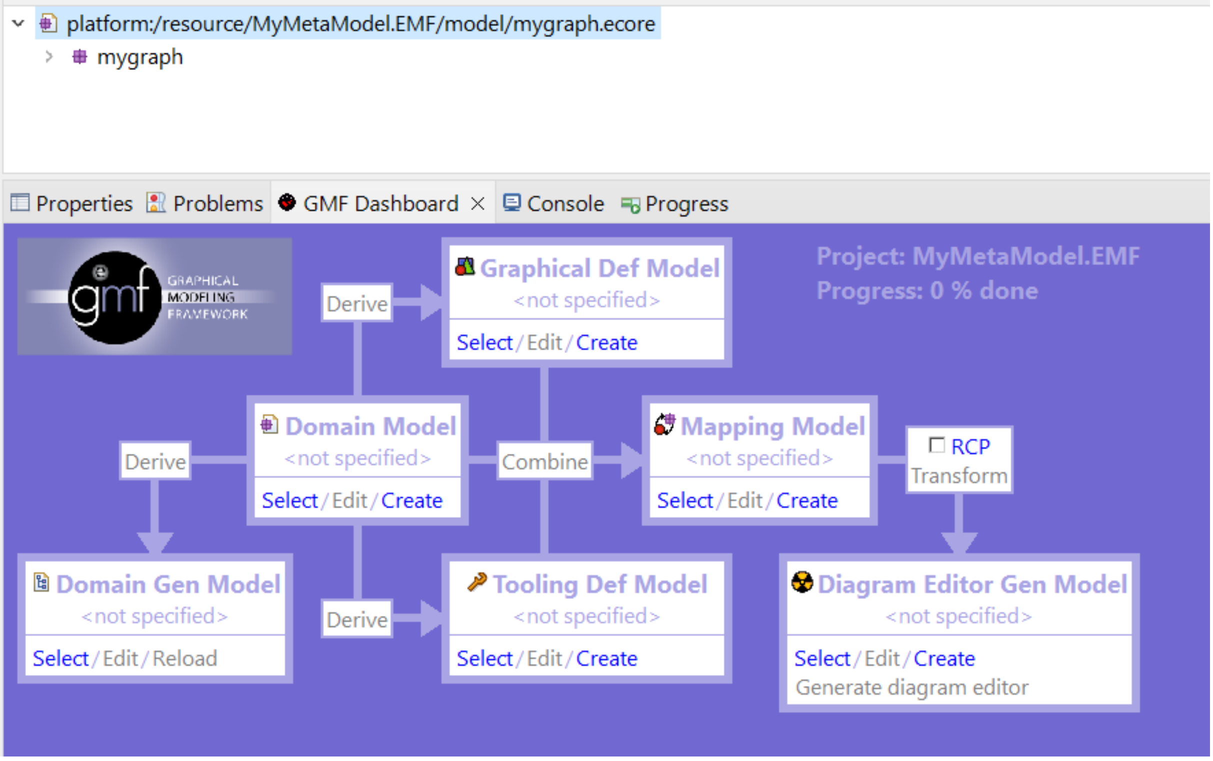
- Firstly, a domain model is selected, which serves as the metamodel for creating the graphical editor. Various metamodel options are available, such as Annotated Java code, Ecore model, class model, UML model, or XML Schema.

- Secondly, the Domain Gen Model (.genmodel) file is utilized to generate the code for the domain model using EMF (Eclipse Modeling Framework).
- Thirdly, the Graphical Def Model (.gmfgraph) file is employed to define the graphical elements associated with the chosen domain model.
- Next, the Tooling Def Model (.gmftool) file is used to specify the palette of tools available within the graphical editor.
- The Mapping Model (.gmfmap) file serves as the link between the domain model, the graphical model (.gmfgraph), and the tooling model (.gmftool).
- Finally, the Diagram Editor Gen Model (.gmfgen) file plays a crucial role in generating the GMF graphical editor alongside the EMF code produced by the .genmodel file.

3. Presentation of the application:

3.1. Create the domain model:

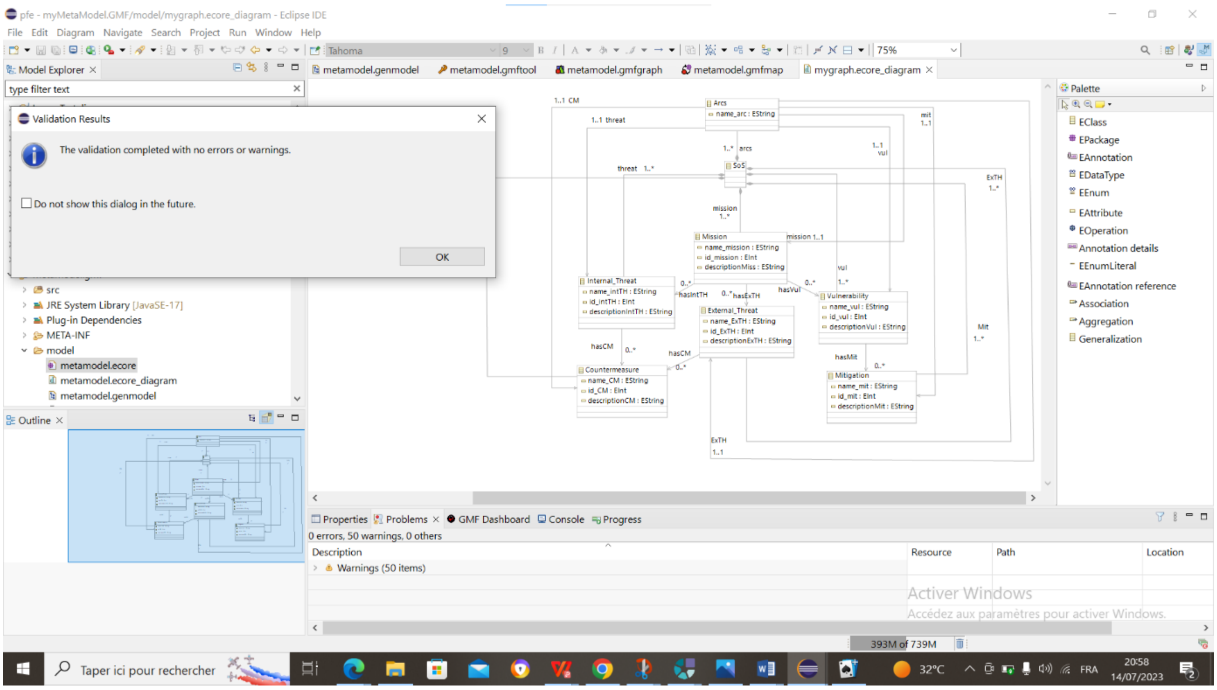
The following figure 9 represents the first step in creating the meta-model, which involves creating the GMF file (File > new > Graphical modeling framework > graphical editor project) in order to create a .ecore file inside it (new > eclipse modeling framework > ecore model).



platform:/resource/MyMetaModel.EMF/model/mygraph.ecore
mygraph

Figure 9 : GMF dashboard

In figure 10 we created our meta model with EMF and validated it :



Validation Results

The validation completed with no errors or warnings.

☐ Do not show this dialog in the future.

OK

0 errors, 50 warnings, 0 others

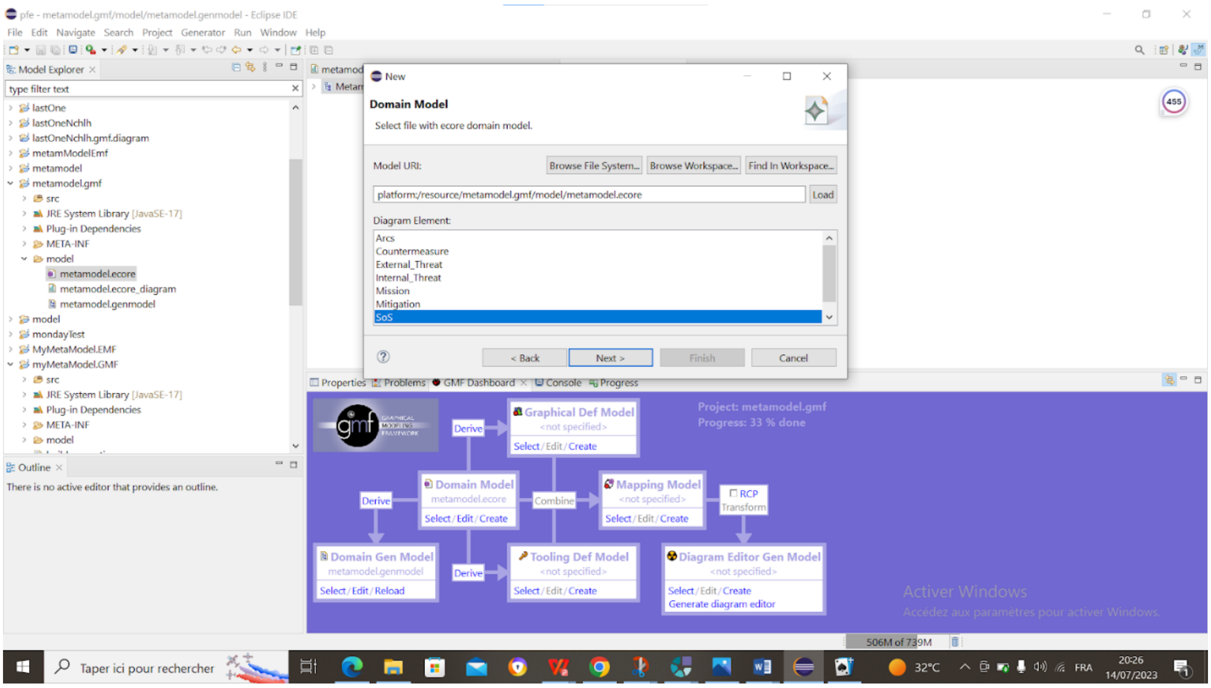
Warnings (50 items)

Figure 10: creating and validating the meta-model

After creating and validating our meta-model we obtain the domain gen model.

3.2. Generate diagram code :

Firstly we started by choosing the domain model elements to process and validating tooling palette in figure 11,12 :



New

Domain Model

Select file with ecore domain model.

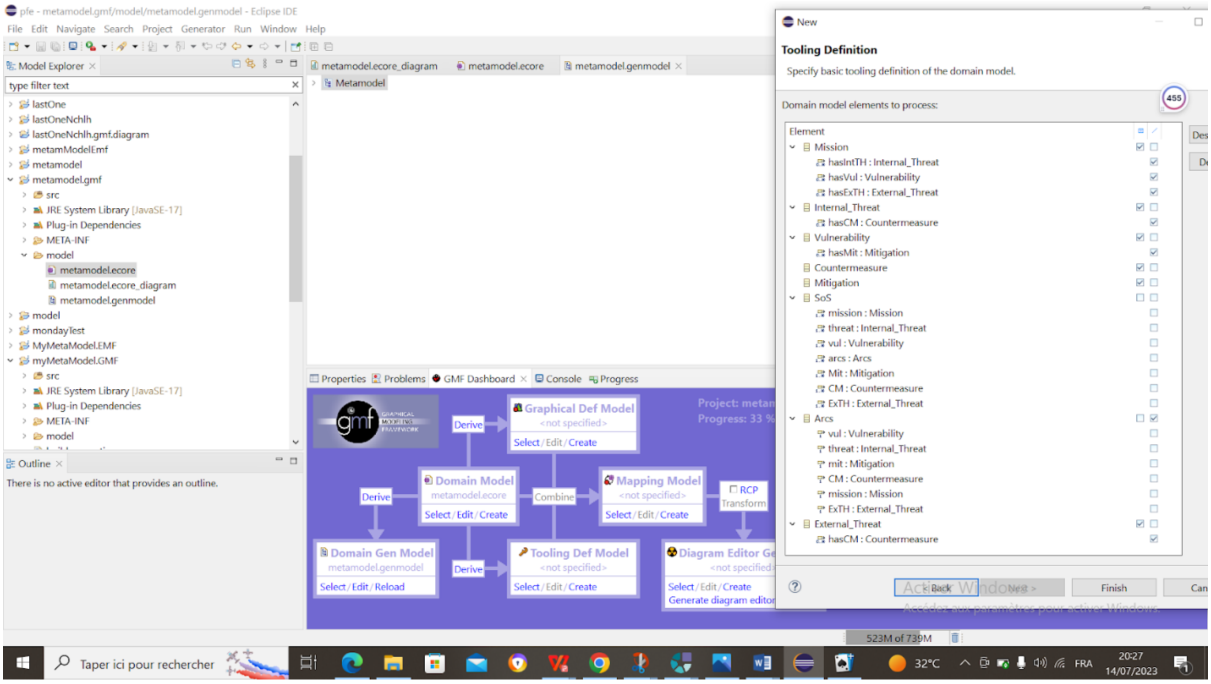
Model URI: platform:/resource/metamodel.gmf/model/metamodel.ecore

Diagram Element:
Arcs
Countermeasure
External_Threat
Internal_Threat
Mission
Mitigation
SoS

There is no active editor that provides an outline.

Project: metamodel.gmf
Progress: 33 % done

Figure 11: create the tooling palette


pfe - metamodel.gmf/model/metamodel.genmodel - Eclipse IDE
File Edit Navigate Search Project Generator Run Window Help
Model Explorer
Outline
There is no active editor that provides an outline.
Properties Problems GMF Dashboard Console Progress
Graphical Def Model
Domain Model
Mapping Model
Domain Gen Model
Tooling Def Model
Diagram Editor Gen
Derive
Combine
RCP Transform
Select / Edit / Create
Project: metamodel
Progress: 33 %
Taper ici pour rechercher
32°C
FRA
20:27
14/07/2023

New

Tooling Definition

Specify basic tooling definition of the domain model.

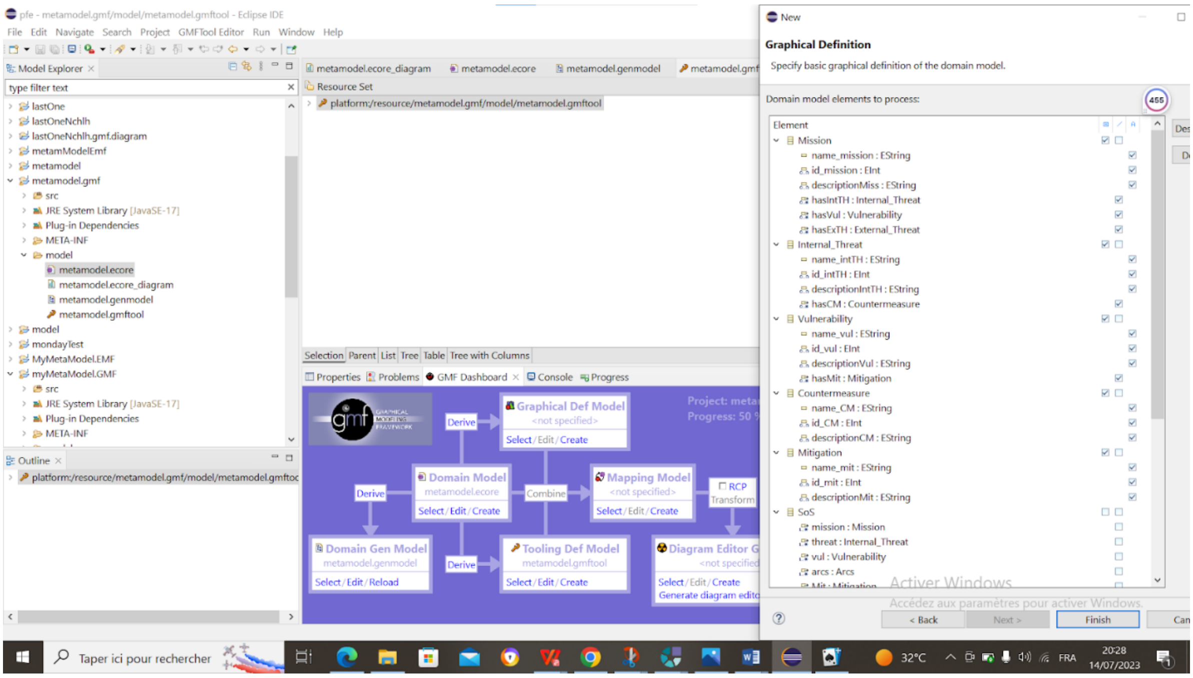
Domain model elements to process:

Element
- Mission ☑ □
  - hasIntTH : Internal_Threat ☑
  - hasVul : Vulnerability ☑
  - hasExTH : External_Threat ☑
- Internal_Threat ☑ □
  - hasCM : Countermeasure ☑
- Vulnerability ☑ □
  - hasMit : Mitigation ☑
- Countermeasure ☑ □
- Mitigation ☑ □
- SoS □ □
  - mission : Mission □
  - threat : Internal_Threat □
  - vul : Vulnerability □
  - arcs : Arcs □
  - Mit : Mitigation □
  - CM : Countermeasure □
  - ExTH : External_Threat □
- Arcs □ ☑
  - vul : Vulnerability □
  - threat : Internal_Threat □
  - mit : Mitigation □
  - CM : Countermeasure □
  - mission : Mission □
  - ExTH : External_Threat □
- External_Threat ☑ □
  - hasCM : Countermeasure ☑

< Back | Next > | Finish | Cancel

Figure 12: choosing the domain model elements to process

In figure 13,14 we created and validated the graphical definition model by choosing the graphical definition elements and selecting the root element of the domain model :


pfe - metamodel.gmf/model/metamodel.gmftool - Eclipse IDE
File Edit Navigate Search Project GMFTool Editor Run Window Help
Model Explorer
Resource Set
platform:/resource/metamodel.gmf/model/metamodel.gmftool
Selection Parent List Tree Table Tree with Columns
Properties Problems GMF Dashboard Console Progress
Graphical Def Model
Domain Model
Mapping Model
Domain Gen Model
Tooling Def Model
Diagram Editor Gen
Derive
Combine
RCP Transform
Project: metamodel
Progress: 50 %
Taper ici pour rechercher
32°C
FRA
20:28
14/07/2023

New

Graphical Definition

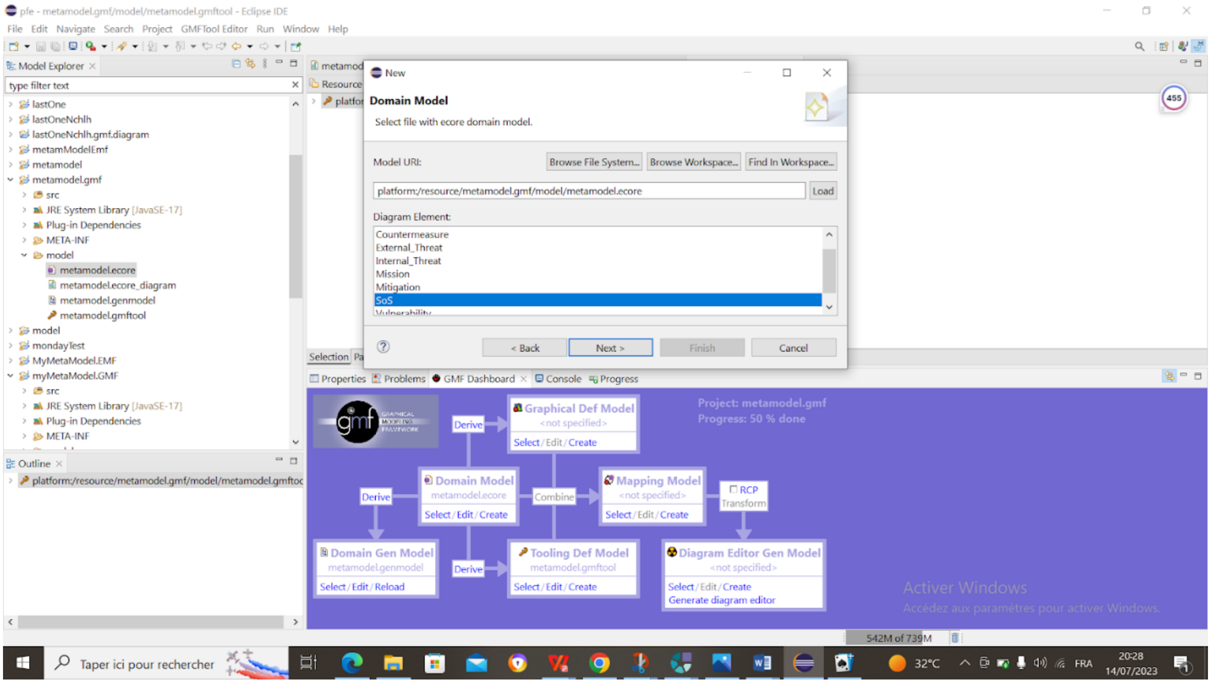
Specify basic graphical definition of the domain model.

Domain model elements to process:

Element
- Mission ☑ □
  - name_mission : EString ☑
  - id_mission : EInt ☑
  - descriptionMiss : EString ☑
  - hasIntTH : Internal_Threat ☑
  - hasVul : Vulnerability ☑
  - hasExTH : External_Threat ☑
- Internal_Threat ☑ □
  - name_intTH : EString ☑
  - id_intTH : EInt ☑
  - descriptionIntTH : EString ☑
  - hasCM : Countermeasure ☑
- Vulnerability ☑ □
  - name_vul : EString ☑
  - id_vul : EInt ☑
  - descriptionVul : EString ☑
  - hasMit : Mitigation ☑
- Countermeasure ☑ □
  - name_CM : EString ☑
  - id_CM : EInt ☑
  - descriptionCM : EString ☑
- Mitigation ☑ □
  - name_mit : EString ☑
  - id_mit : EInt ☑
  - descriptionMit : EString ☑
- SoS □ □
  - mission : Mission □
  - threat : Internal_Threat □
  - vul : Vulnerability □
  - arcs : Arcs □
  - Mit : Mitigation □

< Back | Next > | Finish | Cancel

Figure 13: choosing the graphical definition elements



New

Domain Model

Select file with ecore domain model.

Model URI: Browse File System... Browse Workspace... Find In Workspace...

platform:/resource/metamodel.gmf/model/metamodel.ecore Load

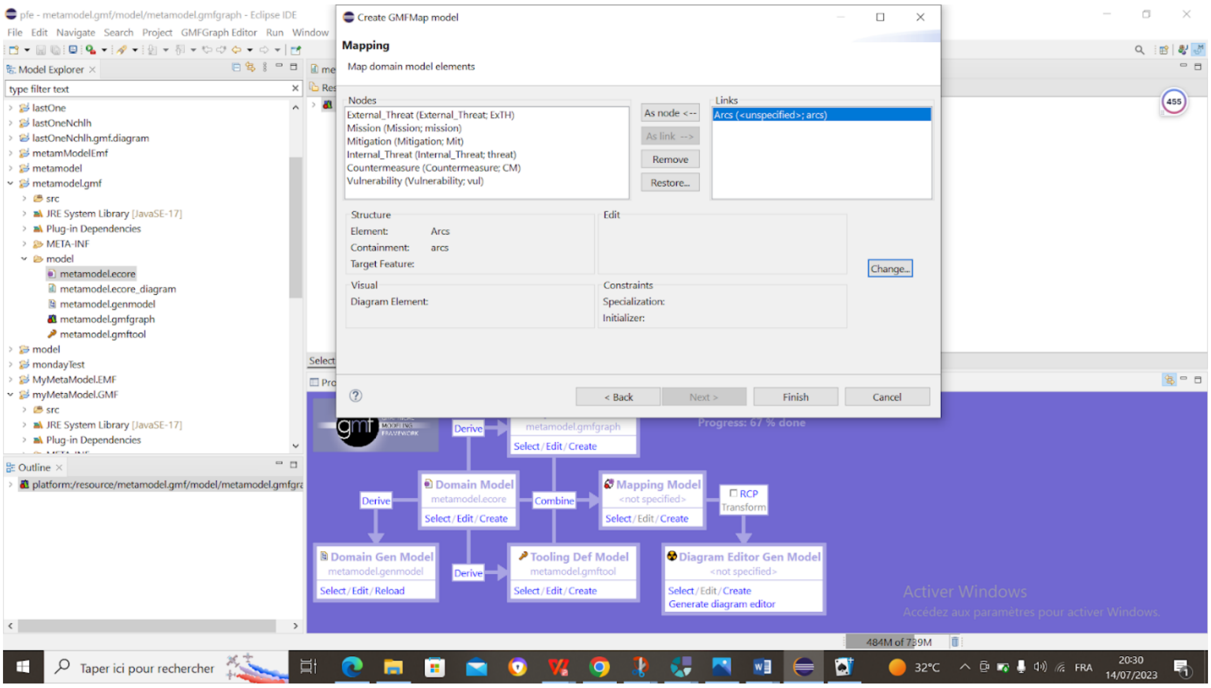
Diagram Element:

- Countermeasure
- External_Threat
- Internal_Threat
- Mission
- Mitigation
- SoS
- Vulnerability

< Back Next > Finish Cancel

Figure 14: selecting the root element of the domain model

In figure 15 we created and validated the mapping model :



Create GMFMap model

Mapping

Map domain model elements

Nodes

- External_Threat (External_Threat; ExTH)
- Mission (Mission; mission)
- Mitigation (Mitigation; Mit)
- Internal_Threat (Internal_Threat; threat)
- Countermeasure (Countermeasure; CM)
- Vulnerability (Vulnerability; vul)

As node <-- As link --> Remove Restore...

Links

- Arcs (<unspecified>; arcs)

Structure

Element: Arcs

Containment: arcs

Target Feature:

Edit

Change...

Visual

Diagram Element:

Constraints

Specialization:

Initializer:

< Back Next > Finish Cancel

Figure 15: creating the mapping model

After creating and validating the meta-model in Eclipse, we followed all the steps of GMF (previously mentioned in brief) and successfully completed 100% of them. As a result, we were able to generate the code and run the application as it shown in the figure 16 and 17:

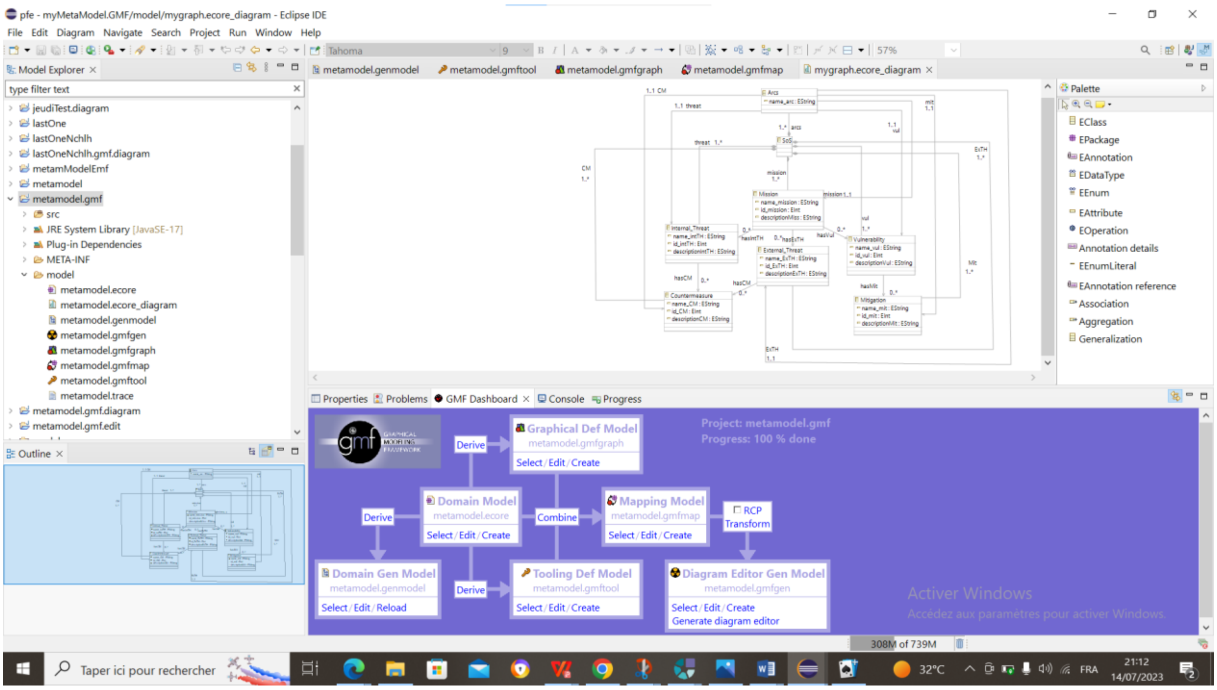


Figure 16: Progress 100% done

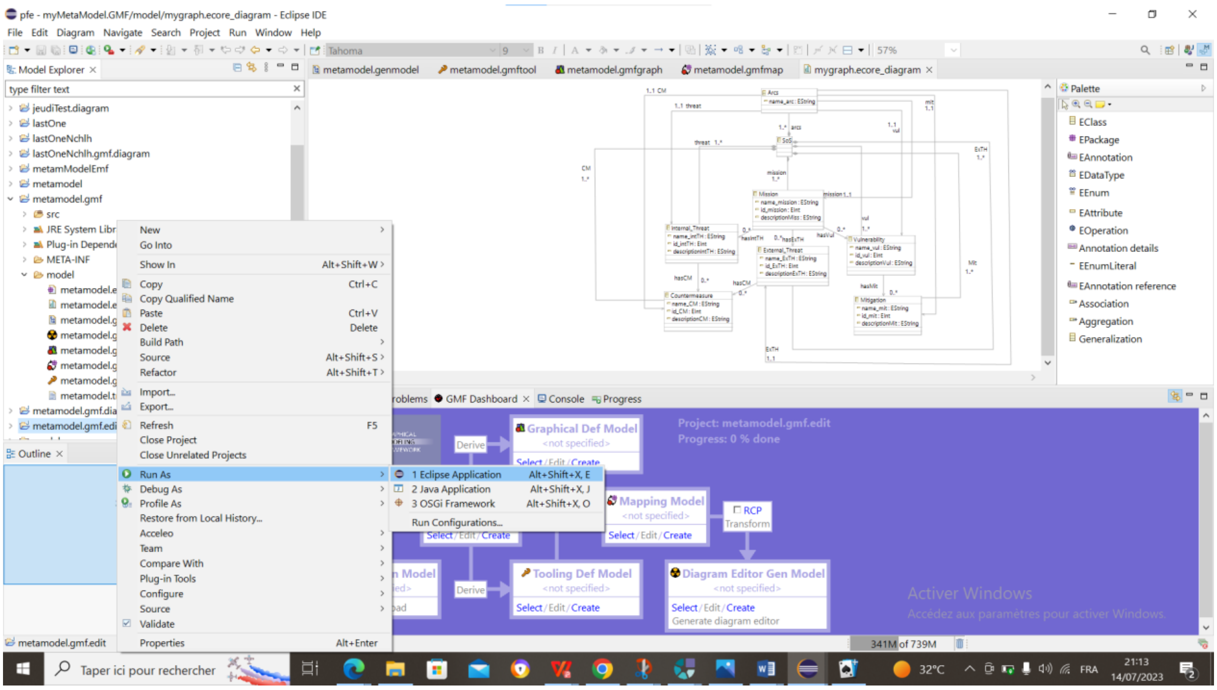
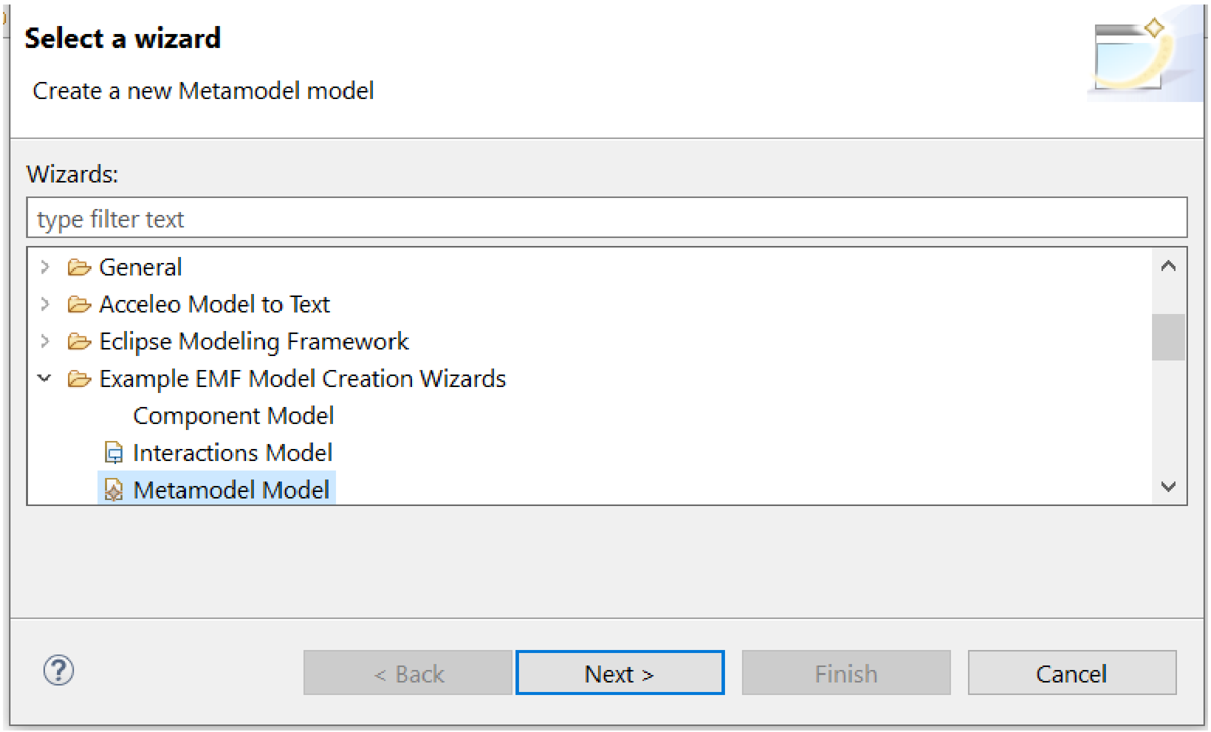

Run As
1 Eclipse Application Alt+Shift+X, E
2 Java Application Alt+Shift+X, J
3 OSGi Framework Alt+Shift+X, O
Run Configurations...
Project: metamodel.gmf.edit
Progress: 0 % done

Figure 17: The Application ready to run

4. Testing the application:

After opening the application, we click in File > new > Example EMF Mode Creation Wizards > Metamodel Model as it shown in figure 18:



Select a wizard

Create a new Metamodel model

Wizards:

type filter text

Figure 18: Creating the Metamodel model

When we click in SoS we are able to create new child and choose one of the meta-classes:

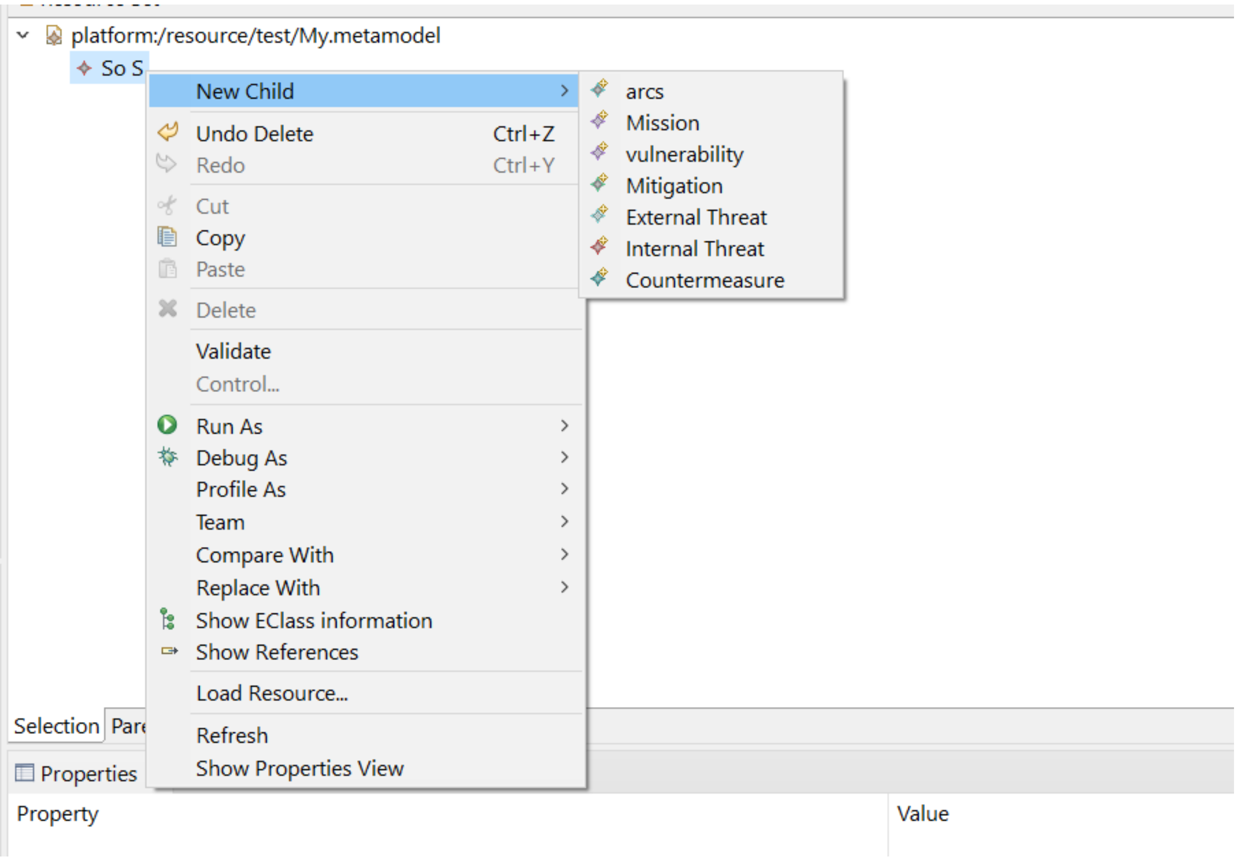
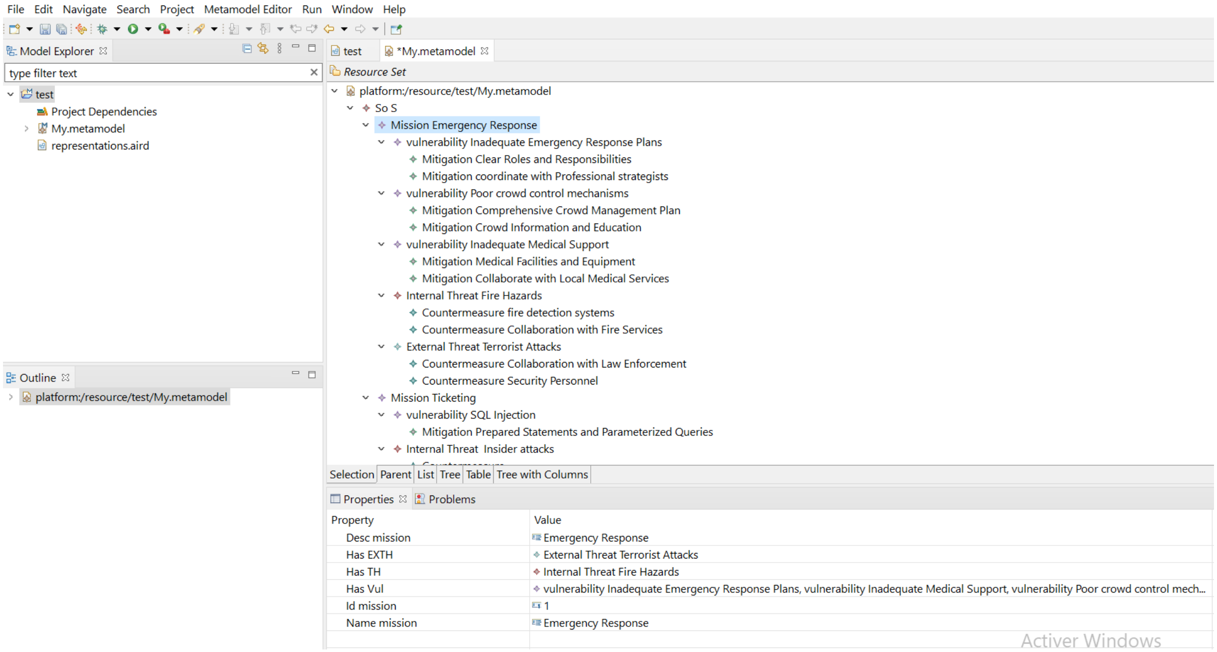


Figure 19: creating a new child of an SoS

To test the application, we created some meta-classes, in figure 20 we created an emergency response mission and a ticketing mission and attributed their vulnerabilities and threads also their mitigations and countermeasures:



```
platform:/resource/test/My.metamodel
  So S
    Mission Emergency Response
      vulnerability Inadequate Emergency Response Plans
        Mitigation Clear Roles and Responsibilities
        Mitigation coordinate with Professional strategists
      vulnerability Poor crowd control mechanisms
        Mitigation Comprehensive Crowd Management Plan
        Mitigation Crowd Information and Education
      vulnerability Inadequate Medical Support
        Mitigation Medical Facilities and Equipment
        Mitigation Collaborate with Local Medical Services
      Internal Threat Fire Hazards
        Countermeasure fire detection systems
        Countermeasure Collaboration with Fire Services
      External Threat Terrorist Attacks
        Countermeasure Collaboration with Law Enforcement
        Countermeasure Security Personnel
    Mission Ticketing
      vulnerability SQL Injection
        Mitigation Prepared Statements and Parameterized Queries
      Internal Threat  Insider attacks
```

Property	Value
Desc mission	Emergency Response
Has EXTH	External Threat Terrorist Attacks
Has TH	Internal Threat Fire Hazards
Has Vul	vulnerability Inadequate Emergency Response Plans, vulnerability Inadequate Medical Support, vulnerability Poor crowd control mech...
Id mission	1
Name mission	Emergency Response

Figure 20: A comlete Test of the application

5. Conclusion:

In this chapter we presented the implementation of our metamodel, we started by defining the work environment and the development tools. Secondly, we explained briefly the steps of the implementation and finally we presented our application with a small test within the context of our use case study.

General conclusion:

In conclusion, the thesis titled "Toward an Integrated Approach for Security Risk Assessment in Mission-Oriented System of Systems" has successfully achieved its objective of proposing an integrated approach for security risk assessment in mission-oriented System of Systems (SoS). By integrating the NIST Risk Assessment Steps into the Mission-Oriented Process of Systems of Systems Engineering, the thesis has contributed to the development of a comprehensive approach that addresses security risks in a structured and systematic manner.

The approach has provided a solid foundation for identifying threats and vulnerabilities within mission-oriented SoS. These steps, encompassing threat identification, vulnerability assessment, and risk mitigation, offer a structured approach to assess security risks and make informed decisions.

The use case study of football crowd management served as a practical application of the proposed framework. By implementing the steps in this specific scenario, the framework showcased its effectiveness in identifying and mitigating security risks. The example demonstrated the importance of considering the interdependencies and interactions among the various systems within the SoS, and how the proposed framework enables a targeted and effective risk assessment approach.

The creation of a meta-model and its implementation further strengthened the practicality and usability of the framework. The generated meta-model provides a visual representation of the framework, aiding decision-makers and stakeholders in understanding and implementing the proposed approach.

Looking forward, further validation and refinement of the future work of the framework:

Accomplishing the whole process of the proposed approach for risk assessment. This will ensure its applicability to diverse mission-oriented SoS scenarios and enhance its effectiveness in addressing evolving security challenges.

Moreover, continuous research and the integration of emerging technologies and threat intelligence should be incorporated into the framework. As the threat landscape evolves, updating the framework to adapt to new and emerging threats will be crucial for maintaining its relevance and effectiveness.

Striving for continuous improvement and development to enhance the application's functionality and user experience

In conclusion, the integrated approach for security risk assessment in mission-oriented System of Systems presented in this thesis offers a valuable contribution to the field. It provides decision-makers with a structured and systematic framework to identify and mitigate security risks, enhancing the overall security posture of mission-oriented SoS and ensuring the successful achievement of their objectives.

References:

[1] ISO/IEC/IEEE, “21839 systems and software engineering — system of systems (sos) considerations in life cycle stages of a system,” ISO/IEC/IEEE, Tech. Rep., 2019.

[2] Xia, B., Zhao, Q., Dou, Y., & Zhan, C. (2016, July). Robust system portfolio modeling and solving in complex system of systems construction. *2016 35th Chinese Control Conference (CCC).*

[4]: MARK W. MAIER . Architecting Principles for Systems-of-Systems. INCOSE International Symposium, 6(1):565–573, 1996.

[5]: MAIER MARK W. Architecting principles for systems-of-systems. Systems Engineering, 1(4):267–284, 1998.

[6] ISO/IEC/IEEE International Standard - Systems and software engineering – Taxonomy of systems of systems. ISO/IEC/IEEE 21841:2019(E), pages 1–20, 2019. 20, 21

[7] JO ANN LANE. What is a System of Systems and Why Should I Care? 2013. v, 1, 2, 21, 22, 23

[8] MAIER MARK W. Architecting principles for systems-of-systems. Systems Engineering, 1(4):267–284, 1998. 1, 21, 23

[9] J. S. DAHMANN AND K. J. BALDWIN. Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. In 2008 2nd Annual IEEE Systems Conference, pages 1–7. IEEE, 2008. 1, 21, 22, 23

[10] CHARLES KEATING, RALPH ROGERS, RESIT UNAL, DAVID DRYER, ANDRES SOUSA-POZA, ROBERT SAFFORD, WILLIAM PETERSON, AND GHAITH RABADI. System of Systems Engi[1]neering. Engineering Management Journal, 15(3):36–45, 2003. 25

[11] DEPARTMENT OF DEFENSE. Systems Engineering Guide for Systems of Systems, August 2008. v, ix, 1, 2, 20, 21, 22, 25, 27, 28, 30, 32, 33, 58, 76, 80, 123

[12] Cherfa, I., Belloir, N., Sadou, S., Fleurquin, R., &amp; Bennouar, D. (2019,November). Systems of systems: From mission definition to architecture description. Researchgate. Retrieved July 4, 2023,
from https://www.researchgate.net/publication/337172444_Systems_of_systems_From_mission_definition_to_architecture_description

[13] A. Waller and R. Craddock, "Managing runtime re-engineering of a System-of-Systems for cyber security," *2011 6th International Conference on System of Systems Engineering*, Albuquerque, NM, USA, 2011, pp. 13-18

[14] Vanea Chiprianov, Laurent Gallon, Manuel Munier, Philippe Aniorte, Vincent Lalanne. Challenges in Security Engineering of Systems-of-Systems. *3ème Conférence en IngénieriE du Logiciel (CIEL'2014)*, Jun 2014, Paris, France. pp.137-151

[15] D.J. Bodeau. System-of-systems security engineering. In Computer Security Applications Conference, 1994. Proceedings., 10th Annual, pages 228–235, Dec 1994.

[16] Michael Kennedy, David Llewellyn-Jones, Qi Shi, and Madjid Merabti. System-of-systems security: A survey. In The 11th Annual Conference on the Convergence of Telecommunications, Networking & Broadcasting (PGNet 2010), 2010.

[17] ISO/IEC. ISO/IEC 27005:2011: Information security risk management. Technical report, International Organization for Standardization (ISO), Geneva, Switzerland, 2011.

[18] E.I. Neaga and M.J. de C Henshaw. Modeling the linkage between systems interoperability and security engineering. In 5th Intl Conference on System of Systems Engineering, SoSE, June 2010.

[19] ISO 31000 Risk management (2009)

[20] ISO/IEC 27001Information security management systems (2005)

[21] NIST Special Publication 800-30 : Risk Management Guide for Information Technology Systems : Recommendations of the National Institute of Standards and Technology (2002)

[22] J. Dahmann, G. Rebovich, R. Lowry, J. Lane and K. Baldwin, “An Implementers view of Systems Engineering for Systems of Systems,” 2011 IEEE International Systems Conference Proceedings, April 2011, pp. 212-217.

[23] : E. H. Page, L. Litwin, M. T. McMahon, B. Wickham, M. Shadid, and E. Chang, “Goal-Directed Grid-Enabled Computing for Legacy Simulations,” 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing Conference Proceedings, May 2012, pp. 873-879

[24] Dr. J. Dahmann, G. Rebovich, G. Turner, “Security Engineering in a System of Systems Environment”, Prepared for 20th International Command and Control Research and Technology Symposium, The MITRE Corporation ,2015.

[25]J. Dahmann, G. Rebovich, G. Turner, “An Actionable Framework for System of Systems and Mission Area Security Engineering”, The MITRE Corporation,2014.

[26] D. Ki-Aries , S. Faily , H. Dogan , C. Williams, 2022 ,’Assessing system of systems information security risk with OASoSIS’,Computers &amp; Security , volume 117.

[27] https://www.workingwithcrowds.com/wp-content/uploads/2018/02/THE-CAUSES-AND-PREVENTION-OF-CROWD-DISASTERS-by-John-J.-Fruin-Ph.D.-P.E..pdf

[28] https://docs.oracle.com/en/java/javase/11/docs/api/index.html visited 20/05/2023

[29] Eclipse, Eclipse Modeling Framework (EMF) , https://www.eclipse.org/modeling/emf/ visited 20/05/2023

[30]. Eclipse Consortium. Eclipse Graphical Modeling Framework (GMF), https://wiki.eclipse.org/Graphical_Modeling_Framework/Tutorial/Part_1 visited 20/05/2023

[31] Eclipse Consortium. Eclipse Graphical Modeling Framework (GMF), https://wiki.eclipse.org/Graphical_Modeling_Framework/Tutorial/Part_2 visited 20/05/2023