People's Democratic Republic of Algeria.
Ministry of Higher Education and Scientific Research.
Computer Science department Saad Dahleb Blida University.

A framework for evaluating mission-oriented SoSs architectures

Submitted for the degree of Master SIR and IL.

Written by:	**Ouaissa Manel** **Sefta Dounia**
Supervised by:	**Imane Cherfa**
In front of the jury:	**Mme.Lahiani Nesrine: President** **Mme.Chikhi Imene: Examiner**

.

2021/2022

Thanks

First of all, we thank Allah for guiding us towards the path of light and knowledge, and for giving us the courage and the will to carry out this work. We would like to sincerely thank those who helped us develop this thesis, particularly our supervisor Dr Imane Cherfa for their invaluable help and guidance. We would also like to thank the members of the jury for the honor they do us by agreeing to judge our work. We would also like to thank all the teachers in the IT department who contributed to our training. May all those who have given us their support and who have helped us from near or far for the realization of this project find here the expression of our deep and sincere gratitude, in particular, our families and our friends.

Dedication

Every challenging work needs self-effort as well as the guidance of elders especially those who were very close to our hearts. My work, my effort I dedicate to

Father and Mother

Mom and dad, I can not begin to describe how blessed I am to have such amazing parents in my life. Seeing true love through you makes my heart happy, your encouragement and praying over days and nights make me able to get such success. Thanks for being my source of inspiration, and for providing me your moral, spiritual, emotional, and financial support. Thanks for teaching me to believe in myself, in god, and in my dreams, I have never done this without you.

My family, Sisters, Brother, Friends

Thanks for sharing your words of advice and encouragement to finish this study.

My partner "Manel Ouaissa"

Thanks for your effort.

My supervisor "Dr Imane Cherfa"

Thank you for guiding us in this work.

Sefta Dounia.

Dedication

I dedicate this modest work:
In memory of my paternal grandfather, may Allah welcome him to his vast paradise.
To my dear maternal grandmother "mamaninou" for her tenderness and affection, may this modest work be the expression of the wishes that you have never ceased to formulate in your prayers. May Allah preserve your health and long life.
To my dear parents, you have never hesitated to sacrifice yourselves for my success and happiness. I hope that one day I will be able to give you back at least the minimum because whatever I do I will never be able to give you back everything. May Allah protect and keep you.
To my adorable sisters "Meriem, Lina and Lydia".
To my dear brother "Samir", his wife "Zahira", and to my nephew "Mouhamed Wassim".
To my supervisor Dr.Imane Cherfa.
Ultimately, I warmly dedicate this work to my partner Sefta Dounia.

Ouaissa Manel.

ملخص

يشمل نظام الأنظمة (SoS) مجموعة من الأنظمة المستقلة والموزعة التي تعمل معًا من خلال التآزر فيما بينها لتحقيق مهمة مشتركة. تتحقق مهمة SoS من خلال تكامل CSs المستقلة ، مما يجعل التحديات المعمارية لإرضاء أداء المهمة قضايا مهمة.
هناك حاجة لفهم كيف يمكن للهياكل المحتملة المختلفة أن تحسن أداء المهمة ، ولتوفير الوسائل للاختيار بين الحلول المعمارية المختلفة.
الهدف من هذا العمل هو مواجهة هذه التحديات واقتراح إطار لتقييم بنية SoSs. أولاً ، اقترحنا نموذجًا مفاهيميًا يسلط الضوء على مقاييس التقييم المختلفة لمهمة SoS. ثانيًا ، نقوم بنمذجة جزء من الاستجابة لحالات الطوارئ SoS باستخدام لغة نمذجة النظام ،(SysML) ونوضح في التنفيذ كيفية تقييم هياكل SoS المختلفة.

الكلمات المفتاحية: نظام الأنظمة (SoSs) ، المهمة ، الهندسة المعمارية ، لغة نمذجة النظام .(SysML)

Abstract

System of systems (SoS) encompasses a group of independent and distributed systems which through synergism between them, work together towards a common mission. SoS's mission is achieved through the integration of independent CSs, which make architectural challenges for satisfying mission performance important issues. There is a need to understand how different potential architectures might improve mission performance, and to provide the means to choose between different architectural solutions.
The aim of this work is to address these challenges and to propose a framework for the evaluation of SoSs architecture. First, we proposed a conceptual model that highlights the different evaluation metrics for SoS mission. Second, we model a part of emergency response SoS using the System Modeling Language(SysML), and shows in the implementation how to evaluate the different SoSs architectures.

Keywords: System of Systems(SoSs), mission, architecture, System Modeling Language(SysML).

Résumé

Les systèmes de systèmes (SoS) englobent un groupe de systèmes indépendants et distribués qui, grâce à la synergie entre eux, travaillent ensemble à une mission commune. La mission du SoS est réalisée par l'intégration de CS indépendants, ce qui fait des défis architecturaux pour satisfaire les performances de la mission des problèmes importantes. Il est nécessaire de comprendre comment différentes architectures potentielles pourraient améliorer les performances des missions, et de fournir les moyens de choisir entre différentes solutions architecturales. L'objectif de ce travail est de relever ces défis et de proposer un cadre pour l'évaluation de l'architecture des SoSs. Tout d'abord, nous avons proposé un modèle conceptuel qui met en évidence les différentes mesures d'évaluation des missions SoS. Ensuite, nous modélisons une partie d'un SoS d'intervention d'urgence à l'aide du System Modeling Language (SysML), et nous montrons dans la mise en œuvre comment évaluer les différentes architectures de SoS.

Mots-clés: Système de systèmes (SoSs), mission, architecture, langage de modélisation des systèmes (SysML).

Contents

List of Figures

List of Tables

Notation and Acronym

SoS System of Systems

OMG Object Management Group

UML Unified Modeling Language

SysML System Modeling Language

CS Constituent System

MoP Measure of Performance

MoE Measure of Effectiveness

ABMS Agent Based Modeling and Simulation

DSL Domain Specific Language

SMC Statistical Model Chacking

DynBLTL Dynamic Bounded Linear Temporal Logic

LTL Linear Temporal Logic

ADL Architectural Description Language

DoD Departement of Defense

INCOSE International Council on Systems Engineering

BDD Block Definition Diagram

IBD Internal Block Diagram

UC Use Case diagram

ACT ACTivity diagram

SD Sequence Diagram

STM STate Machine diagram

PAR PARametric diagram

REQ REQuirements diagram

PKG PacKaGe diagram

ERSoS Emergency Response System of System

AI Artificial Intelligence

EIA Electronic Industries Association

GENERAL INTRODUCTION

Context

System of systems (SoS) encompasses a group of independent and distributed systems which through synergism between them, work together towards a common mission. SoSE (System of systems Engineering) is challenged by the operational and managerial independence of constituent systems, their geographic distribution, evolutionary development and emergent behavior. The SoSE literature addresses a wide variety of application domains such as commerce, transportation, healthcare and military, where independent and heterogeneous systems, often preexisting are implied to deliver a service. Emergency response is also an example of SoS. In the literature, various interesting approaches were proposed that tackle the design and development of SoSs. Model-Based System Engineering (MBSE) represents a promising path for the development and analysis of SoS.

Problem Statement

Although several propositions exist in MBSE to tackle with SoSs, it is still an emerging research field. Several aspects of SoSs as the operational and the managerial independence of constituent systems (CSs) create additional challenges. Indeed, the SoSs mission is achieved through the integration of independent CSs, which make architectural challenges for satisfying mission performance important issues. There is a need to understand how different potential architectures might improve mission performance and to provide the means to choose between different architectural solutions and evaluate them.

Objectives

The aim of this work is to address these challenges and propose a framework for evaluating SoSs architecture, which may also be used to compare different SoSs architectures. By framework, we mean a conceptual model that identifies the different evaluation metrics for mission in SoSs context as well as the steps and rules allowing the development of those metrics. According to INCOSE, the use of metrics allows having insight into the likelihood of achieving the operational objectives or capabilities, assessing the progress of the technical solution, and evaluating the technical risk as the solution evolves. Thus, it will be interesting to show how to use them in SoSs context.

Thesis structure

This manuscript will be presented in five main chapters as follow:

- **chapter 1 (System of systems definitions and basic concepts)** starts with an overview of what a system is, a brief history of SoS, SoS definitions, SoS characteristics and categories and ends with the system of systems modeling and the SoSE definition and its different issues.
- **chapter 2 (State of the art on evaluation of system of systems)** discusses state of the art in the evaluation of an SoS architectures and lists the works that are interested in the evaluation of an SoS architectures. Finally it ends with a discussion of these works.
- **chapter 3 (Contribution: system of systems mission evaluation framework)** we presents the conceptual model of the SoS architecture evaluation. We've described the structure and the behavior of our SoS using the SysML four pillars diagrams (block definition diagram

(BDD), requirement diagram (req), activity diagram (act), and the parametric diagram (par)). The case study that we used is Emergency Response System of Systems (ERSoS).

- Finally, in **chapter 4 (Implementation)**, we presented our implementation by creating a web application to evaluate ERSoS architectures and choosing the architecture that satisfies the main mission of the SoS.

Chapter 1

1 SYSTEM OF SYSTEMS DEFINITIONS AND BASIC CONCEPTS

1.1 Introduction

In this chapter, we present an overview of Systems of Systems (SoS), It begins with its historic appearance and the different definitions , characteristics,and its major categories. The chapter ends by addressing the SoS modeling and system of systems engineering (SoSE) definition and its different issues.

1.2 Definition of a system

There are many definitions of the system, we will present the most general definition of them.

1.2.1 Meriam-Webster's

Meriam-Webster's Collegiate Dictionary,defines system as "a regulary interacting or interdependent group of items forming a unified whole"[7].

1.2.2 MIL-STD-499

One of the early Military standards on the subject .MIL-STD-499 , defines system as "a composite of equipement, skills, and techniques capable of performing or/and supporting an operational role. A complete system includes all equipement, related facilities, material, software, services, and personnel required for its operation and support to the degree that it can be considered a self-suffficient unit in this intended envirement"[7].

1.2.3 EIA/IS-632

EIA/IS-632, defines system as "an integrated composite of people, product, and processes that provide a capability to satisfy a stated need or objectives"[7].

1.2.4 INCOSE

INCOSE, defines system as "A system is an arrangement of parts or elements that together exhibit behaviour or meaning that the individual constituents do not"[8].

1.3 Definition of SoS

There was no widely accepted SoS definition, several researchers gave various definitions, and others preferred to distinguish SoSs by listing their characteristics[5].

1.3.1 Eisner et al.

Defined SoS as: “A set of several independently acquired systems, each under a nominal systems engineering process; these systems are interdependent and form in their combined operation a multifunctional solution to an overall coherent mission.
The optimization of each system does not guarantee the optimization of the overall system of systems”[9] .

1.3.2 Shenhar

Describe SoS as: “A large widespread collection or network of systems functioning together to achieve a common purpose ”[9].

1.3.3 Jamshidi

Describe SoS as: “SoS are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal. The goal, may be cost, performance, robustness, etc”[5].

1.3.4 INCOSE

A SoS is a collection of independent systems, integrated into a larger system that delivers unique capabilities. The independent constituent systems collaborate to produce global behaviour that they cannot produce alone[10].

1.4 Differences between a system and an SoS

According to Liu[6], both systems engineering and system of systems engineering (SoSE) employ current techniques and knowledge, while also considering the characteristics of the system and SoS in designing and managing their respective types of structure and architecture. The system is defined as a totality composed of multiple interconnecting and interacting agents and components with the objective of achieving certain functions. Since the SoS is defined as a dynamic, parallel, distributed, and non-predetermined-scaled metasystem composed of many autonomous complex CSs, the function of an SoS is to achieve the objectives of the metasystem, as well as its constituent system, through the interoperation of its individual systems. We differentiate the system from the SoS based on the following aspects: autonomy, belongingness, connectivity, complexity, diversity, human involvement, synergism and interoperability, resource allocation, structure, policy intervention,uncetainty, and emergence. Table 1 summarizes those differences:

Table 1: Diffrences between a System and an SoS[6]

Features	System	SoS
Autonomy	Autonomy of parts is overridden by system.	The function of the SoS can only be achieved by granting its constituent systems full autonomy.
Belongingness	The belongingness of system elements is not determined by themselves but by their parents, the designated function, and the specific design of the system.	The belongingness of constituent systems to the SoS is determined by whether or not they can achieve their own objectives and purpose of the SoS in a cost-effective way.
Connectivity	Extensive connections are established among system components, while the connections among major subsystems are minimized.	Based on the practical requirements, the constituent systems of the SoS can dynamically bridge all possible connections among them.
Complexity	Complexities come from intra-system dynamics and limited number of combination of system parts.	Intra-system dynamics, inter-system dynamics, and astronomically numerous connections among constituent systems render the SoS extremely complex.
Diversity	The diversity of system elements is enveloped by different functional modules with the objective of reducing the complexity cast to the next level of the system hiararchies.	The autonomy, belongingness, numerous connections, and dynamic/combinatorial complexities augment the diversityof the SoS.
Human involvement	The necessity and time of human involvement in a system is predetermined and planned.	During the operation and evolution of the SoS, human involvement is determined by the inter-systems dynamics.
Interoperability and Synergism	The function of elements and components of a system is predefined. The parts of the system cannot employ other elements and systems data to independently make decisions and change system functions accordingly.	While the constituent systems of the SoS possess autonomy, by sharing data via the numerous connections, it is possible to have interoperation among these systems by adjusting their strategies and tactics as needed in order to realize their indivdual objectives, as well as the SoS objective.
Resource allocation and utilization	The allocation of resources is based on the function of the system. Itis possible to make a relatively accurate budget for the resources needed for designing and deploying a system.	The constituent systems have to adjust their projection for the resourcecs as needed. The resource utilization in the SoS exercises complex inter-change, inter-conversion, and inter-operation in order to achieve the purposes of the SoS and its constituent systems on a cost/benefit basis.

Structure or architecture	System structure at the most fundamental level is fixed , which determines how the system behavior might change over time.	Constituent systems and the numerous connections of the SoS form a dynamic system architecture that changes itself correspondingly in order to acheive the purpose of the SoS and its constituent systems in a cost-effective manner.
Policy intervention	Policy intervention generally changes the system structure, and consequently the system behavior.	Policy interventions generally occur at one or multiple constituent systems, not at the SoS level, which perceptibly affect the implementation course of other policies in different systems in a way that might render other policies invalid.
Uncertainty	Uncertainties may stem from nonlinearity, human decisions, unnoticed interventions, and dynamics of system.	Uncertainties arise from tow sides: one from the dynamics of indivdual constituent systems and the other from the dynamics of the context (architecture) and numerous interactions through different connections among different systems.
Emergency	The design principle of a system is to provide an allownace for handling predictable events by considering and testing all possible scenarios.	The SoS is a dynamic archiecture that generates emergence-handling environments with the development of an emergent event, in order to prevent, mitgate, and eliminate unfavorable behaviors.

1.5 Brief history of SoS

The initial mention of the SoS can be traced to Boulding, Jackson and Keys, Ackoff, and Jacob[9]. Boulding imagined SoS as a “gestalt” in theoretical construction creating a “spectrum of theories” greater than the sum of its parts. Jackson and Keys suggested using the “SoS methodologies” as interrelationship between different systems-based problem-solving methodologies in the field of operation research. Ackoff considered SoS as a “unified or integrated set” of systems concepts. Jacob stated that a SoS is “every object that biology studies.” It was not until 1989, with the Strategic Defense Initiative, that we find the first use of the term “system-of-systems” to describe an engineered technology system[9]. Figure 1 shows the graphical time-line of the typical contributors of SoS from 1990 to 2014.

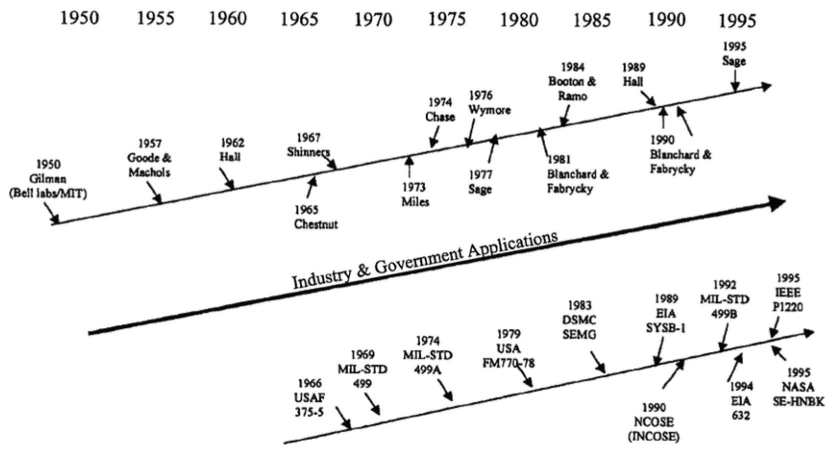

Figure 1: Increasing SE knowledge and practice[1]

1.6 System of systems Characteristics

Many characteristics have been proposed in the literature to differentiate system of systems from monolithic systems.

1.6.1 Maier's characteristics

Maier identified five major characteristics to distinguish between systems of systems and complex monolithic systems, often known by the acronym "OMGEE"[11]:

- **Operational Independence:** The constituent systems have purpose, even if detached from the SoS. The purposes often conflict with each other and also conflict with the purposes of the SoS, resulting in conflicts among the constituent system stakeholders.
- **Managerial Independence:** The constituent systems are developed and managed for their own purposes. Each system has an independent owner and independent stakeholders, who may or may not overlap with the SoS stakeholders. This independence further exacerbates the conflicts.
- **Geographic Distribution:** Spreading the constituent systems across a geographic extent forces the elements to exchange information in a remote way, resulting in difficult technical communications issues.
- **Evolutionary Development:** Functions and purposes are added, removed, and modified within the system in an ongoing way. As the constituent systems change, there are constant revisions and difficult integration issues.
- **Emergent Behavior:** The SoS performs functions that are not achievable by the independent constituent systems; stakeholders want assurance of this emergent behavior even in the face of the challenges created by the following characteristics.

1.6.2 Boardman and Sauser characteristics

Boardman and Sauser also proposed distinguishing characteristics, that separate monolithic systems from a SoS. They identify five characteristics with the acronym "ABCDE" [5, 12]:

- **Autonomy:** Each system is independent and it's able to complete his own goals without any entity's control.
- **Belonging:** Belonging does mean partness for the autonomous system. Each system must form new relationships, with other autonomous systems, to render service, and to collaborate with other systems to achieve a common SoS goal .
- **Connectivity:** The constituent systems must be connected and linked with a powerful and dynamic network, with other systems towards the fulfillment of the SoS goals .
- **Diversity:** The constituent systems are diverse and different from one another. This diversity will make SoS open for evolution and adaptation.
- **Emerging:** mean that an emergent capability results from the interactions between the constituent systems and can be attributed to the overall SoS. This emerging designed by virtue of the other factors: preservation of constituent systems autonomy, choosing to belong, enriched connectivity, and commitment to diversity of SoS manifestations and behavior.

1.6.3 Abbott's characteristics

Abbott describe how SoS are qualitatively and structurally different from traditional system. He claimed that a SoS properly understood as an environment along with the systems operating within it[13] .These characteristics are :

- **Open at the top :** Signifies that a SoS is not defined in terms of some fixed top-level application. The system of system enables the continual introduction of new applications.
- **Open at the bottom :** Signifies that there is no fixed bottom level for a SoS. The lowest level of SoS may be modified out from under it at any time .
- **Continually evolving, but slowly:** Signifies that a SoS is never finished. It evolves continually as the environment within which it operates changes. SoS evolve in at least three ways.
  - Technology changes.
  - Usage changes: New features are added, and existing features are modified.
  - Standards and interfaces change .

1.7 System of systems Categorization

The U.S. Department of Defense makes use of four categories of SoS[14] ,these types illustrated in Figure 2 :

- **Directed:** The SoS is built to fulfill specific purposes. Constituent systems have the ability to operate independently, but are managed to satisfy a concrete purpose.
- **Collaborative:** The constituent systems are not compelled to follow a central management, but voluntarily participate in a collaboration to fulfil the goal.
- **Acknowledged:** The SoS recognizes a common purpose and goal, while the constituent systems retain independent control and objectives. Evolution of the common purpose is based on collaboration between the SoS and the constituent systems.

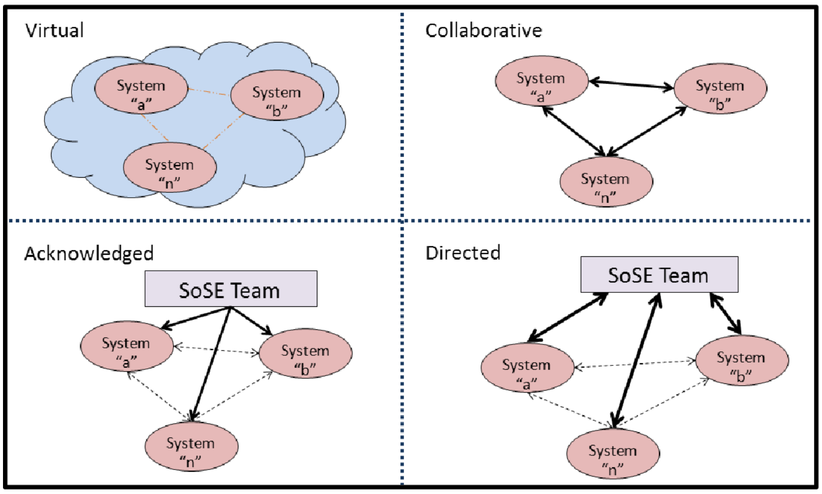

Figure 2: System of systems categories.

- **Virtual:**The SoS is without either managerial control or a common purpose. This makes the behaviour and the fulfilled goals highly emergent, but also entails that the exact means and structures producing the system functionality are difficult to discern and distinguish.

A categorization is required in order to guide the selection of architecting principles. The categories are based on the degree of managerial control because this determines how adaptable and cooperative each constituent system will be with respect to requirements, interfaces, data formats, and technologies. In turn, this influences the challenges faced when constructing the SoS[14].

1.8 System of systems modeling

Models can be expressed in many forms ranging from graphical sketches or text to mathematical formalisms[14].The main purpose in creating a model is to replicate and capture a specific characteristic of our study targets, in order to observe specific behaviours and easily manage complex systems[15]. The area that capture systems behaviour is known as Model Based Systems Engineering (MBSE).

1.9 System of systems engineering definition

System of Systems Engineering (SoSE) is a field that has emerged from the need to deal with a specific type of problem labelled System of Systems (SoS). This need evolved from the recognition that many capabilities and desired outcomes are developed through the integration of existing or legacy systems with the potential integration of new components or systems that provide desired capability or assist in their integration[16].
Keating et al. (2003a,b) state that SoSE is the "transformation of higher order metasystems that must function as an integrated complex system to produce desirable results". The Defense Acquisitions Guidebook indicates the purpose of SoSE is to "satisfy capabilities that can only be met with a mix of multiple, autonomous, and interacting systems". Similarly, a USAF SAB (2005) report stipulates that SoSE is intended to "integrate the capabilities of a mix of existing and new systems into a system-of-systems capability"[16].
The SoSE is the "process of planning, analyzing, organizing, and integrating the capabilities of a

mix of existing and new systems into a system-of-systems capability that is greater than the sum of the capabilities of the constituent parts"[17].

1.9.1 System of systems engineering issues

According to[5], SoSE's principal research areas could be classified into the following issues:

- **Modeling and architecting:** the development of models in which there is the use of existing systems as components of the SoS, and optimize the architecture while taking into account the SoS dimensions. Furthermore, the use of Domain-Specific Languages (DSL) to analyze SoS mission and capability objectives, and to set the concepts of operational development.
- **Simulation:** the proposition of simulation tools to analyze and understand the complexity of SoS behavior.
- **Testing:** the implementation of testing techniques in case of: complex and large SoS, the use of different standards, multi-stakeholder situation, dynamic evolution of SoS configurations, etc.
- **Verification:** the development of verification tools to support simulation and testing, for analyzing different properties.

1.10 Conclusion

We started this chapter by giving an overview of the system, the system of systems, a brief history of an SoS, SoS characteristics and its categorisation, and finally, the chapter ends with the SoS modeling and the definition of SoS engineering and its different issues (modeling and architecting, simulation, testing and verification).

Chapiter 2

2 STATE OF THE ART ON EVALUATION OF SYSTEM OF SYSTEMS

2.1 Introduction

One of the most important phases in the SoS life cycle is the evaluation of the SoS's architectures. The evaluation of several architectures permits us to compare those architectures and choose one that satisfies the SoS mission. In this chapter, we will see works that are interested in the evaluation of SoS architectures and discuss them.

2.2 Evaluation of SoS

Evaluation of software architectures is an important activity to the quality of software systems, as it verifies conformance and completeness of such architecture regarding requirements and goals. In another perspective, System-of-Systems (SoS) have emerged as a new class of software systems, which aggregates independent and heterogeneous constituent systems for performing new, emergent capabilities. Likewise, evaluation of SoS software architecture is also important for ensuring that important quality attributes are met in the SoS[18].

2.3 SoS mission

Several definitions of the term "mission" were proposed in the literature. They are listed in CHERFA[5]:

- **DoD definition:** the task, together with the purpose, that clearly indicates the action to be taken and the reason therefore[19].
- **Charles S. Wasson definition:** a pre-planned exercise that integrates a series of sequential or concurrent operations or tasks with an expectation of achieving outcome-based success criteria with quantifiable objectives[20].
- **R. DEIOTTE and R. K. GARRETT definition:** given a set or type of stimuli, the mission is the collection of tasks, goals and objectives that have to be achieved to successfully address the stimuli. The mission includes all of the physical assets necessary to meet the goals as well as all of the techniques and procedures necessary to effectively employ them[21].

A mission is a pre-planned exercise that integrates a series of sequential or concurrent operations or tasks with an expectation of achieving outcome-based success criteria with quantifiable objectives[22].

2.4 What is a Measure?

The measure is a dimension, quantity, or capacity (degree), ascertained by comparison to an accepted standard (e.g length in inches, weight in pounds, temperature in degrees Celsius)[23]. The International Council On Systems Engineering (INCOSE) describes technical measurement as "Technical measurement is the set of measurement activities used to provide the supplier and/or acquirer insight into progress in the definition and development of the technical solution and the associated risks and issues."[2]. Measurement helps the project manager to (1)Monitor the progress and performance of activities, (2)communicate effectively throughout the project organization, (3)identify and correct problems early, (4)make key tradeoffs, (5)track specific project objectives and (6)defend and justify decisions. The objective of measurement is to obtain insight into issues that impact project cost, schedule, and technical (performance, functionality, and quality) objectives in order to enable the project decision-makers to make informed decisions[2]. According to INCOSE, the commonly used technical measures are Measures of Effectiveness(MoEs), Measures of Performance(MoPs), and Technical Performance Measures(TPMs).

2.5 What are Metrics?

Metrics are measures of quantitative assessment commonly used for assessing, comparing, and tracking performance or production[24]. A Metric is a set of measures or methods that ascertain the progress a system is making toward achieving its goal. The SoS metrics are collected and analyzed as part of analyses to assess whether the SoS is making progress towards objectives[23].

2.6 INCOSE Metrics

The following are short definitions of the measures proposed by the INCOSE:

2.6.1 Measures of Effectiveness MoEs:

MoEs are operational measures of success closely related to the achievement of the mission objective being evaluated, in the intended operational environment under a specified set of conditions. They are stated from the user's viewpoint and represent the most important criteria against which the quality of a solution is assessed[25]. They represent stakeholder expectations and are used to validate that the system meets the user's intended needs. According to Sproles[3] MoE is "standards against which the capability of a solution to meet the needs of a problem may be judged. They are independent of any solution and specify neither performance nor criteria". So MoE measures the fitness of a system to fulfill the needs of its customers.

2.6.2 Measures of Performance MoPs:

MoPs are the measures that characterize physical or functional attributes relating to the system operation, they measure attributes considered important to ensure that the system has the capability to achieve operational objectives. MoPs are used to assess whether the system meets design or performance requirements that are necessary to satisfy the MoEs[2]. The reference [25] defines MoPs as "MoPs are measures that characterize physical or functional attributes relating to system operation, measured or estimated under specified testing and/or operational environment conditions. They are stated from the developer's viewpoint and look at how well the delivered system performs or is expected to perform, against system requirements."

2.6.3 Technical Performance Measures TPMs:

TPMs measure the attributes of a system element to determine how well that element is satisfying or expected to satisfy a technical requirement, they are derived from MoPs and they are used to confirm progress and identify deficiencies that might jeopardize meeting a system requirement[25].

2.6.4 Key Performance Parameters KPPs:

KPPs are a critical subset of the performance parameters representing those capabilities and characteristics that are so significant that failure to meet the threshold value of performance can be cause for the concept or system selected to be reevaluated or the project to be reassessed or terminated, they are the minimum number of performance parameters needed to characterize the major drivers of operational performance, supportability and interoperability[2].

2.6.5 Relationship of MoEs, MoPs, TPMs, and KPPs

D. Kaslow, B.Ayres, P.Cahill et al.[25] discussed the relationships between technical measures as follows: MoEs reflect the stakeholder's intention. They indicate an attribute a system must possess in order to meet an operational need. MoPs are derived from MoEs. MoPs are concerned with the actual performance of a system solution. They are used to assess whether the system meets requirements that are necessary to satisfy the MoEs. TPMs are then derived from MoPs and represent attributes of elements of the system architecture. TPMs are used to determine progress towards meeting a technical requirement and they provide a lower level view of specific aspects of the system solution. MoEs represent stakeholder expectations and are used to validate that the system meets the users' intended needs. MoPs represent the key performance characteristics the system should have in order to satisfy the MoEs and are used to verify that the system meets the stated requirements.
KPPs are generally derived from MoEs and are a primary influence on the selection of the MoPs[2].

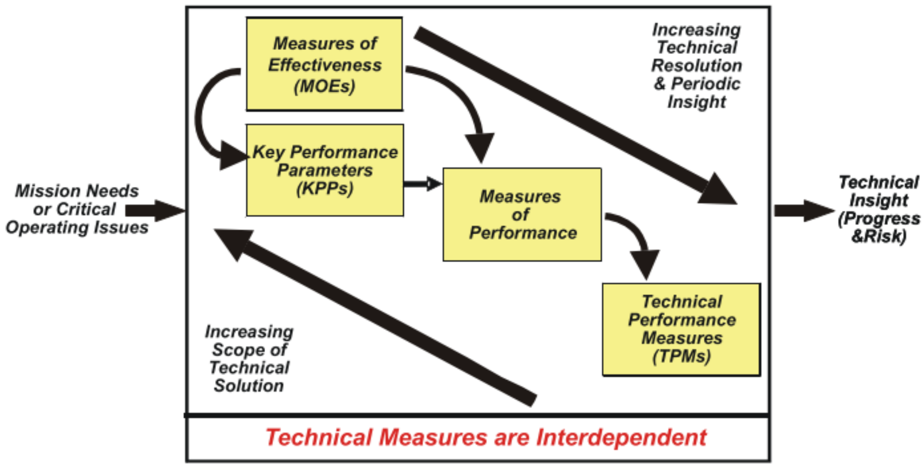

Figure 3: Relationship of the Technical Measures[2]

2.7 Work of Silva et al.: On the verification of mission-related properties in software-intensive systems-of-systems architectural design

Silva et al. propose a method for formally verifying mission-related properties in architectural models. This proposal relies on the formalism adopted in mKAOS, which uses DynBLTL language to describe properties, missions and emergent behaviors formally[26]. The properties are automatically extracted from the mKAOS model that describes the SoS. These properties are then verified using statistical model checking **SMC**.

mKAOS is a Domain Specific Language (DSL) for mission modeling in the context of system-of-systems, it was developed to tackle the problem of properly representing missions in this context, it provides a way of defining what are the objectives of the SoS[27]. mKAOS provides a set of seven kinds of diagrams, the so-called *mKAOS Mission Models* is the most important diagram (it defines global missions, individual missions, and specifies how those are related to each other) is responsible for modeling the objectives of an SoS, the individual missions and how those objectives are related to the CSs, data objects, emergent behaviors, etc.

In traditional development methodologies, the software architecture is the main artefact in the design process, describing the structure and the behavior of the system. In the SoS context, an architecture describes the CSs and their interactions[26], there are additional challenges when designing architectures for SoS, the dynamism of such kind of systems makes the whole design process difficult. Since this work focus on verification, so for the verification of models, this dynamism can impact the feasibility of traditional exhaustive approaches. To tackle this issue **SMC** is proposed[27].

SMC is a probabilistic, simulation-based technique intended to verify, at a given confidence level, if a certain property is satisfied during the execution of a system. SMC was proposed to support the formal verification of architectural properties in dynamic systems[28]. According to Silva[26] SMC relies on simulation, considering a finite sequence of execution states and probabilities to identify whether an architecture satisfies its constraints given a certain degree of confidence, limiting the problem space.

DynBLTL is a language for expressing the properties in such a manner that they can be used by SMC in the simulation process. It allows the dynamic bound of operations, allowing the system to maintain execution states with a degree of uncertainness. DynBLTL was built to support the formalization and simulation of dynamic systems (systems that can change its architectural configuration at runtime, such as SoS)[27].

2.8 Worke of Webster et al.: An Ontology for Evaluation of Network Enabled Capability Architectures

Webster et al. propose a framework for the evaluation of architectures and enables the comparison of different architectures and their use. The framework identifies the elements needed to perform such an evaluation, how those elements are used and how they should be obtained or developed. The framework is built on measures of effectiveness (MoEs) and measures of performance (MoPs) and is split into a layer that validates configurations of services at a high conceptual level and a layer that verifies concrete resources against specifications. These verification layers are linked together by an integration layer that allows conceptual designs to be evaluated and also allows resource implementations to be verified for use in configurations. Evaluation of MoE is based on specified function and QoS attributes, and the evaluation of MoP is based on implementations that verify the specifications[3]; resources verified in this way can be included in the configuration.

According to Webster MoPs are used to measure the performance, or more generally the properties (connectivity, performance, flexibility and dependability), of a system, component or service and to

validate implementations of service descriptions, whilst MoEs are used to measure the effectiveness of a system or system of systems within a defined or given scenario, how the system meets the goals of the operation and to assess a configuration of services composed to implement a capability.
MoEs are applied using a configuration of service descriptions and can therefore be applied at a conceptual level, whilst MoPs are used at an implementation level to validate individual service implementation against service descriptions, service descriptions are used by the MoEs, so there is a decoupling between the evaluation of a physical implementation (concrete level) and the evaluation of a capability (conceptual level), this decoupling is made by an integration layer to allows the evaluation to be useful at several stages in a procurement lifecycle.
For more details, the upper ontology describes the major parts of the Evaluation Framework.

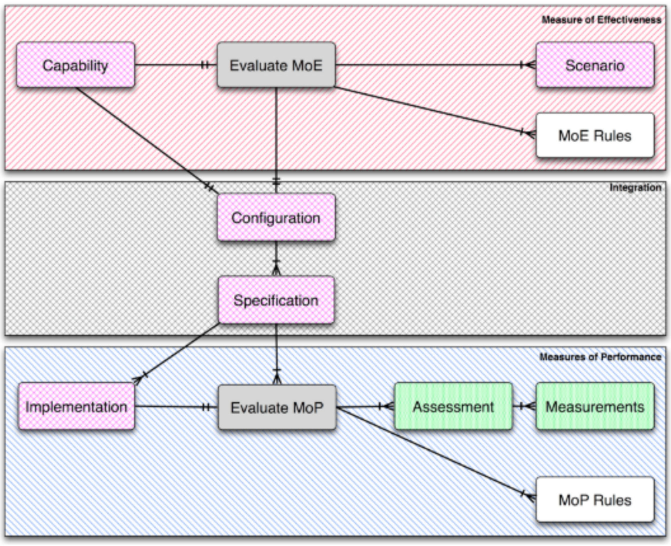

Figure 4: Upper Ontology[3]

The upper ontology is divided into three layers that allow a decoupling between a conceptual definition of a capability and a physical resource, the three layers are:

- **MoE Layer:** this layer utilizes a view of capability in the context of a scenario combined with an abstract definition of a capability defined by the integration layer.
- **Integration Layer:** this layer defines an abstract configuration based on service specifications.
- **MoP Layer:** this layer defines the actual MoP for all implementations of a service that will be used to implement the configuration given in the integration layer.

Each element in the upper ontology defined in Figure 4 is defined in terms of a procedure, so:

- **MoE Layer:**
  - **Capability:** this procedure will define how to generate a document (a component of the meta-model "Upper Ontologie" which is a procedure that creates an evaluation) that will capture a Capability definition for use in the evaluation framework.
  - **Scenario:** this procedure will define how to generate a document that will capture a Scenario in which the Capability document is utilized.

- **MoE Rules:** this procedure will derive rules to access the MoE of the capability based on the Scenario document. The generated set of rules will then be applied to the integration layer documents to derive an MoE for the specified configuration in the context of the scenario.
- **Evaluate MoE:** this is the process that applies the MoE rules to a configuration to assess its MoE in the context of a Scenario.

• **Integration Layer:**

- **Configuration:** this procedure defines how a document capturing an abstract configuration of services should be defined.
- **Specification:** this procedure will define how to generate service Specification documents. This procedure will draw on existing architectural definitions of service specifications.

• **MoP Layer**

- **Implementation:** this procedure will define an implementation of a Specification.
- **MoP Rules:** this procedure will derive a set of rules based on the Specification to evaluate the Implementation against the Specification and determine an MoP.
- **Assessments:** this procedure will define criteria for selecting appropriate assessment methodologies and documenting them in a form when they can be used by the Evaluation Framework.

2.9 Test and evaluation of systems of systems

Authors in [29] address the unique aspects of T&E of SoS and outlines strategies and techniques for handling them.

2.9.1 What is T&E?

The process by which an SoS or its constituent systems are compared against capability requirements and specifications.

2.9.2 T&E in the SoS Systems Engineering Process

Systems of systems (SoS) differ from traditional systems in several ways. So it requires the application of systems engineering to the SoS .The most important SoS systems engineering elements, that some of which are critical to T&E of the SoS are:

1. Translating capability objectives.
2. Understanding system and relationships.
3. Monitoring and assessing changes.
4. Addressing requirements and solution options.
5. Orchestrating upgrades to SoS.
6. Assessing performance to capability objectives.

7. Developing and evolving SoS.

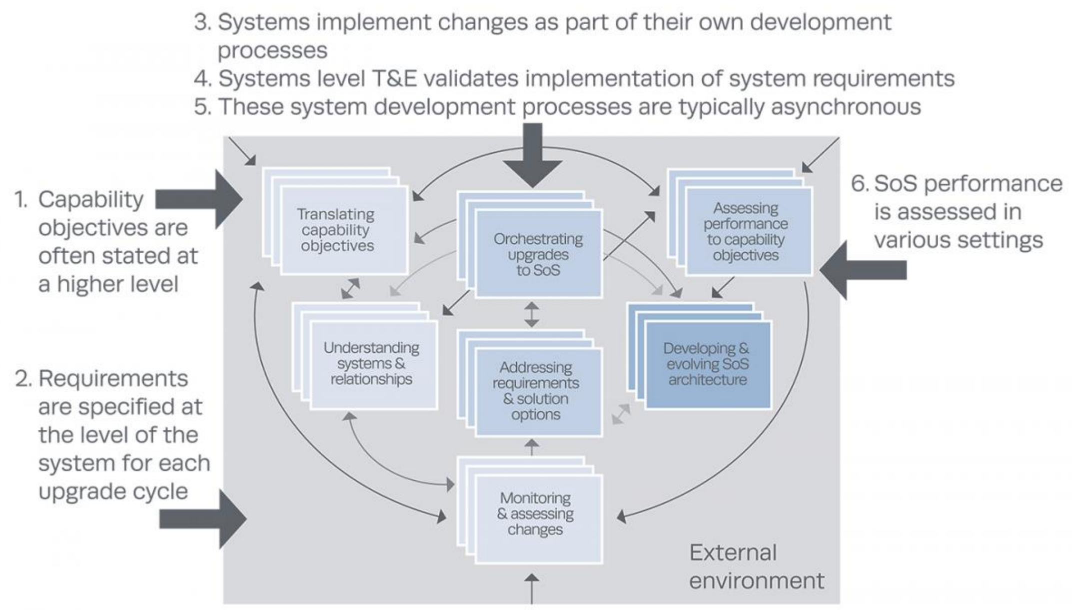

Figure 5: SoS Systems Engineering Core Elements and Their Relationships to T&E[29].

Figure 5 shows the SoS Systems Engineering Core Elements and how T&E activities fit into the SoS systems engineering core elements.

2.9.3 Best practices for T&E of SoS

- **Approach SoS T&E as an evidence-based approach to addressing risk:** Full conventional T&E may be impractical for incremental changes to SoS due to the difficulty in bringing all constituent systems together. That's why an incremental risk-based approach to identifying key T&E issues is recommended. This approach identifies areas critical to success and areas that could have adverse impacts on user missions. Risk is assessed using evidence from different sources. The evidence can be based on activity at the SoS level, or on roll-ups of activity at the CSs level.The activity can range from explicit verification testing, results of models and simulations, use of linked integration facilities.

- **Encourage development of analytic methods to support planning and assessment:** The use of analytical models of the SoS as a tools to assess system-level performance against SoS scenarios, and to validate the requirements allocations to systems and provide an analytical framework for SoS-level capability verification.

- **Address independent evaluation of networks that support multiple SoS:** The network has assumed a central role as a unique constituent of every SoS.The assessment of SoS performance demands evaluation of both network performance and potential for degradation under changing operational conditions.

- **Employ a range of venues to assess SoS performance over time.**

- **Establish a robust process for feedback once fielded:** By establishing robust feedback mechanisms between field organizations and their operations and the SoS systems engineering and management teams, SoS T&E can provide a critical link to the ongoing operational needs of the SoS.

2.10 Discussion

The various papers presented provide interesting concepts for the evaluation of missions. However, none of the works describes in detail the link between the metrics, the constituent systems of an SoS as well as the mission. In fact, it is necessary to understand how several potential architectural solutions might enhance capability and to supply the means to choose between various architectural solutions. That's why we want to address this challenge, relying on the metrics presented in this chapter.

2.11 Conclusion

In this chapter, we've presented the state of the art in the evaluation of the architectures of an SoS, and we presented some works that are interested in the evaluation of an SoS and we've discussed them.

Chapter 3

3 CONTRIBUTION: SYSTEM OF SYSTEMS MISSION EVALUATION FRAMEWORK

3.1 Introduction

System engineering is a multidisciplinary and holistic approach to develop solutions for complex engineering problems. The continuing increase in system complexity demands more rigorous and formalized systems engineering practices. In response to this demand along with advancements in computer technology, the practice of SE is undergoing a fundamental transition from a document-based approach to a model-based approach, the emphasis shifts from producing and controlling documentation about the system to producing and controlling a coherent model of the system[30].

3.2 Model Based System Engineering

MBSE is defined as the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases[25]. A traditional system engineering approach focuses on the development of textual specifications and design documentation, while MBSE focuses on the development of a coherent system model that consists of requirements, design, analysis, and verification and is characterized as a "model-centric" approach.
In comparison to the traditional approach, MBSE provides a more rigorous method for capturing, integrating, and maintaining outputs of systems engineering activities.
Research in the SoSE domain claim that the use of MBSE to model systems is a promising path, the adoption of MBSE approach brings five benefits according to the INCOSE: (i) improved communications, (ii) increased ability to manage system complexity, (iii) improved product quality, (v) enhanced knowledge capture and reuse of information, and (vi) improved ability to teach and learn SE fundamentals[5].

3.3 System Modeling Language SysML

SysML is commonly used in MBSE. It is a graphical modeling language devloped by the Object Management Group (OMG) to be used for modeling a wide range of systems engineering problems. It's not dependent on any single systems engineering method and is intended to support multiple methods[25].
According to [30] SysML is a graphical modeling language with a semantic foundation for representing requirements, behavior, structure, and properties of the system and its components. It is intended to model systems from a broad range of industry domains such as aerospace, automotive, health care, and others.
Figure 6 depicts the SysML diagram Taxonomy:

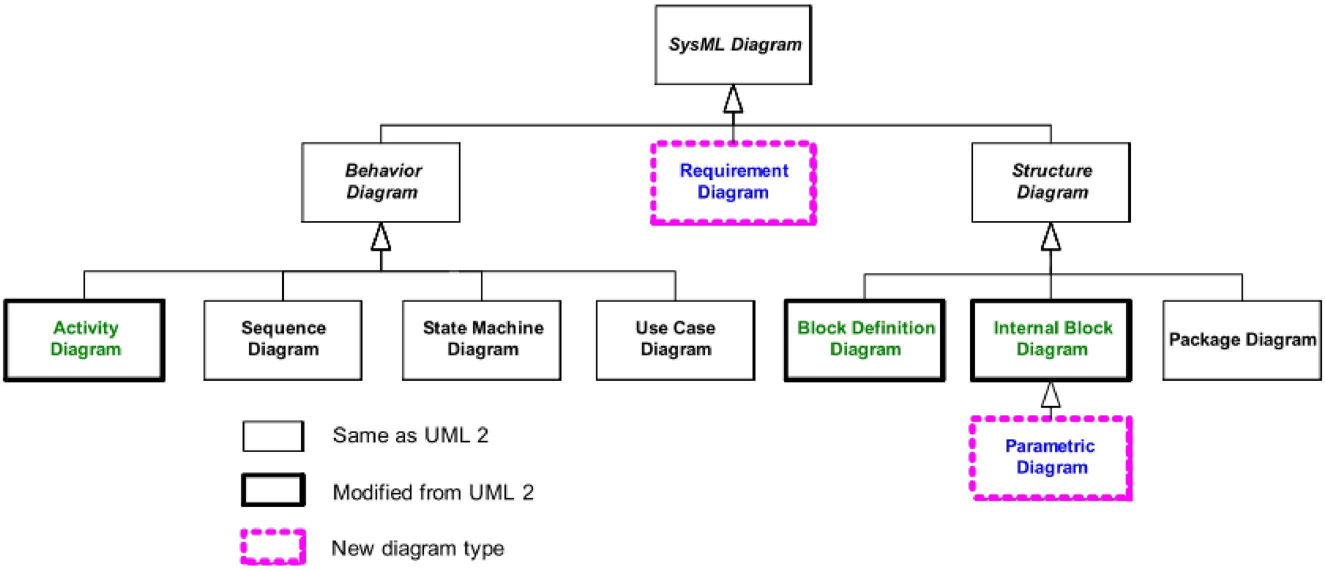

Figure 6: SysML diagram Taxonomy[4].

This is a brief explanation of these diagrams[31]:

- **Block definition diagram (BDD):** is used to display elements such as blocks and value types and the relationships between those elements.
- **Internal block diagram (IBD):** is used to specify the internal structure of a single block i.e IBD shows the connections between the internal parts of a block and the interfaces between them.
- **Use case diagram (UC):** is used to convey the use cases that a system performs and the actors that invoke and participate in them. A use case diagram is a black-box view of the services that a system performs in collaboration with its actors.
- **Activity diagram (ACT):** is used to specify a behavior, with a focus on the flow of control and the transformation of inputs into outputs through a sequence of actions.
- **Sequence diagram (SD):** is used to specify a behavior, with a focus on how the parts of a block interact with one another via operation calls and asynchronous signals.
- **State machine diagram (STM):** is used to specify a behavior, with a focus on the set of states of a block and the possible transitions between those states in response to event occurrences.
- **Parametric diagram (PAR):** is used to express how one or more constraints (specifically, equations and inequalities) are bound to the properties of a system. Parametric diagram support engineering analyses, including performance, reliability, availability, power, mass, cost, and others.
- **Package diagram (PKG):** is used to display the way a model is organized in the form of a package containment hierarchy.

- **Requirements diagram (REQ):** is used to display text-based requirements, the relationships between requirements, and the relationships between requirements and the other model elements that satisfy, verify, and refine them.

3.4 Life Cycle Adopted and General Approach

Our work is an extension of the work of CHERFA .So we rely on the same life-cycle presented by the later, which is the wave model. It consists on five steps, they are depicted in Figure 7: (i) initiate mission, (ii) conduct mission analysis, (iii) develop mission architecture, (iv) evaluate mission, (v) implement updates. The engineering activities that we used and that are contained in each step are described in the following.

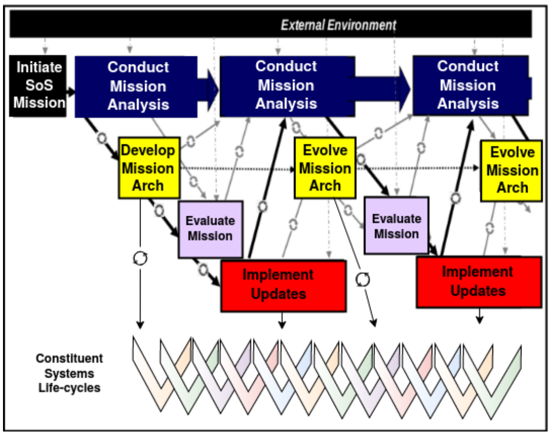

Figure 7: Adopted Life Cycle[5]

The mission structural and behavioral model with the parametric model and the concrete architecture are charged to analyze the performance metrics and compare them to those already defined. The mission functional model detects deficiencies and problems, and from this, we take architectural decisions and decide about updates and implement them.
Our general approach is presented in Figure 8

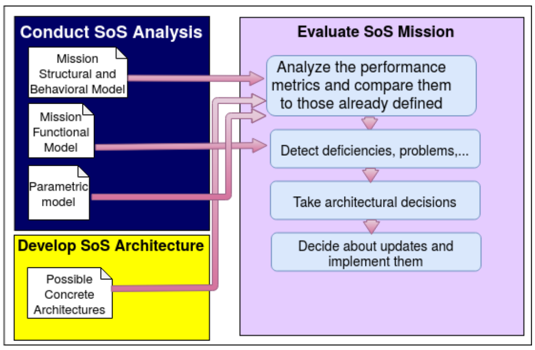

Figure 8: General Approach[5]

3.5 Mission Evaluation Conceptual Model

With the SysML diagrams, we will describe and create global models of the system spanning the various independent systems detailing interaction points, depicting entity relationships in disparate models and to have an overall graphical representation of the system and outlining the principal behaviours.

The meta-model we used for our SoS is presented by CHERFA et al[5] We proposed the extension of this conceptual model based on the modeling experience using the SysML language. The concepts we added imply two INCOSE metrics TPM and KPP, and we also the scenario. These concepts are defined and highlighted in the figure 9 The **mission** is the main concept in the conceptual model.

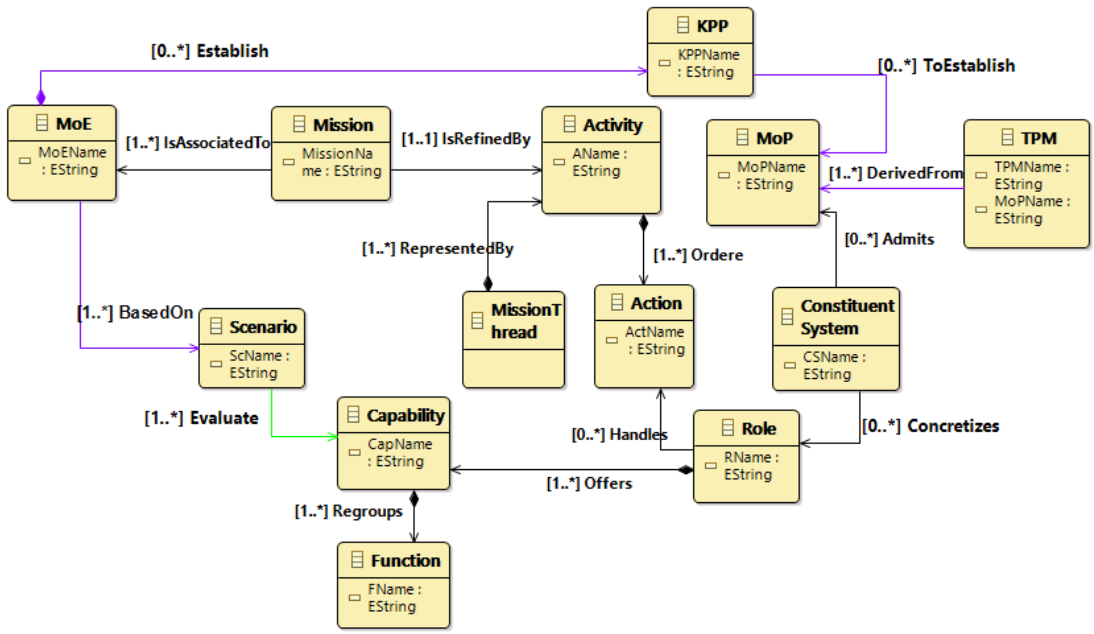

Figure 9: SoS meta model

The **mission** is the goal that the SoS must achieve by collaborating constituent systems(CSs), where each mission is associated with an **activity**, and the activity orders a set of **actions**.

In our process, each mission is associated with effectiveness measures that must be determined, where **MoEs** are used to measure the fitness of an SoS to fulfil the mission.

The **mission threads** are the description of the end-to-end set of **activities** that serve to accomplish a specific subset of the mission goals and objectives. The **role** is an abstraction of the characterization of the ideal behaviour that will fulfil an **Action**, it handles action and gathers the required competencies (**Capability**) to play the role needed to accomplish the **Action**, and the **capability** of the **role** is the ability to provide some expertise to the wider needs of an SoS. It is formed through the integration of several **Functions**. The **constituent systems** are the systems that compose the SoS and could be used to concretize a **Role** in the concrete architecture, a CS is chosen when its capabilities match those required by a role and by considering the **MoPs**. CS is integrated to meet a Role Capability. **MoP** measures the performance or the properties of a system, component or service, it can be compared with its own specifications and with other systems that perform the same functions, they are used to assess whether the system meets design or performance requirements that are necessary to satisfy the **MoEs**. MoP maps the **KPP** in the system specification. **TPM** measures the attributes of a system element to determine how a system or a system element is satisfying a technical requirement or goal and is derived from MoP. **KPP** is used by the developer to establish the key requirements necessary to achieve the MoEs. Used to establish the MoPs which are measured as soon as possible and repetitively throughout development testing and evaluation. The **Scenario** is used to evaluate a capability, many scenarios will be defined to evaluate one capability, according to Webster[3] et al "The scenario must be generated to exercise the capability. Many sets of scenarios may be required to completely evaluate a capability".

3.6 Emergency Response System of System Case Study

Emergency events refer to the harm and disasters caused by natural or man-made factors, or a combination of both. Natural disasters include earthquakes, tsunamis, tornadoes, hurricanes, volcanic eruptions, and floods. Man-made disasters include regional conflict, war, terrorist attacks, chemical/oil spills and explosions, and catastrophic accidents related to airplanes, trains, cars, and ships. A pandemic infection of a disease is a case of the combined effect of natural and human activities; it is first caused by natural factors and subsequently worsened by human factors and policy interventions[6].

ERSoS is a widely used example of SoS. ERSoS is responsible for providing emergency aid on demand to members of the public. This SoS encompasses existing agencies (such as fire, police, and hospital) with independently owned and managed systems nevertheless collaborate to deliver a service on which reliance is placed. ERSoS principal mission is to give the emergency response in case of a significant incident[5].
Our proposed system of systems encompasses several constituent systems, and based on the MBSE approach and the SysML diagrams we carried out our SoS model. The figure 10 shows the CSs and the boundaries of the ERSoS, where ER Call Center, Hospital, Maritime, Police, and Firefighter represent the constituent systems. Caller and Target are considered elements of the environment.

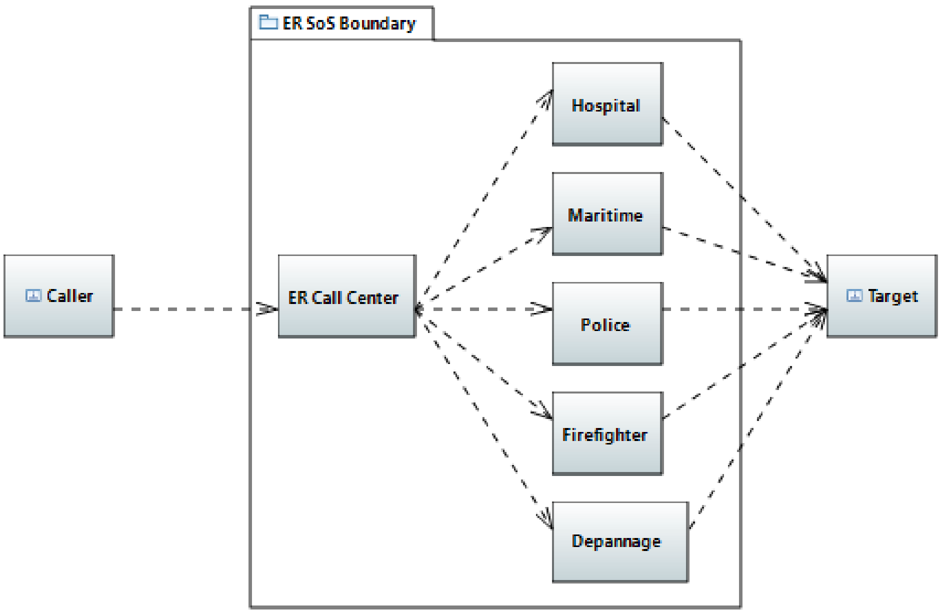

Figure 10: ERSoS Boundary

We consider ER Call Center as an important point because is responsible for receiving the calls, routing them, and managing the emergency response depending on the type of incident.

3.6.1 Block definition diagram

with the BDD we presented the constituent systems of our ERSoS and the relationships between them, where each block represent a CS.

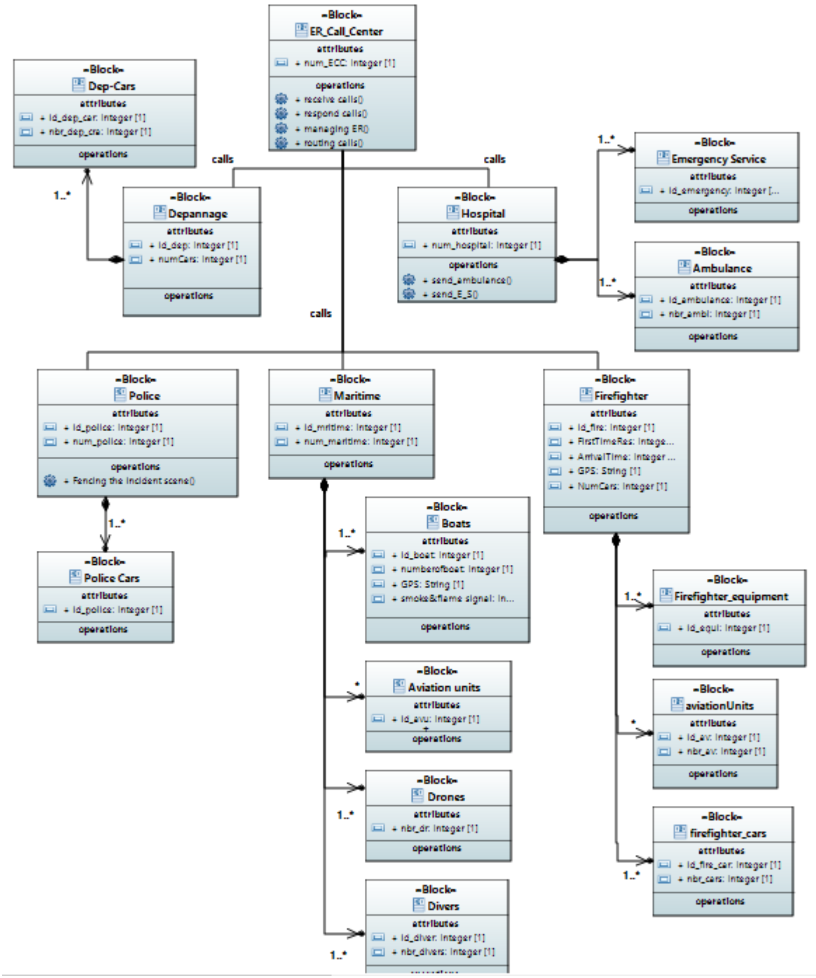

Figure 11: ERSoS block definition diagram

ER_Call_Center is responsible for receiving calls and responding to the caller then managing the emergency and deciding the type of the incident, and finally, choosing the specific constituent systems depending on the incident (**Hospital, Police, Maritime, and Firefighter**). Then the chosen CSs take care of the incident.
The following figure 12 shows the block definition diagram of the case study of people lost at sea.

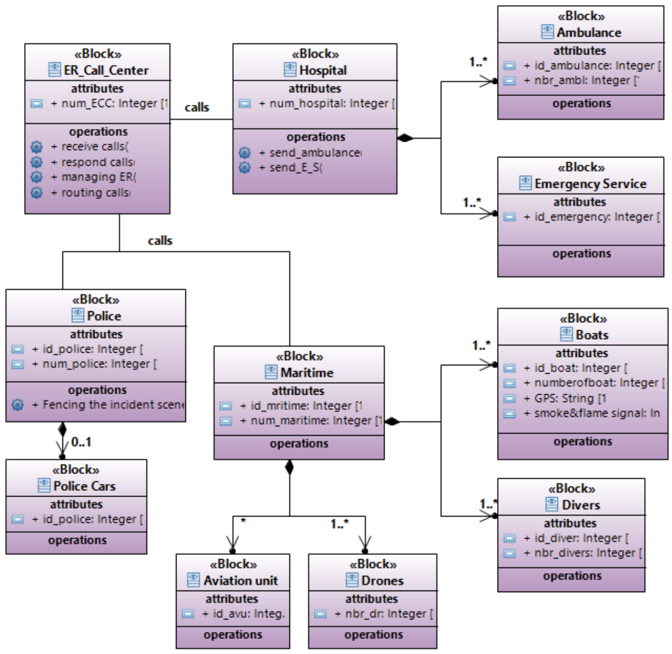

Figure 12: people lost at sea_Bdd

3.6.2 Requirement diagram

The main mission of our SoS is to give Emergency Response in the case of an incident. Our case study is to **save people lost at sea**, to achieve this mission multiple submissions must be achieved. The first one is to **provide a fast first ER**, to fulfil this mission the **ER_Call_Center** must **decide the incident type, determine the incident's location** and **determine the ER units needed**. Then to achieve the global mission, those ER units have to **determine the equipment needed, fence the incident scene** and **save and evacuate victims**.

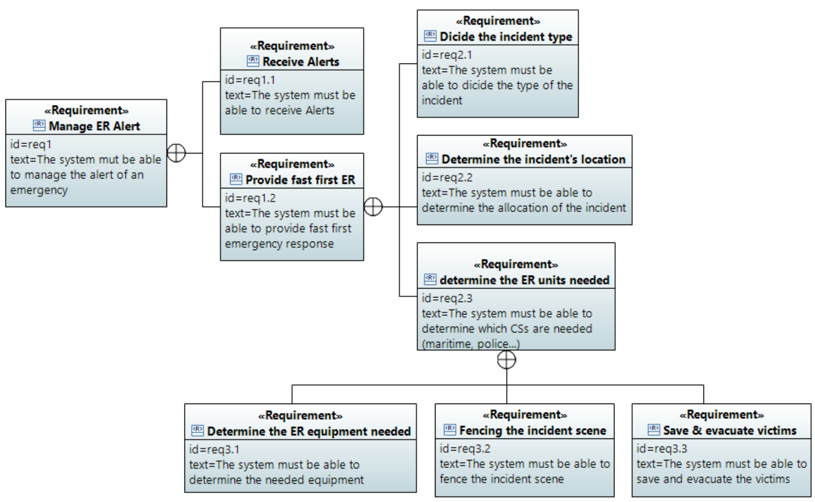

Figure 13: ERSoS requirement diagram

3.6.3 Activity diagram

We used the activity diagram to explain the ERSoS's CSs behaviors (activities or actions). With those activities, the constituent systems must satisfy the main mission of the SoS.

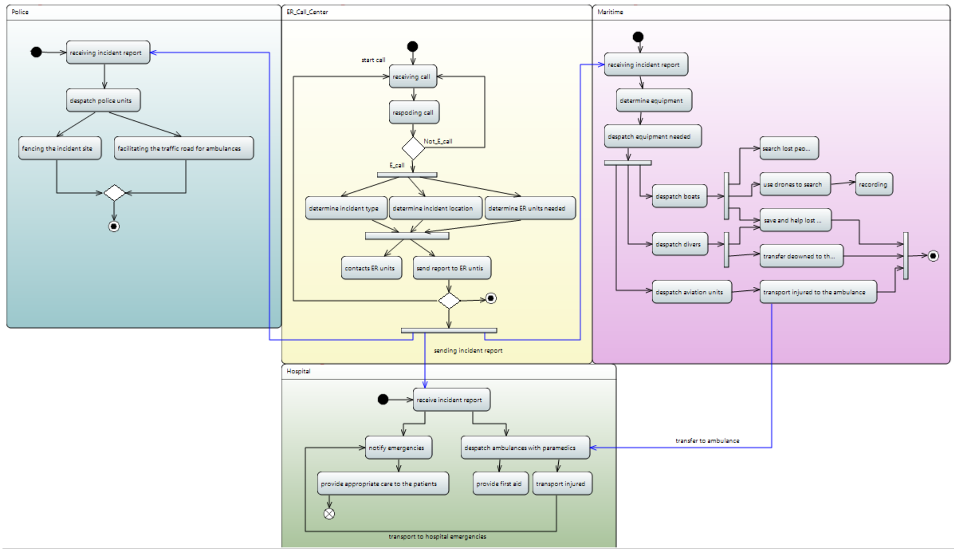

Figure 14: lost people at sea ERSoS activity diagram

What follows is the description of the activity of each system in the case of people lost at sea:

Emergency_Call_Center: The main activities of the emergency call centre are to receive and respond to the calls, determine the type of incident and its location, then he must be able to decide which ER units should send to cope with the incident. After that, it sends the incident report to the chosen units. Figure 15 shows the ER_Call_Center activty diagram:

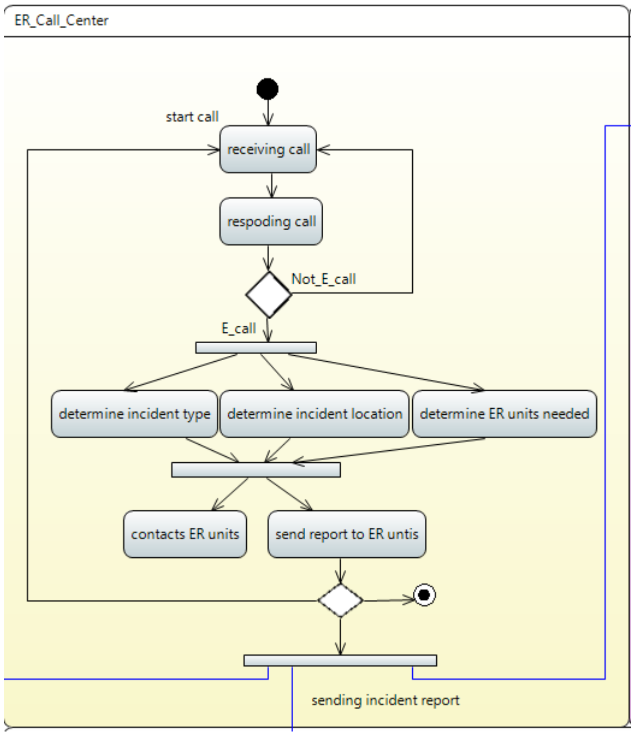

Figure 15: ER_Call_Center activty diagram

Maritime: After receiving the incident report maritime have to determine the equipment needed to save these people. From this equipment, we have boats to search and save lost people which use drones to find them quickly, and divers to help those who can't swim. Sending the aviation unit to evacuate the injured quickly to the ambulance. Figure 16 shows the Maritime activity diagram:

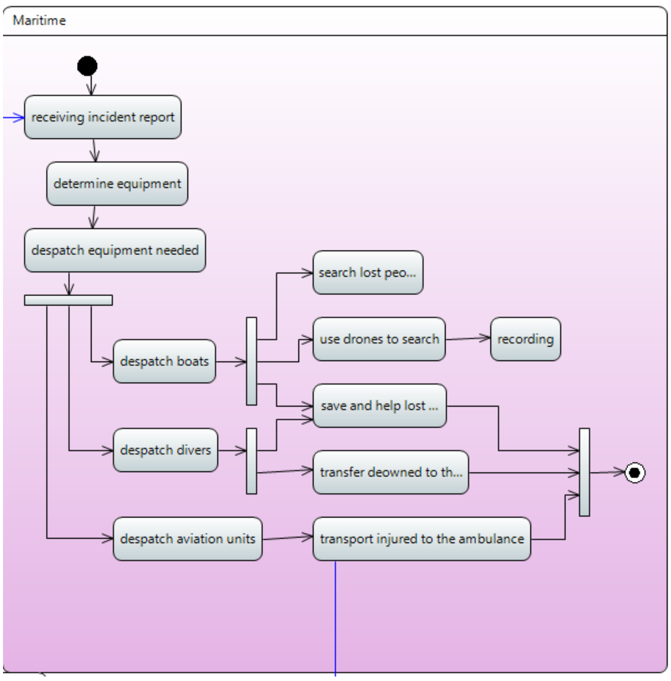

Figure 16: Maritime activity diagram

Hospital: After receiving the incident report, the hospital sends ambulances to the incident site with paramedics and then transports the injured as soon as possible to the hospital to receive appropriate care. Figure 17 shows the Hospital activity diagram:

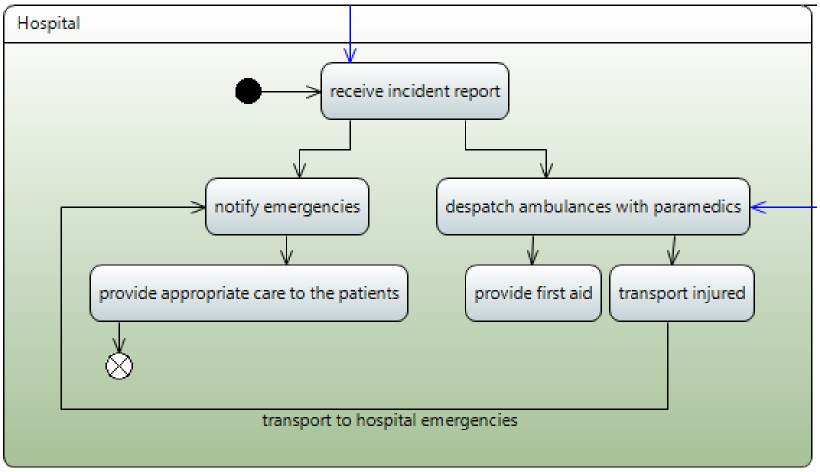

Figure 17: Hospital activity diagram

Police: Police activity consists on fencing off the incident site and facilitating the traffic road for the ambulances. Figura 18 shows the Police activity diagram:

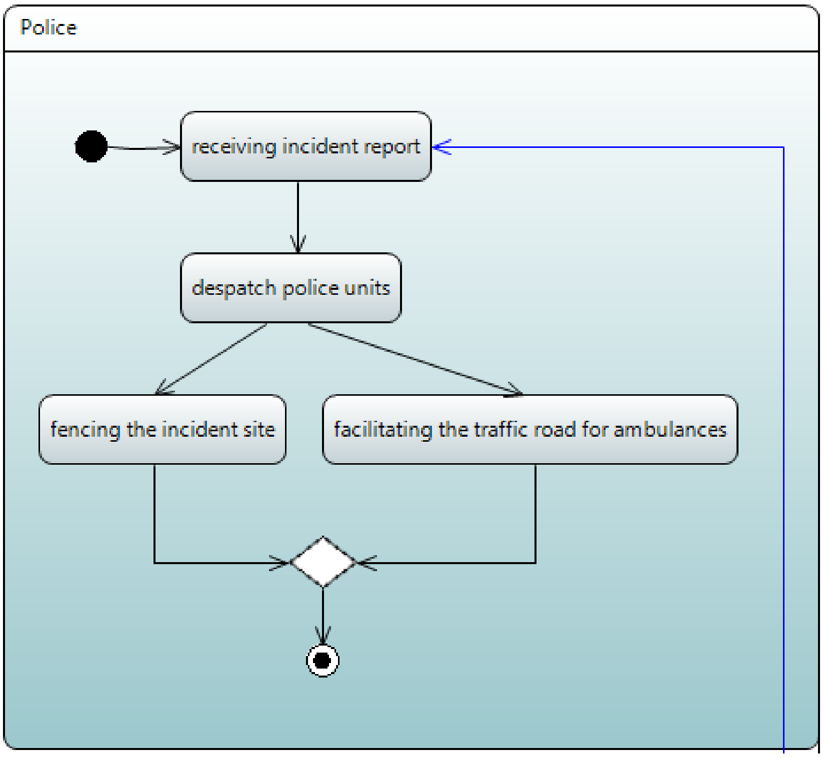

Figure 18: Police activity diagram

3.6.4 Parametric diagram

Based on the process proposed by CHERFA[5] to define MoEs and their indicators for each mission objective, we propose the following parametric diagram in figur 19 to define MoE for the **provide a fast emergency response** mission.

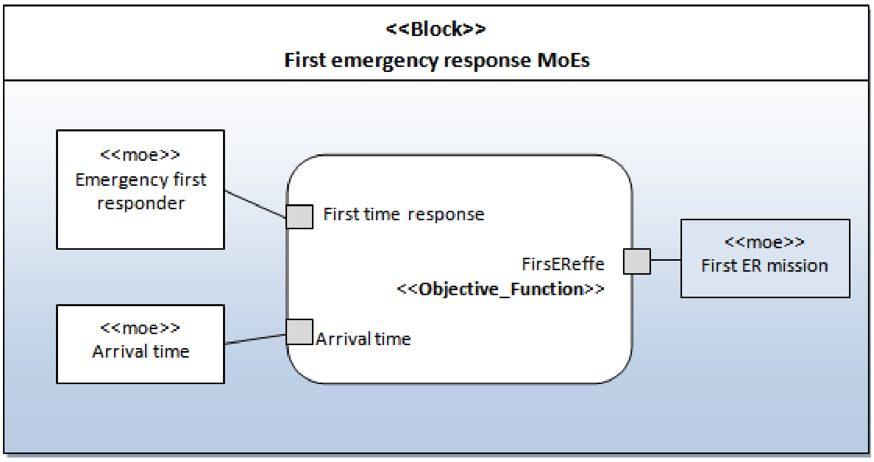

Figure 19: First emergency response MoEs

And for the MoPs, the figure 20 illustrates an example of a boat where availability, efficiency and cost are the most important MoPs for the boat, in this diagram we focused on efficiency.

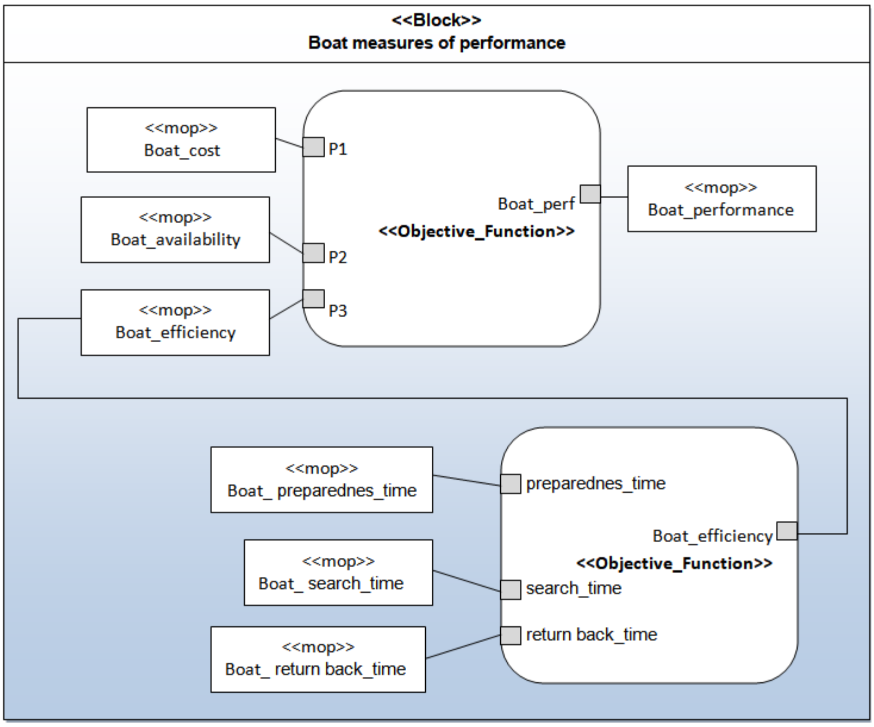

Figure 20: Boat measures of performance

- **preparednes_time:** is the time consumed from the moment that an emergency call arrives to the time the first boat despatches to the incident scene.
- **search_time:** is the time consumed in searching the lost people.
- **return back_time:** is the time consumed to return back and transport the patient to the ambulance.

3.7 Conclusion

In this chapter, we presented our conceptual model for our SoS, and then by using the SysML four pillars diagrams we presented the structure and the behavior of our SoS in the case of emergency response (ERSoS). We presented also our contribution to modeling the SoS highlighting the metrics and the procedures that can help the engineers to evaluate the SoS architectures.

Chapter 4

4 IMPLEMENTATION

4.1 Introduction

In this chapter, we presented the implementation of the models shown in the previous chapter in the context of an emergency response case study and the tools used.
We created a web application that helps engineers to make the best decision, to evaluate several SoS architectures and choose the one that satisfies their needs and the SoS's main mission.

4.2 Development environment

In our application we used the following tolls :

- Java EE (eclipse IDE).
- Mysql (Heidisql + Xampp).
- library in figure 21

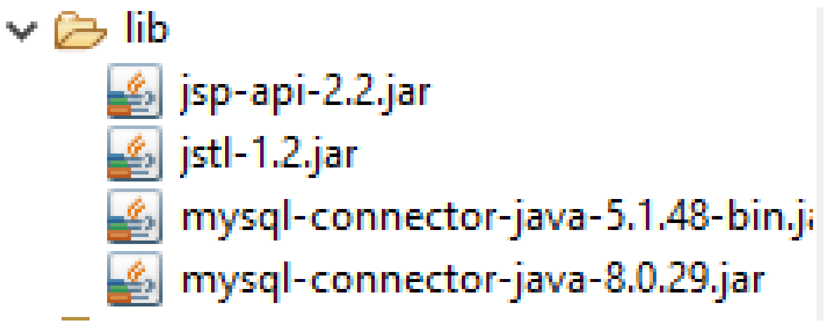

Figure 21: Library used in application

4.3 Presentation of application

1. In the first page of the application that we called EVSoS, the customer can choose the option that he/she wants in the menu bar(Home, About, Services, Testimonials, Get service, Sign up). As we can see in figure 22.

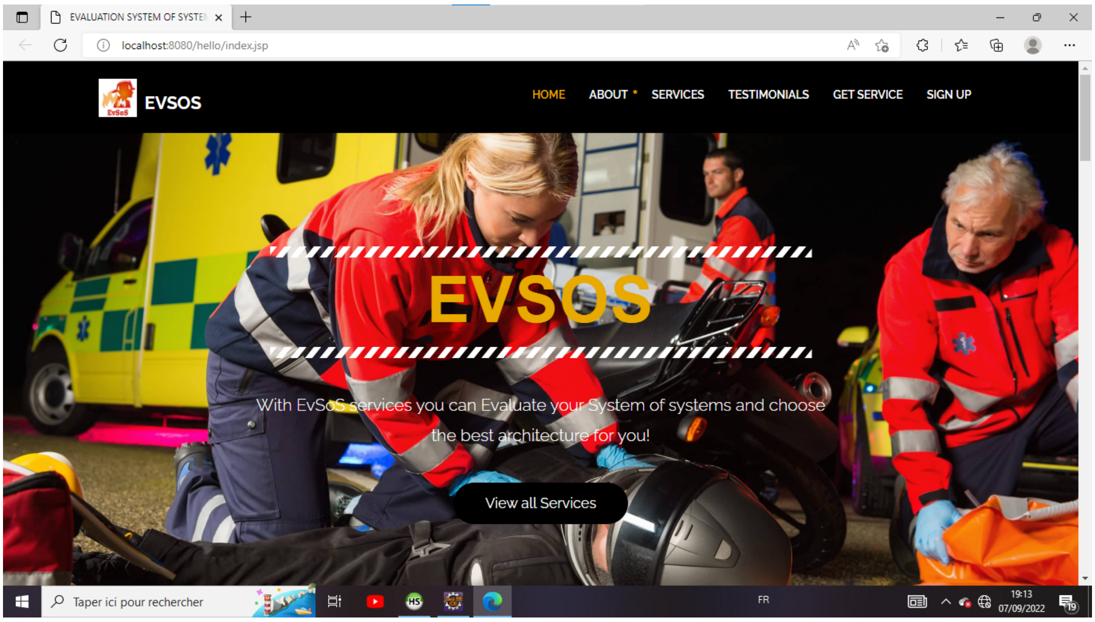

Figure 22: Home page

2. If the customer choose "ABOUT", a page appears, in which he/she finds in what and why we use the application, the purpose for using it, and a list of numbers of the emergency services. As we can see in figure 23.

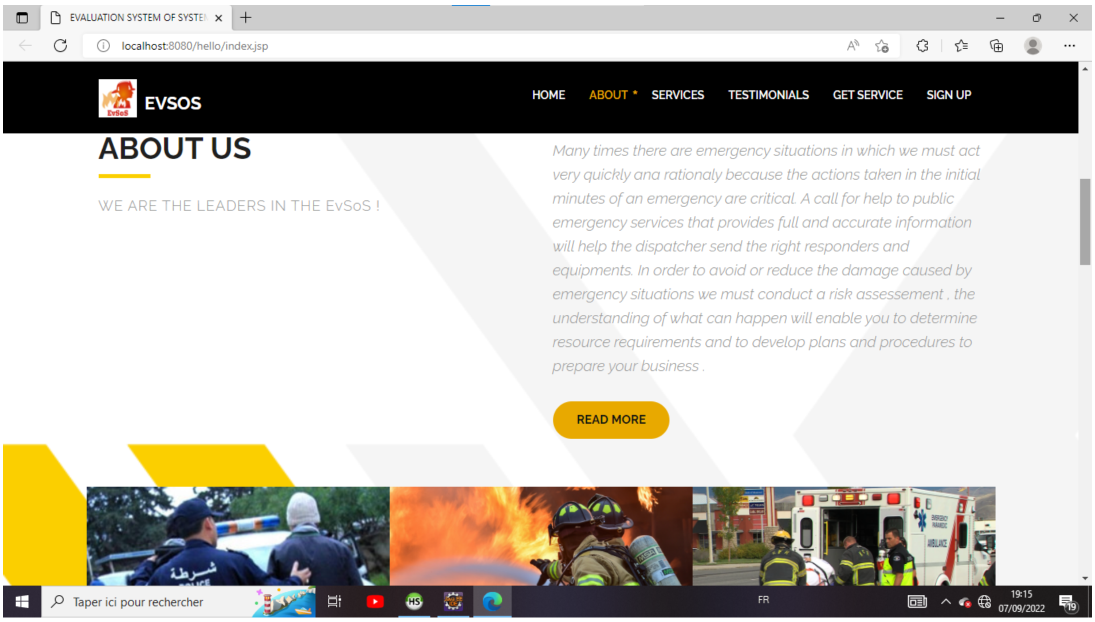

Figure 23: About page

3. If the customer choose "SERVICES", a page appears, in which he/she finds the services provided by the application(Evaluate SoS, Discover Architectures). As we can see in figure 24.

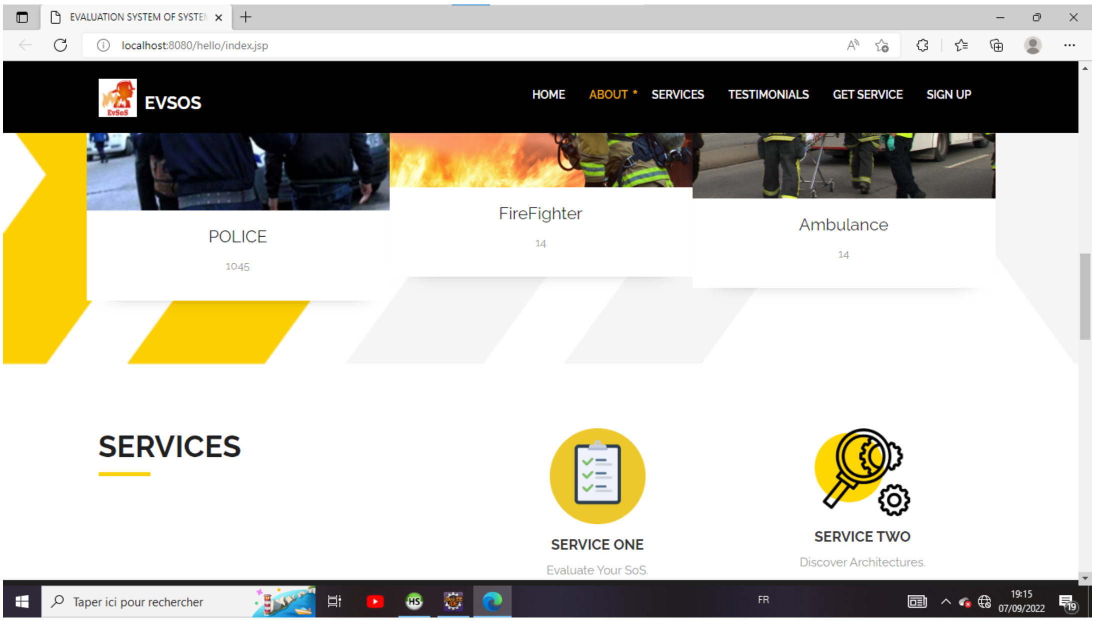

Figure 24: Services page

4. If the customer choose "TESTIMONIALS", a page appears, in which he/she sees some of testimonials and opinions of customers that used the application before. As we can see in figure 25.

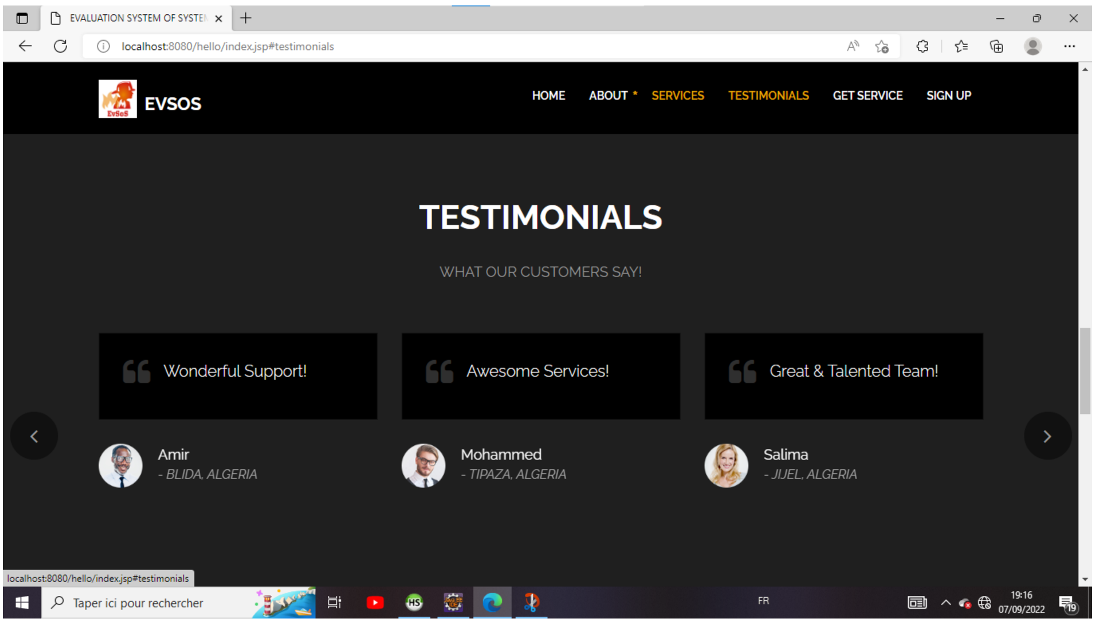

Figure 25: Testimonials page

5. If the customer choose "GET SERVICE", a page appears, in which he/she finds a form which is filled out by him/her.As we can see in figures 26, 27, and 28.

- The customer must already have an account to be able to access and get the desired service(with email and password).
- The customer must select the service that he/she wants(evaluate SOS, discover architectures).
- The customer must also select the type of the emergency (road accident, intentional violence and harm, lost people in sea, fire).

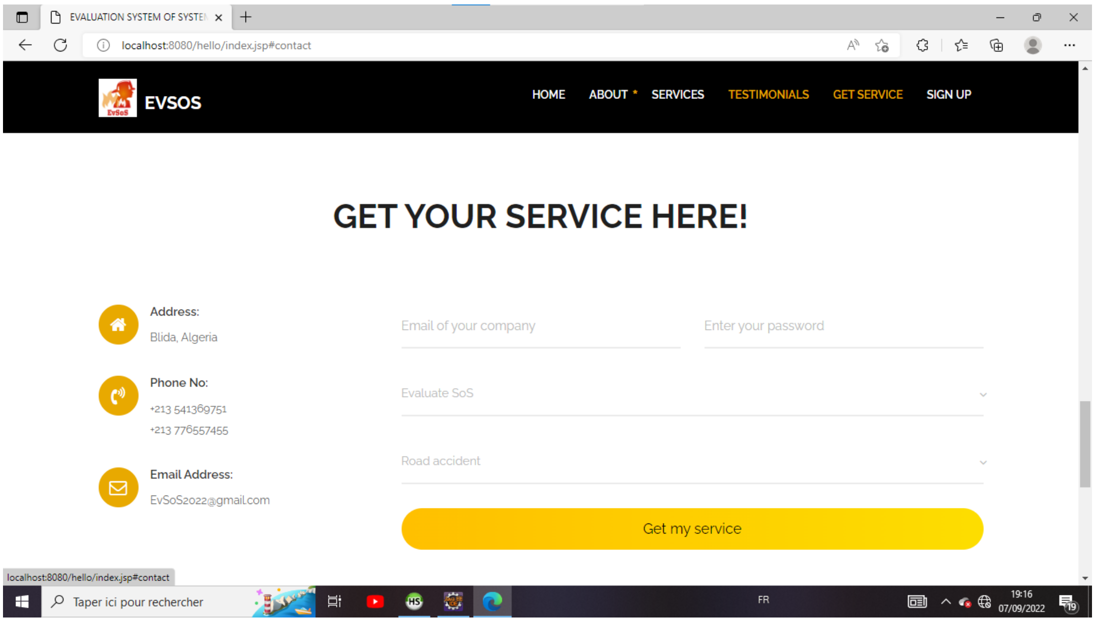

Figure 26: GET SERVICE page(1)

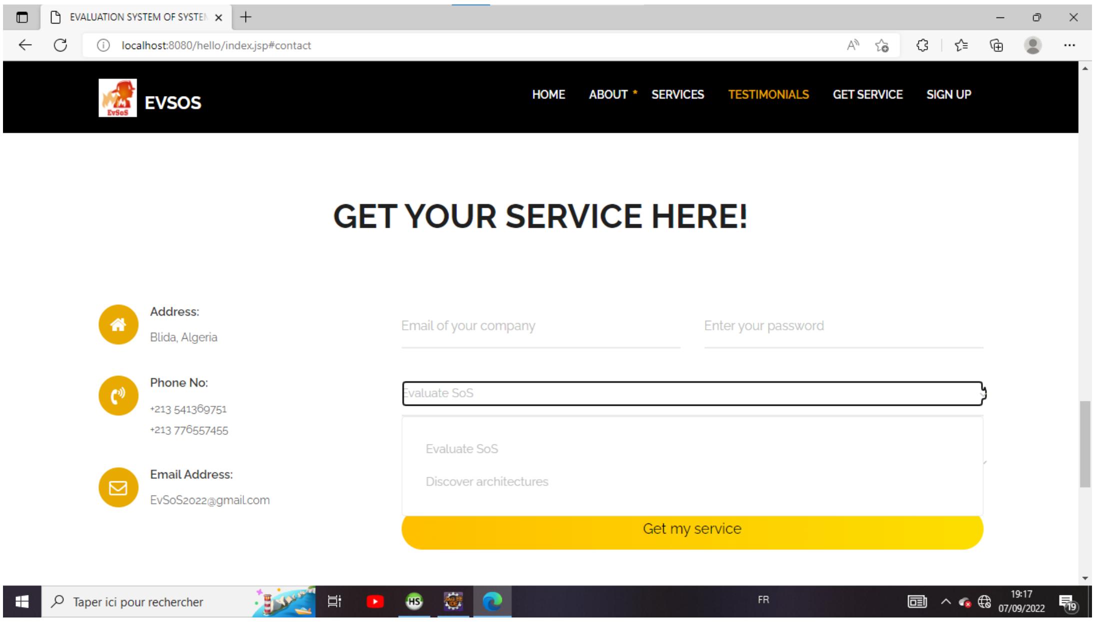

Figure 27: GET SERVICE page(2)

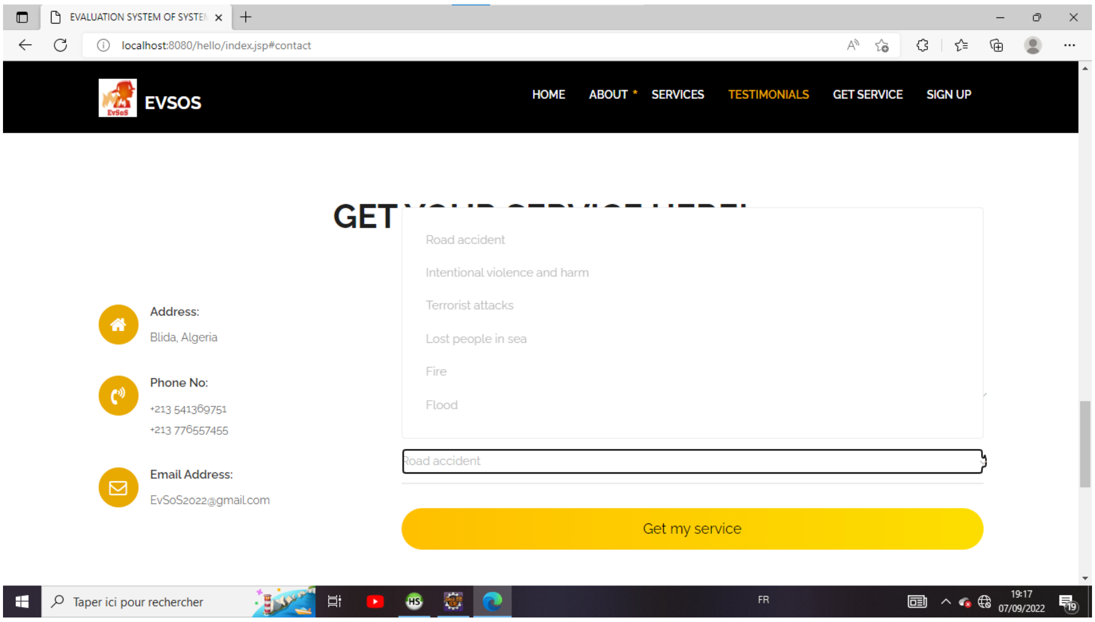

Figure 28: GET SERVICE page(3)

6. If the customer choose ”SIGN UP”, a page appears, in which he/she finds an option to create an account. As we can see in figure 29.

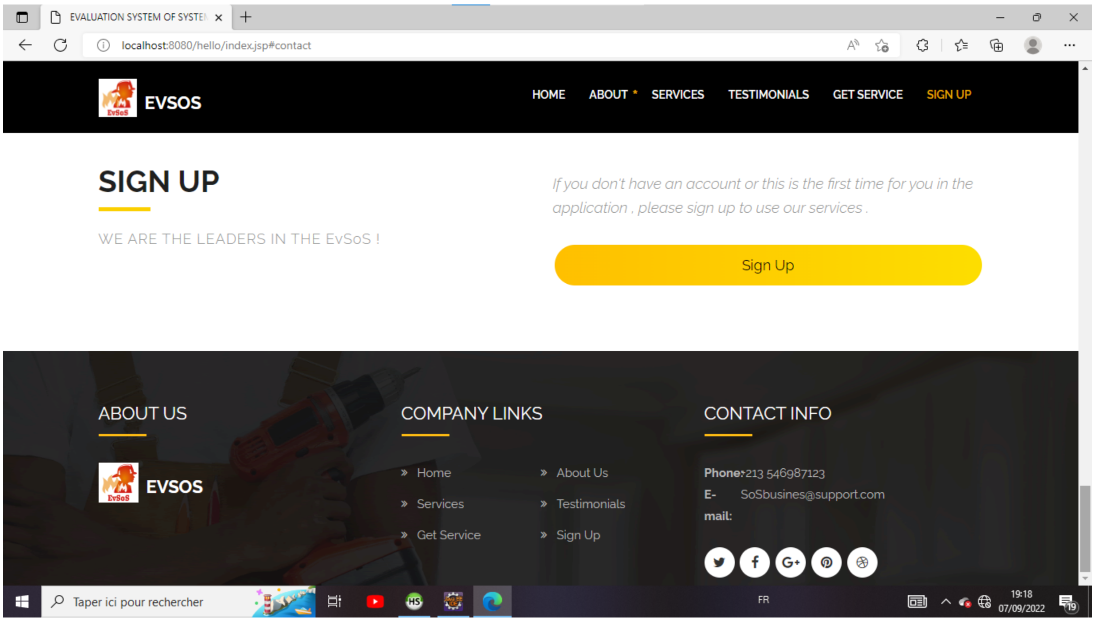

Figure 29: Sign up page

7. If the customer click on the button SIGN UP, a page appears, in which he/she finds a form which is filled out by him/her to create his/her account. He must enter his/her information(first name, last name, email, phone number,password). As we can see in figure 30.

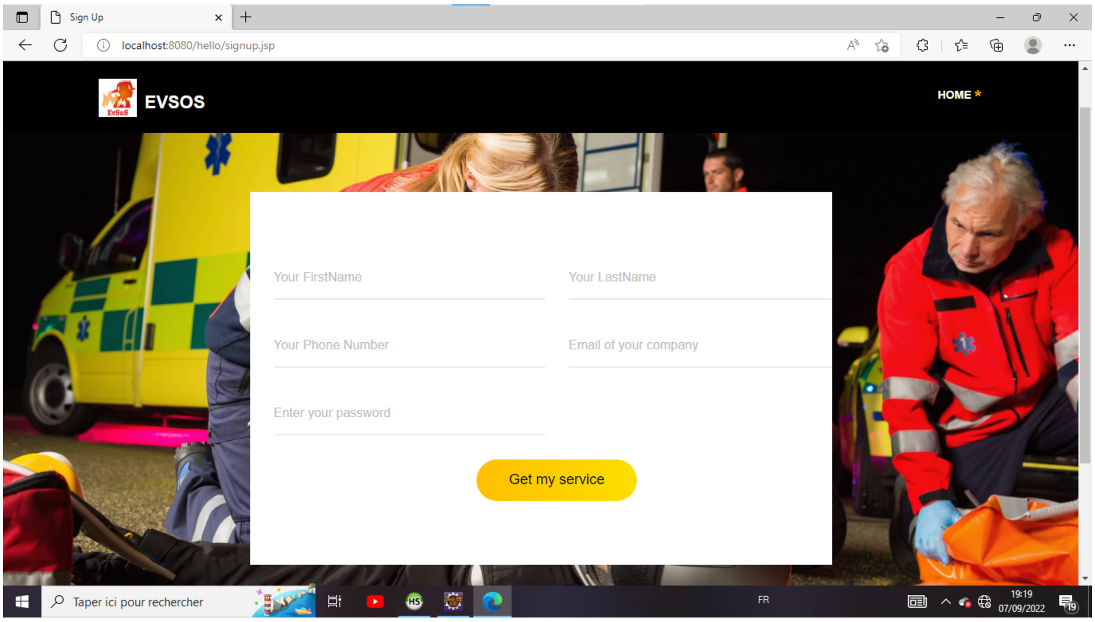

Figure 30: Create account page

8. We take an example, a customer that enter these information :

- First name : Dounia
- Last name : Sefta

- Email : seftadounia6@gmail.com
- Phone number : 0541063239
- Password : 123456

When he/she click on the button SIGN UP, the account created and the information saved in the database. As we can see in figure 31.

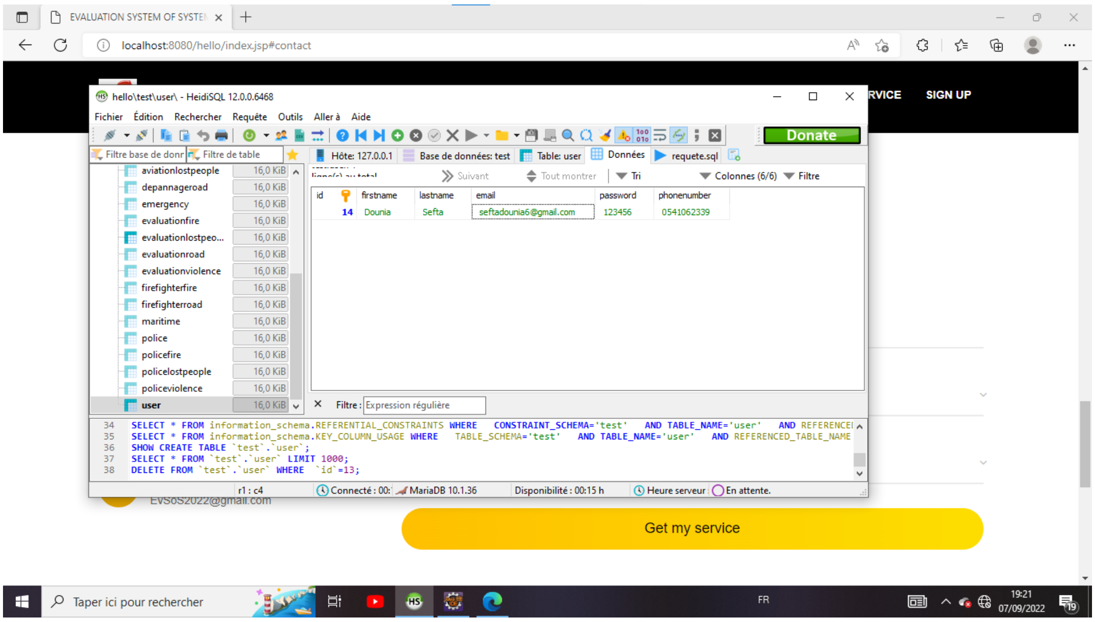

Figure 31: Customer in database

9. After the creation of account, the customer can get the service. He/She must enter his/her email and password, choose the service that he/she wants (evaluate SoS , discover architectures), and select the type of emergency. As we can see in figure 32.

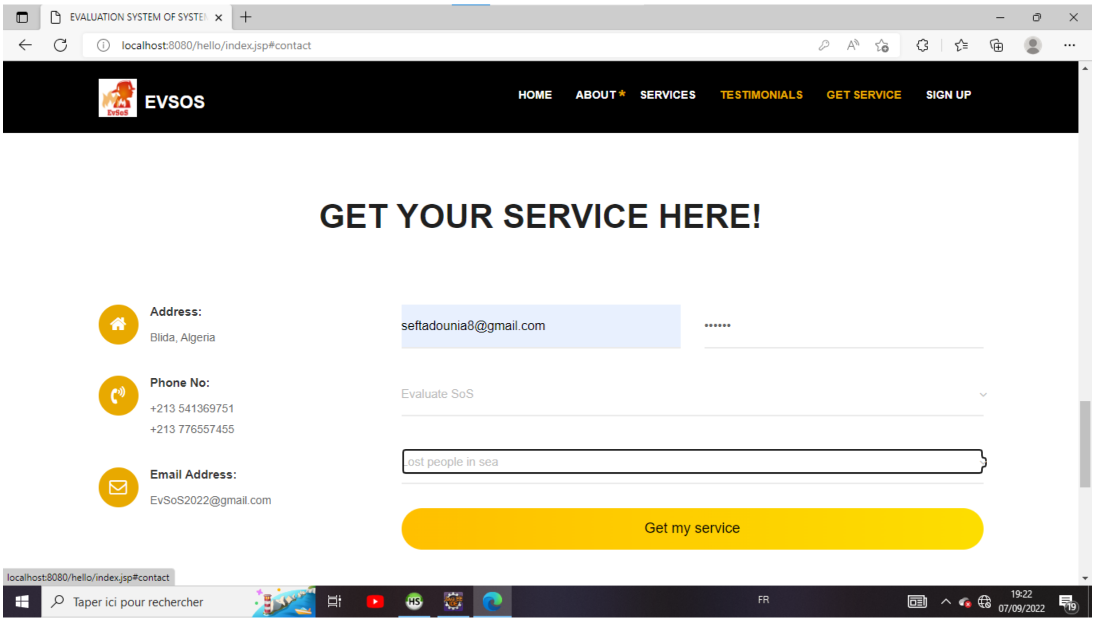

Figure 32: Customer get service page

10. If the email or password wrong, a page appears, containing an error message(You entered a wrong email or password). As we can see in figure 33.

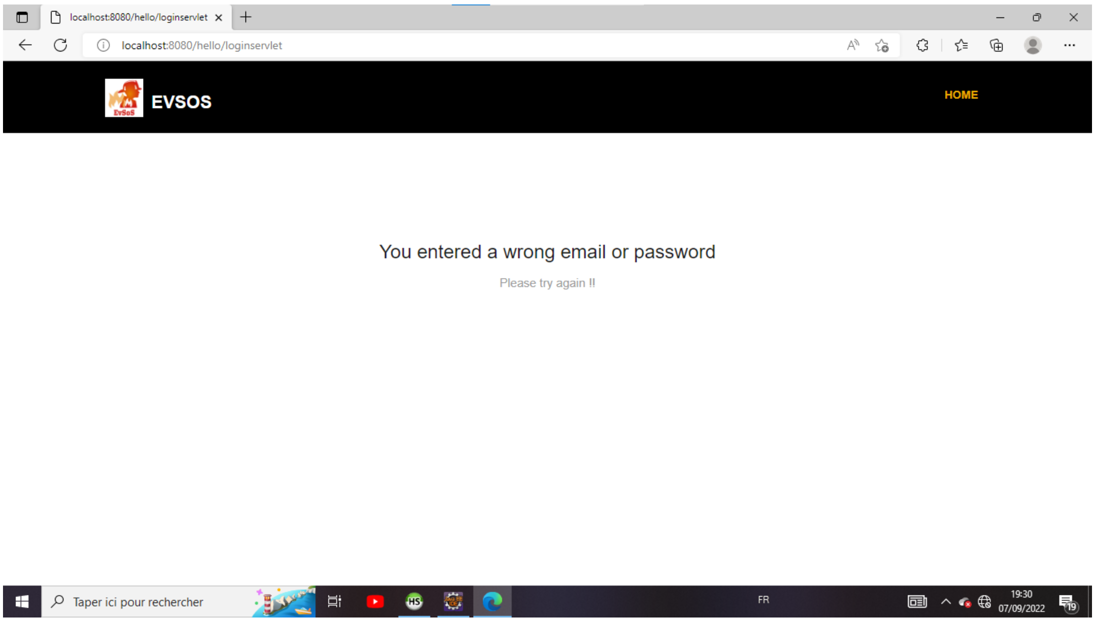

Figure 33: Customer with wrong email page

11. If he/she enter the right email and password. As we can see in figure 34.

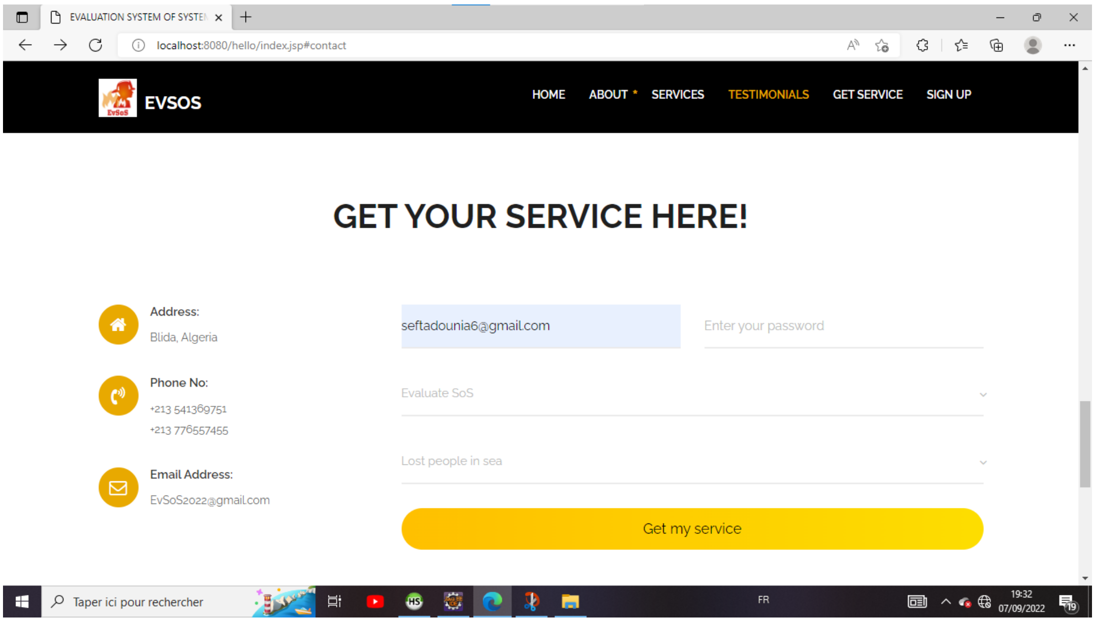

Figure 34: Customer with correct information page

12. In our example, the customer choose to "evaluate a SoS", and a type of emergency "lost people in sea", after the click on the button "Get my service", a page appears, in this page the customer must enter the information about police service. Where he/she finds the number of cars sending to the place of the emergency situation with or without gps, the first time response from the call center, the arrival time to the place of the emergency situation. As can see in Figures 35, 36.

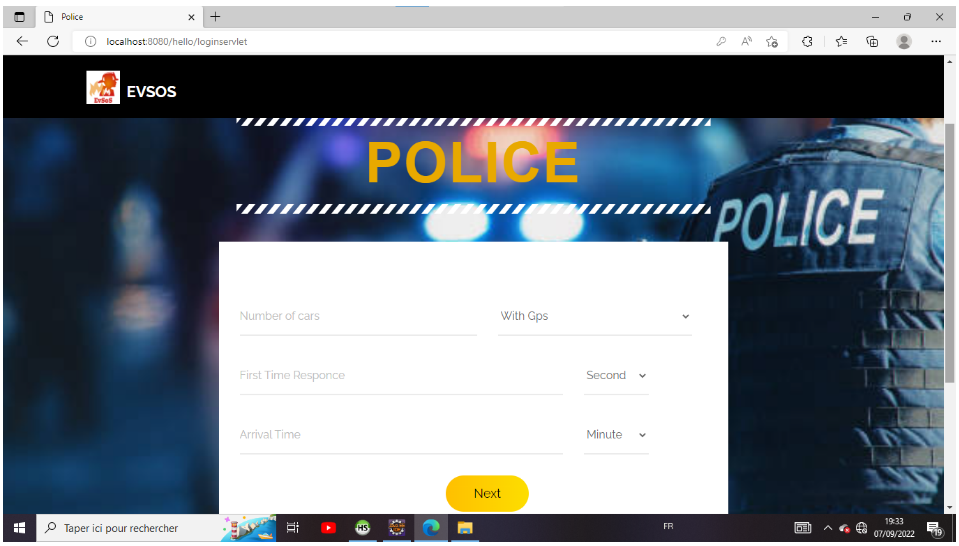

Figure 35: Police service page

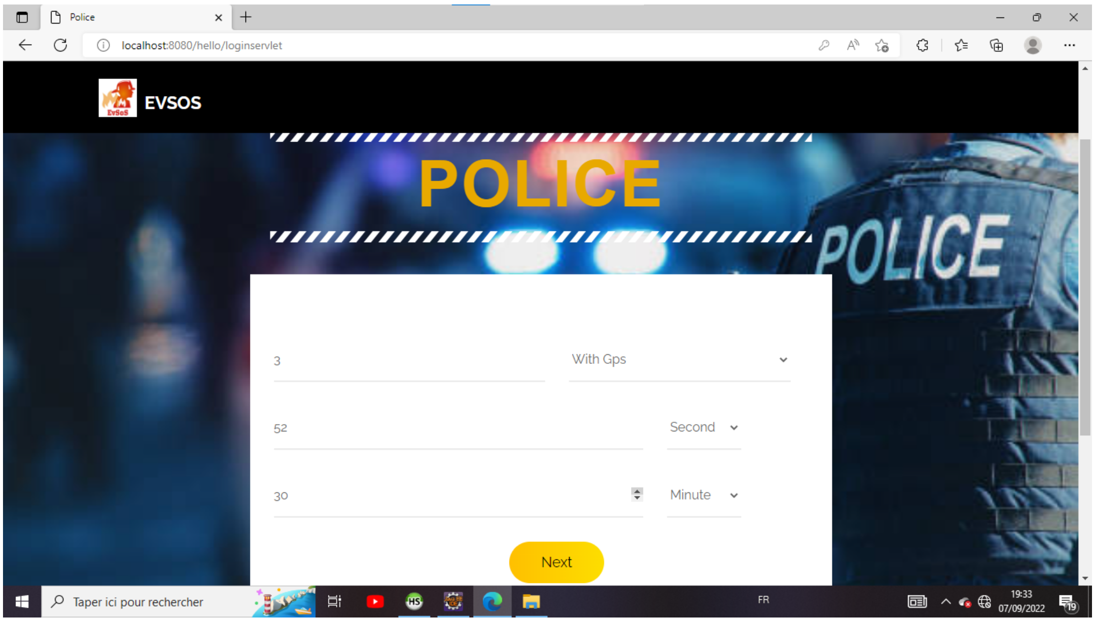

Figure 36: Customer information about police service

13. After the click on the button "Next", Performance of the police service is automatically calculated based on the information entered by the customer using a simple average, as follows :

   - Information entered by the customer will be evaluated(using a standardization). As we can see in figure 37.

```
      // for number of cars
if( numbercars <= 2 ) { numbercarsp = 10;}
if( (numbercars >= 3) & (numbercars <= 5) ) { numbercarsp = 30;}
if (numbercars >= 6)  { numbercarsp = 60;}

     //for gps
if( select1.equals("With Gps") ) { gpsp = 80;}
if( select1.equals("Without Gps") ) { gpsp = 20;}

    //for the first time responce
if((FirsttimeRes<=60) &(select2.equals("second"))   ) {  firsttimep=50;}
if((FirsttimeRes<=5) &(select2.equals("minute"))   ) { firsttimep=30;}
if((FirsttimeRes>5) &(select2.equals("minute"))   ) { firsttimep=20;}

// for the arrival time

if((arrivaltime<=30) &(select3.equals("minute"))   ) { arrivaltimep=50;}
if((arrivaltime>30) &(select3.equals("minute"))   ) { arrivaltimep=30;}
if((arrivaltime<=2) &(select3.equals("hour"))   ) { arrivaltimep=15;}
if((arrivaltime>3) &(select3.equals("hour"))   ) { arrivaltimep=5;}
```

Figure 37: Standardization of police service data

   - We take :
     - vnc : the value of number of cars.

- vgp : the value of gps .
- vftr : the value of first time response.
- vat : the value of arrival time.
- nb : number of elements (in the police service we have 4: vnc, vgp, vftr, vat).

The formula for the simple average will be :

$$MoP_{police} = \frac{vnc+vgp+vftr+vat}{nb}$$

- According to our example, the values will be as follows :
  - $vnc = 30$
  - $vgp = 80$
  - $vftr = 50$
  - $vat = 50$
  - And from it :

$$MoP_{police} = \frac{30+80+50+50}{4}$$
$$MoP_{police} = 52.5\ \%$$

14. In the same time another page appears, in this page the customer must enter the information about ambulance service. Where he/she finds the number of cars, gps, the first time response, and the arrival time, moreover the equipment and paramedics inside an ambulance. As we can see in figures 38, 39.

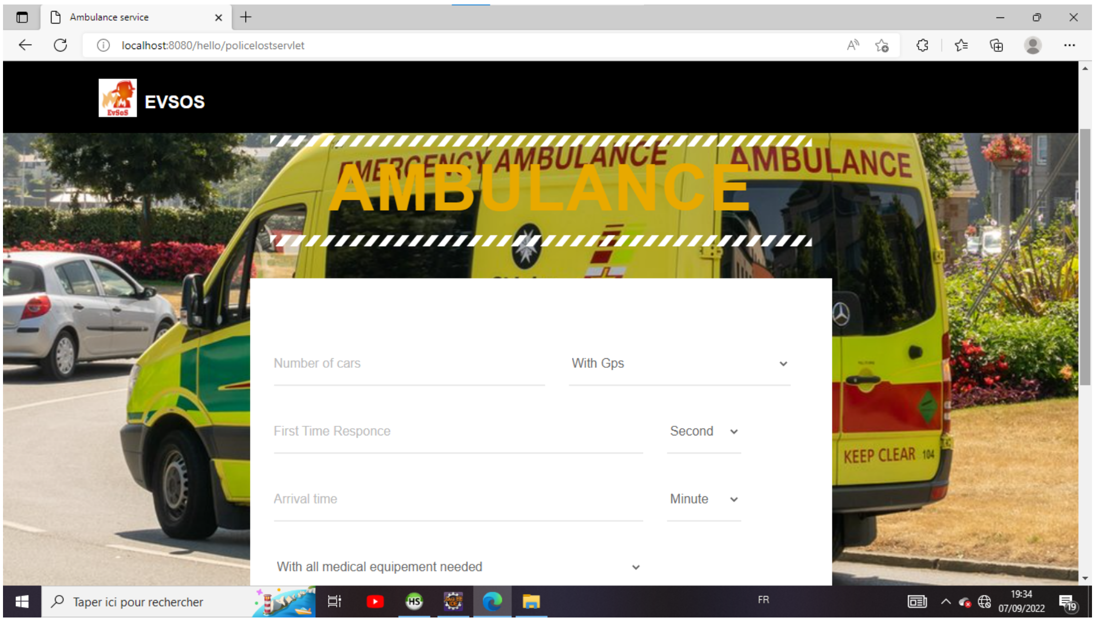

Figure 38: Ambulance service

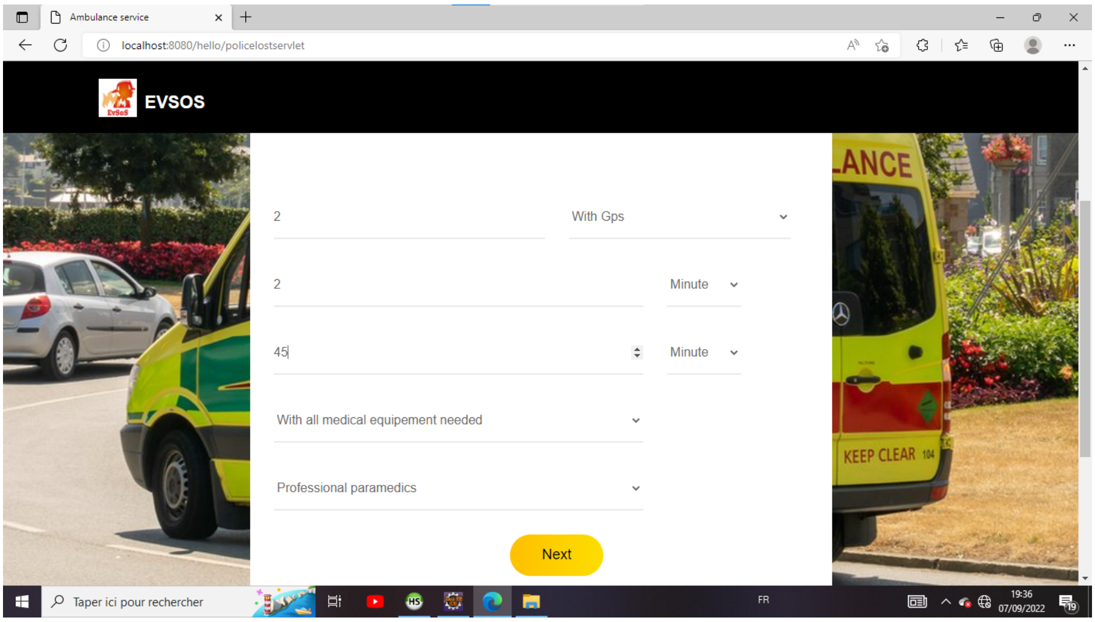

Figure 39: Customer information about ambulance service

15. After the click on the button "Next", Performance of the ambulance service is automatically calculated based on the information entered by the customer.

   - Information entered by the customer will be evaluated(using a standardization). As we can see in figure 40.

```
        // for number of cars
if(numbercars == 1 ) {  numbercarsamblancep = 10;}
if( (numbercars >= 2) & (numbercars <= 4) ) {  numbercarsamblancep = 40;}
if( (numbercars >= 5)  ) {  numbercarsamblancep = 50;}
     //for gps
if( select1.equals("With Gps") ) { gpsp = 80;}
if( select1.equals("Without Gps") ) { gpsp = 20;}
   //for the first time responce
if((firsttime<=60) &(select2.equals("second"))   ) {  firsttimep=50;}
if((firsttime<=5) &(select2.equals("minute"))   ) { firsttimep=30;}
if((firsttime>5) &(select2.equals("minute"))   ) { firsttimep=20;}
// for the arrival time
if((arrivaltime<=30) &(select3.equals("minute"))   ) { arrivaltimep=50;}
if((arrivaltime>30) &(select3.equals("minute"))   ) { arrivaltimep=30;}
if((arrivaltime<=2) &(select3.equals("hour"))   ) { arrivaltimep=15;}
if((arrivaltime>3) &(select3.equals("hour"))   ) { arrivaltimep=5;}
// for the select equipement
if (select4.equals("With all medical equipement needed")) {select4p=70;}
if (select4.equals("With a little of medical equipement")) {select4p=30;}
// for paramedics
if (select5.equals("Professional paramedics")){ select5p=80;}
if (select5.equals("Paramedic with average work experience")) {select5p=20;}
```

Figure 40: Standardization of ambulance service data

   - The MoP of ambulance calculated with the same formula.

$$MoP_{ambulance} = \frac{vnc+vgp+vftr+vat+veq+vpr}{nb}$$

Where :

- veq : the value of equipment.
- vpr : the value of paramedics.

- According to our example, the values will be as follows :
  - $vnc = 40$
  - $vgp = 80$
  - $vftr = 30$
  - $vat = 30$
  - $veq = 70$
  - $vpr = 80$
  - And from it :

$$MoP_{ambulance} = \frac{40+80+30+30+70+80}{6}$$
$$MoP_{ambulance} = 55\ \%$$

16. In the same time another page appears, in this page the customer must enter the information about maritime service. Where he/she finds the number of boats sending with or without gps , the first time response, and the arrival time, moreover the number of drones sending, if the dispatched maritime contain divers or not , if the boats has a smoke and flame signal, VHF radio or not, with or without aviation units. As we can see in figures 41, 42.

- We take the first case(Without aviation units).

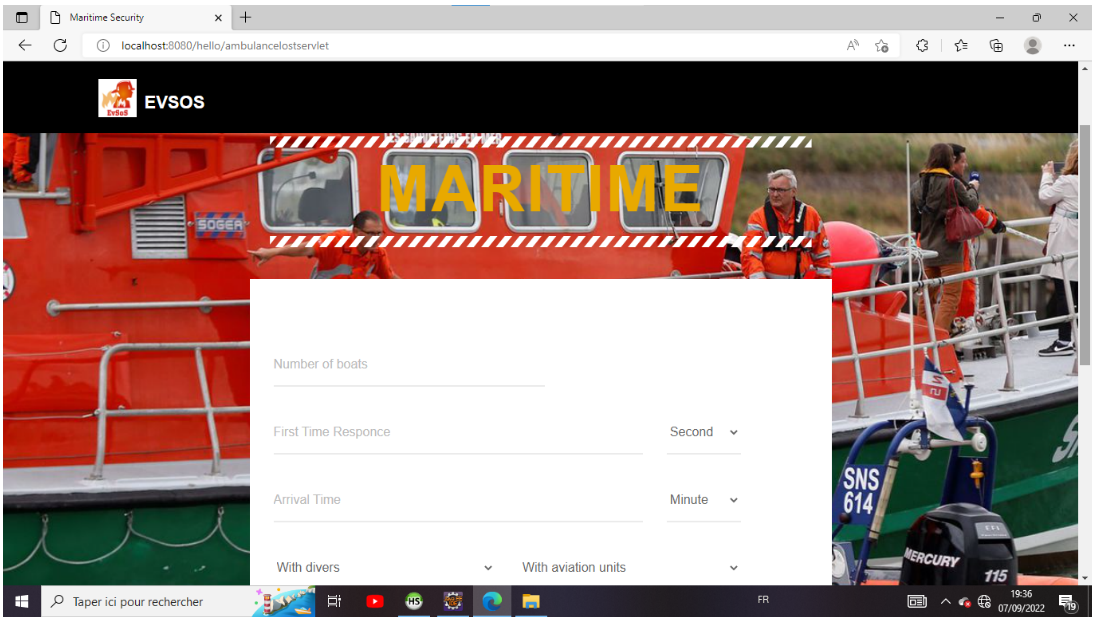

Figure 41: Maritime service

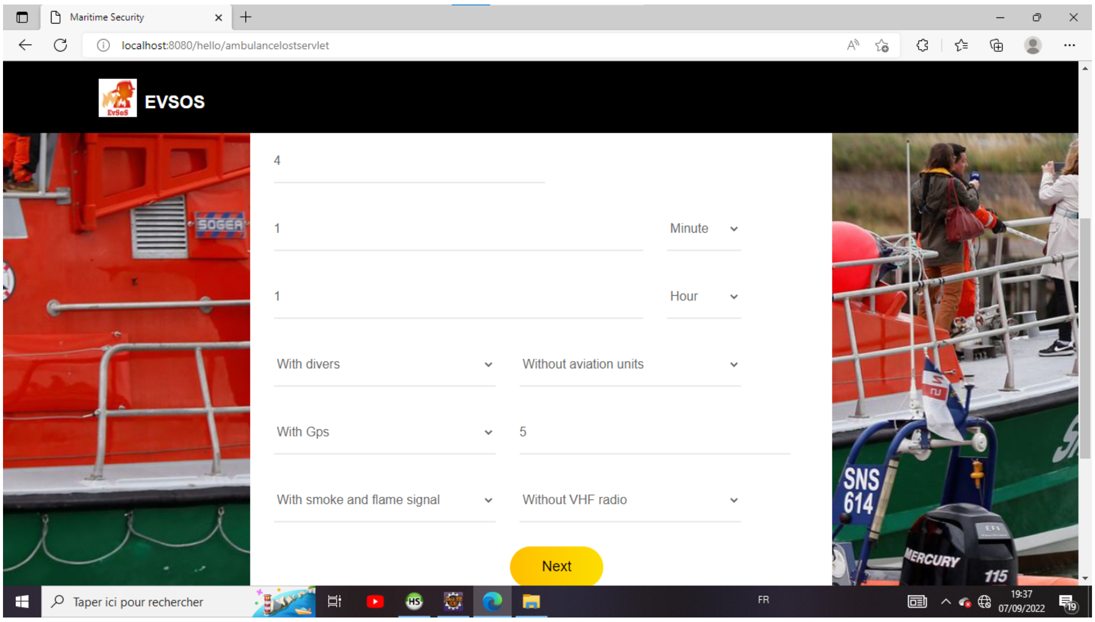

Figure 42: Customer information about maritime service

17. After the click on the button "Next", Performance of maritime service is automatically calculated based on the information entered by the customer.

   - Information entered by the customer will be evaluated(using a standardization). As we can see in figure 43.

```
/*for number of cars */if( (numberboat >= 1) & (numberboat <= 4) ) {  boatsp = 10;}
                       if( ((numberboat >= 5) & (numberboat <= 10) ) ) {  boatsp = 30;}
                       if( (numberboat >= 10)  ) {  boatsp = 60;}
/*for gps*/if( select1.equals("With Gps") ) { gpsp = 80;}
           if( select1.equals("Without Gps") ) { gpsp = 20;}
/*for the first time responce */if((FirsttimeRes<=60) &(select2.equals("second"))   ) {  firsttimep=50;}
                                 if((FirsttimeRes<=5) &(select2.equals("minute"))   ) { firsttimep=30;}
                                 if((FirsttimeRes>5) &(select2.equals("minute"))   ) { firsttimep=20;}
/*for the arrival time*/if((arrivaltime<=30) &(select3.equals("minute"))   ) { arrivaltimep=50;}
                        if((arrivaltime>30) &(select3.equals("minute"))   ) { arrivaltimep=30;}
                        if((arrivaltime<=2) &(select3.equals("hour"))   ) { arrivaltimep=15;}
                        if((arrivaltime>3) &(select3.equals("hour"))   ) { arrivaltimep=5;}
/*for divers*/if( divers.equals("With divers") ) { diversp = 60;}
              if( divers.equals("Without divers") ) { diversp = 40;}
/*for aviation*/if( aviation.equals("With aviation units") ) { avp = 70;}
                if( aviation.equals("Without aviation units") ) {avp = 30;}
/*for number of drone */if( (drone >= 1) & (drone <= 4) ) {  drp = 10;}
                        if( ((drone >= 5) & (drone <= 10) ) ) { drp = 30;}
                        if( (drone >= 10)  ) {  drp = 60;}
/*signal*/if( signal.equals("With smoke and flame signal") ) { sgp = 60;}
          if( signal.equals("Without smoke and flame signal") ) {sgp = 40;}
/*vhf */ if( vhf.equals("With VHF radio") ) { vhfp = 70;}
         if( vhf.equals("Without VHF radio") ) {vhfp = 30;}
```

Figure 43: Standardization of maritime service data

   - The MoP of maritime calculated with the same formula.

$$MoP_{maritime} = \frac{vnb+vgp+vftr+vat+vd+vav+vdr+vs+vvhf}{nb}$$

   Where :

   – vd: the value of divers .

- vav: the value of aviation units.
- vdr: the value of drone.
- vs: the value of signal.
- vvhf: the value of VHF radio.

- According to our example, the values will be as follows :
  - $vnb = 10$
  - $vgp = 80$
  - $vftr = 30$
  - $vat = 15$
  - $vd = 60$
  - $vav = 30$
  - $vdr = 30$
  - $vs = 60$
  - $vvhf = 30$
  - And from it :

$$MoP_{maritime} = \frac{10+80+30+15+60+30+30+60+30}{9}$$
$$MoP_{maritime} = 38.333\,\%$$

18. In the same time a page appears including all information entered by him/her, furthermore the evaluation of his/her SoS (performance of each service or system, and the effectiveness of the SoS). He/She can also print this page. As we can see in Figures 44, 45.

19. The effectiveness is automatically calculated based on the MoP of each service, using a simple average :

$$MoE = \frac{\sum Mop\, of\, each\, service}{number\, of\, services}$$

20. According to our example :

$$MoE = \frac{Mop_{police} + Mop_{ambulance} + Mop_{maritime} + Mop_{aviation}}{4}$$

$Mop_{aviation} = 0$, Because the customer choose without aviation.

$$MoE = \frac{52.5+55+38.333}{4}$$

$$MoE = 36.4583\,\%$$

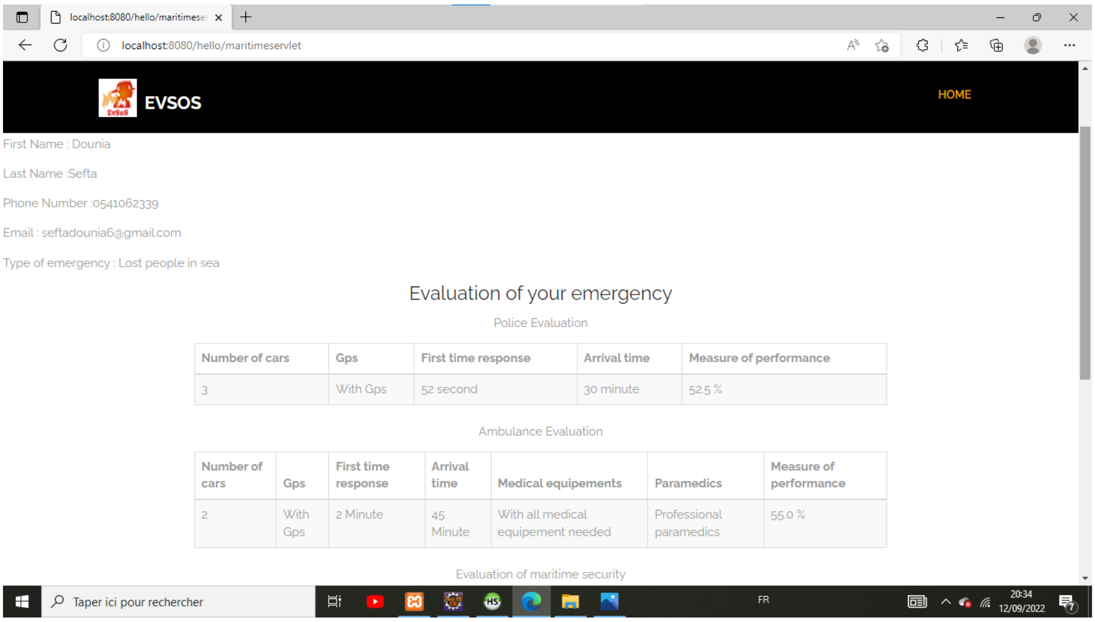

Figure 44: evaluation without aviation units display page (1)

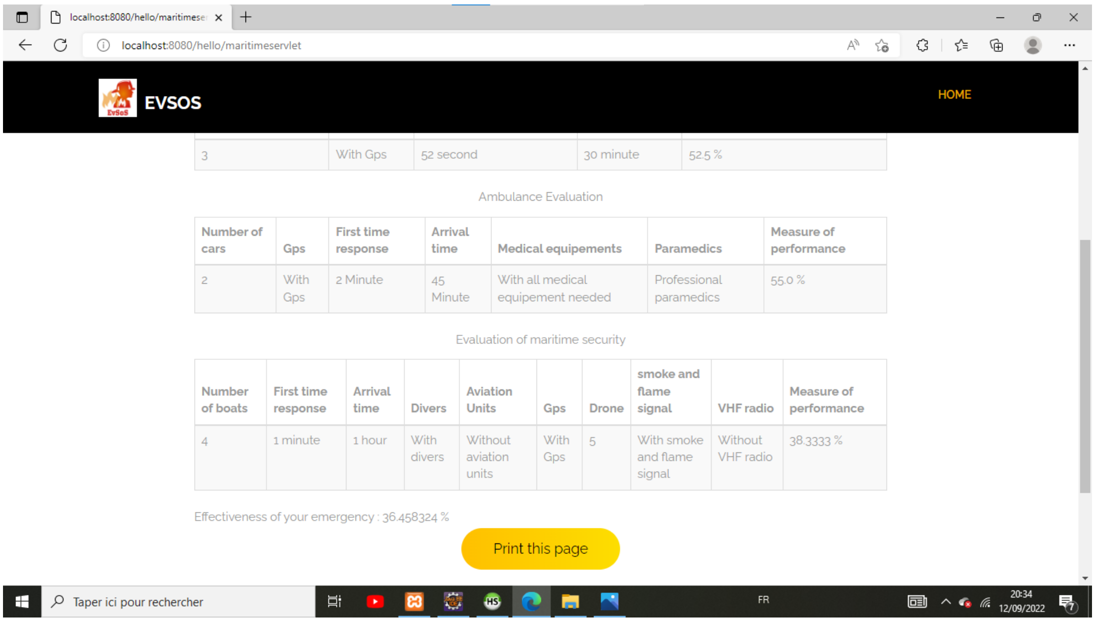

Figure 45: evaluation without aviation units display page (2)

21. If the customer click on "print" , a page appears, as we can see in figure 46, where the customer can print the page of evaluation.

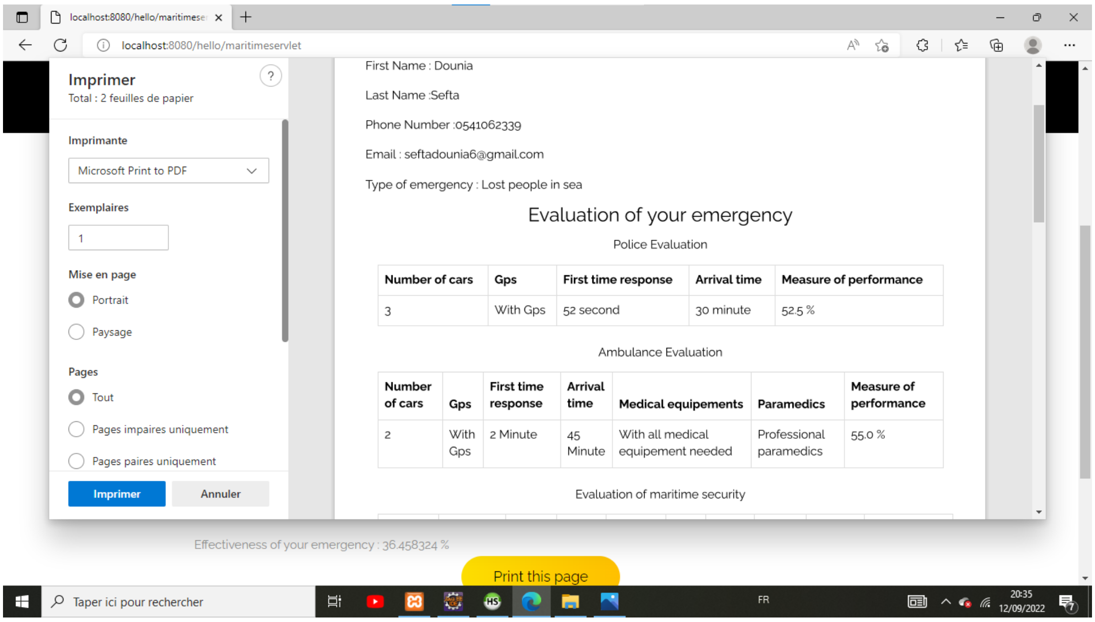

Figure 46: Print page

22. If the customer choose with aviation units in the maritime service then, a page appears, in this page the customer must enter the information about aviation units service, number of aviation sending for the search, first time response from the call center and the arrival time. As we can see in figure 47.

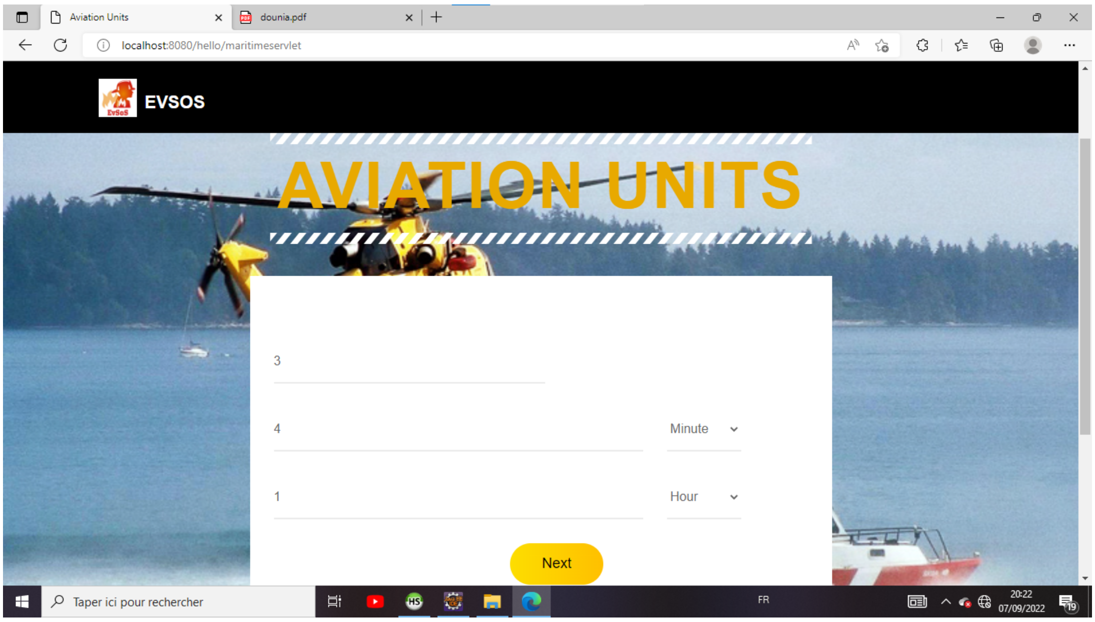

Figure 47: Aviation unit page

23. After the click on the button "Next", Performance of the aviation service is automatically calculated based on the information entered by the customer.

- Information entered by the customer will be evaluated(using a standardization). As we can see in figure 48.

```
        // for number of units
if(numberunits == 1 ) { numberunitp = 10;}
if( (numberunits >= 2) & (numberunits <= 4) ) { numberunitp = 40;}
if( (numberunits >= 5)   ) { numberunitp = 50;}
   //for the first time responce
if((firsttime<=60) &(select2.equals("second"))    ) {  firsttimep=50;}
if((firsttime<=5) &(select2.equals("minute"))    ) { firsttimep=30;}
if((firsttime>5) &(select2.equals("minute"))    ) { firsttimep=20;}
// for the arrival time

if((arrivaltime<=30) &(select3.equals("minute"))    ) { arrivaltimep=50;}
if((arrivaltime>30) &(select3.equals("minute"))    ) { arrivaltimep=30;}
if((arrivaltime<=2) &(select3.equals("hour"))    ) { arrivaltimep=15;}
if((arrivaltime>3) &(select3.equals("hour"))    ) { arrivaltimep=5;}
```

Figure 48: Standardization of aviation service data

- The MoP of aviaton units calculated with the same formula.

$$MoP_{aviation} = \frac{vna+vftr+vart}{3}$$

- According to our example, the values will be as follows :
  - $vnav = 40$
  - $vftr = 30$
  - $vart = 15$
  - And from it :

$$MoP_{aviation} = \frac{40+30+15}{3}$$
$$MoP_{aviation} = 28.333\ \%$$

- The $Mop_{maritime}$ will also change (because the value of aviation change) :

$$MoP_{maritime} = \frac{10+80+30+15+60+30+70+60+30}{9}$$

$$MoP_{maritime} = 42.7778\ \%$$

24. In the same time a page appears including all information entered by the customer, furthermore the evaluation of his/her SoS (performance of each service or system, and the effectiveness of the SoS). He/She can also print this page. As we can see in Figures 49, 50.

25. So the effectiveness in this case will be :

$$MoE = \frac{52.5+55+42.7778+28.333}{4}$$

$$MoE = 44.652\ \%$$

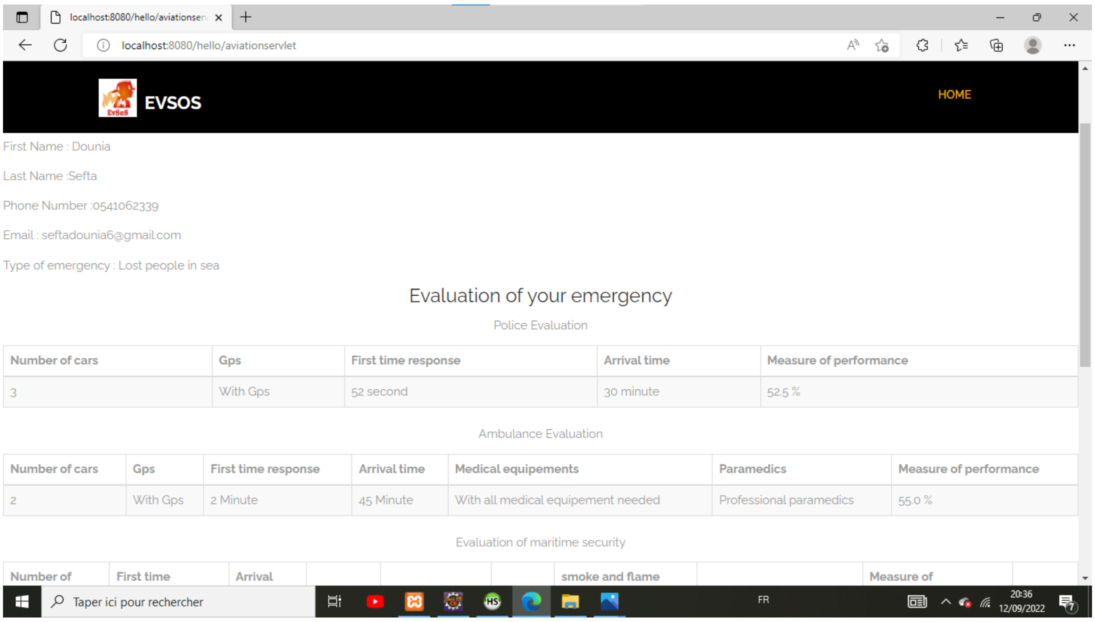

Figure 49: evaluation with aviation units display page (1)

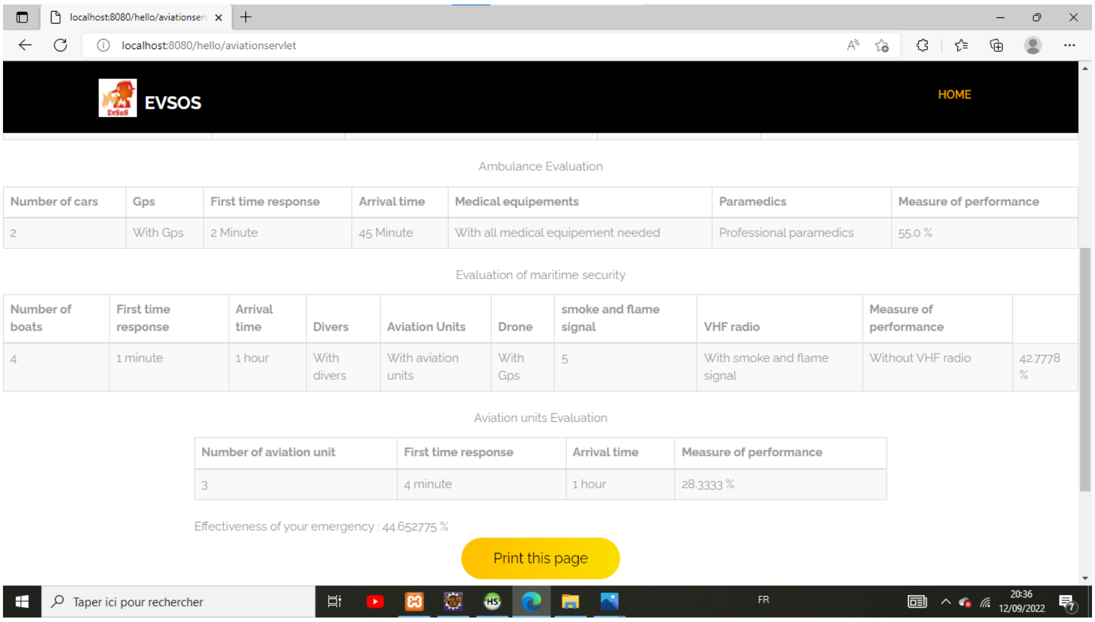

Figure 50: evaluation with aviation units display page (2)

26. If the customer want to discover more evaluation then he/she must click on "home" to come back to the first page, and select "Discover architectures" and the type of emergency wanted. As we can see in figure 51.

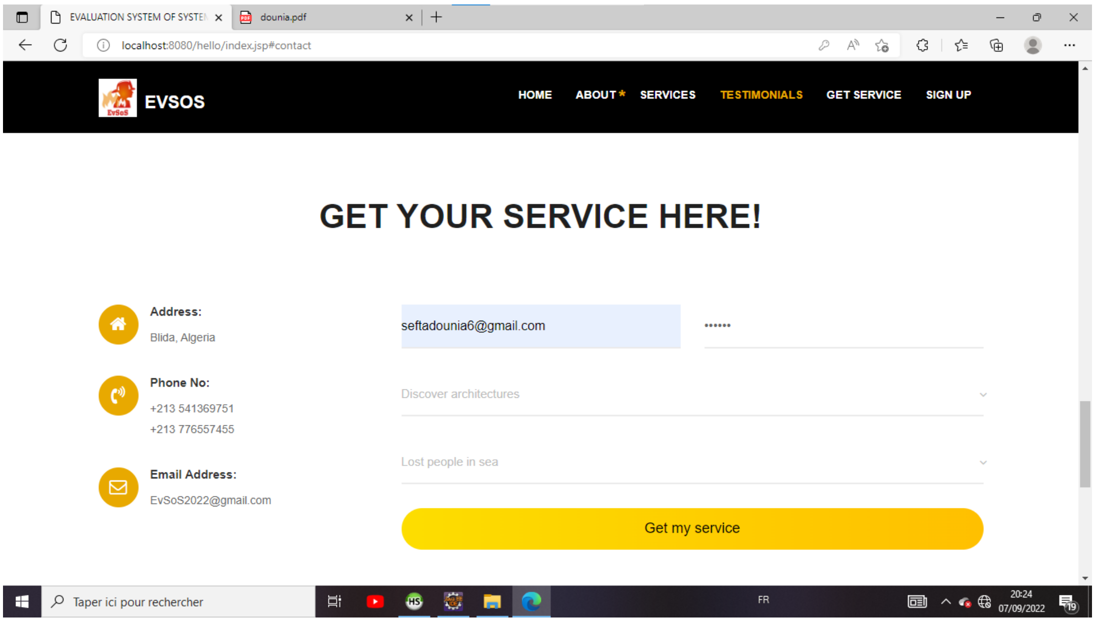

Figure 51: customer select discover architectures

27. When the customer click on "Get my service",a page appears including all the evaluations established by the other customers which has a relation with the type of emergency chosen by him/her. As we can see in figures 52, 53.

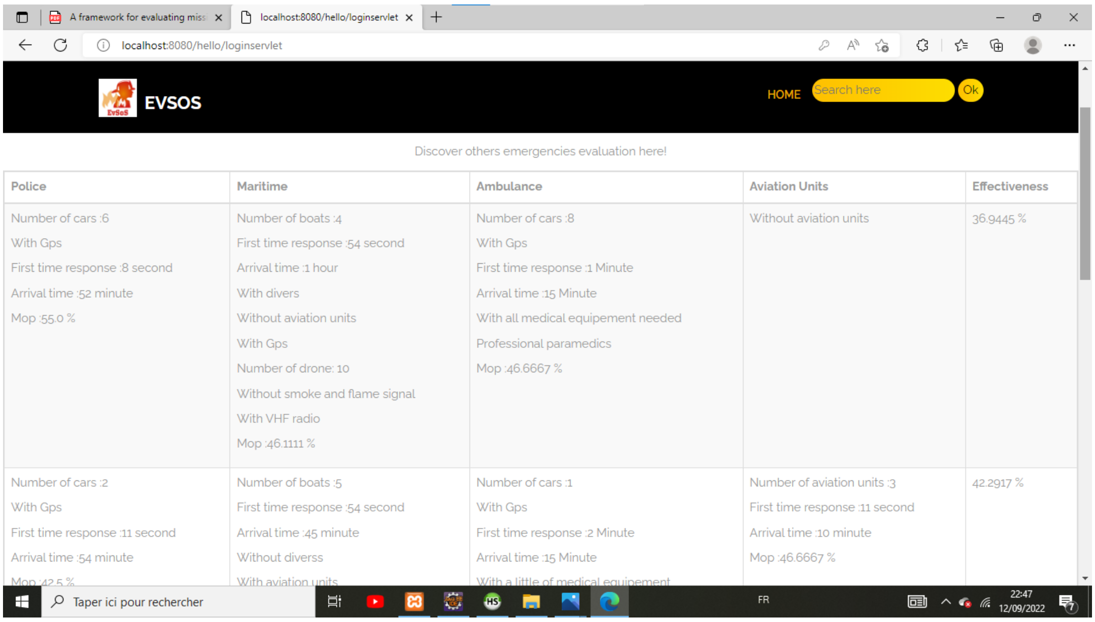

Figure 52: Discover architectures about lost people is sea page (1)

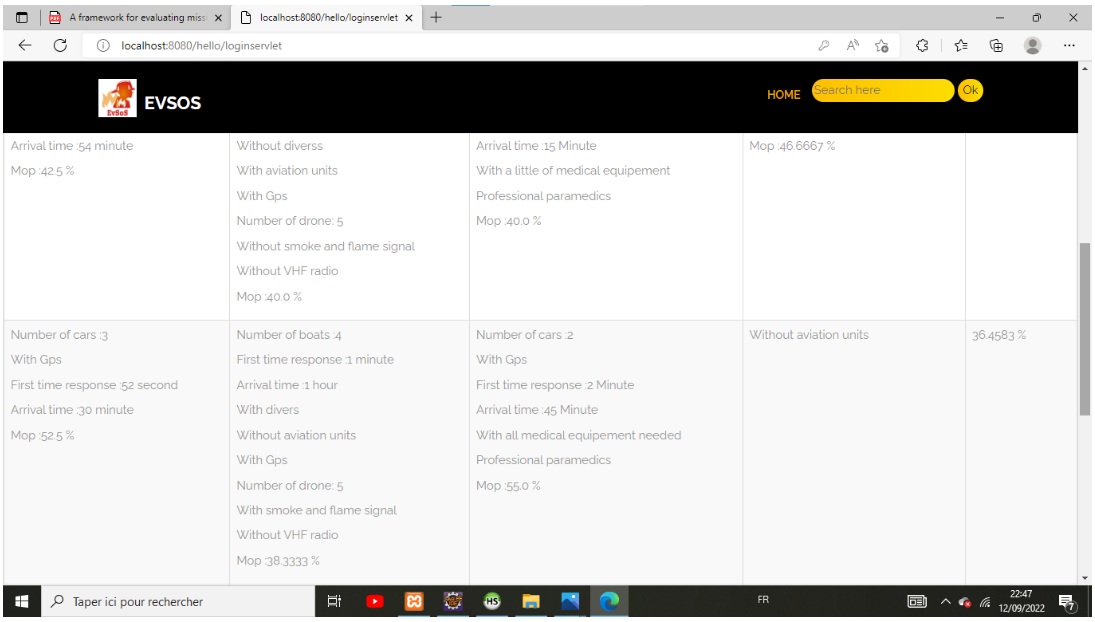

Figure 53: Discover architectures about lost people is sea page (2)

28. The customer can also search what he/she wants in the search bar.

- We take that the customer search for services sending without gps. As we can see in figures 54.

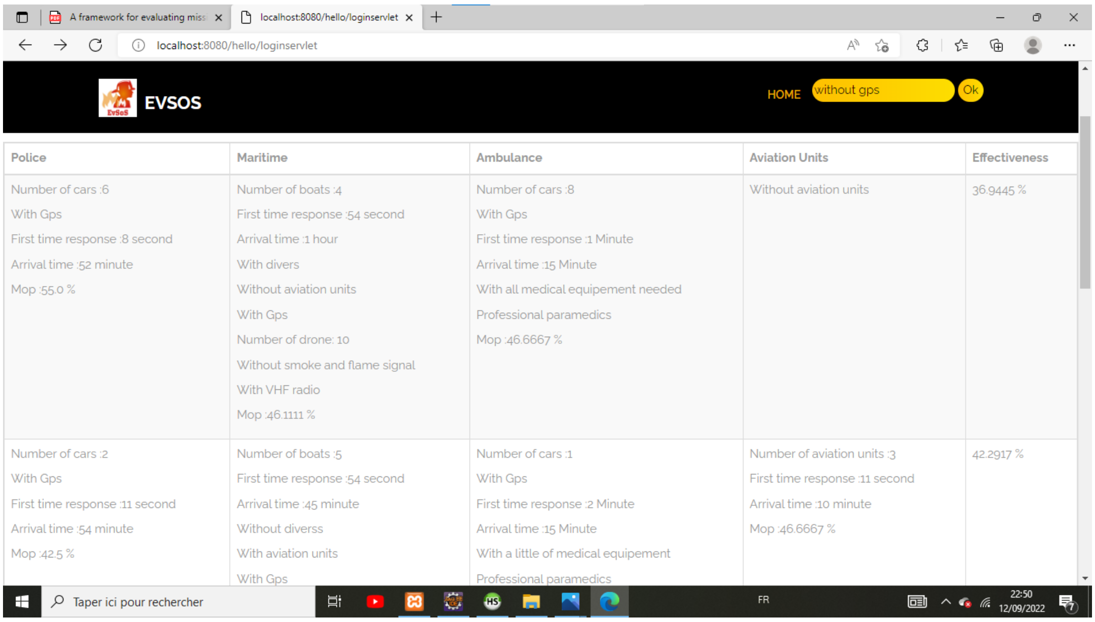

Figure 54: Customer searching page

29. If there is nothing related to the search then a page appears as we can see in figure 55.

   - for our example there is no service sending without gps.

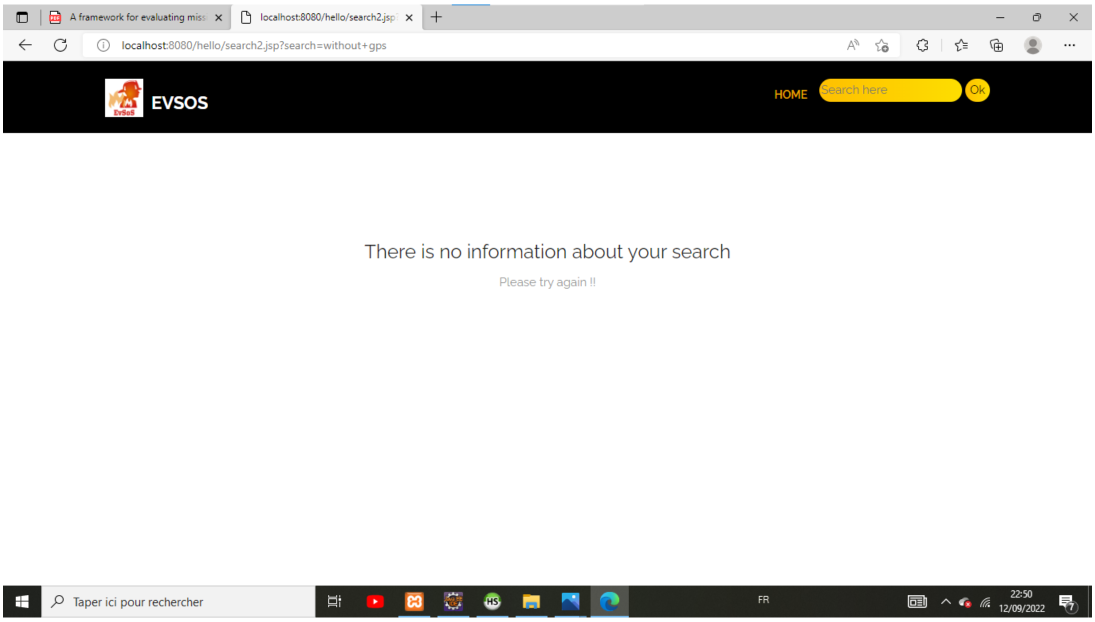

Figure 55: Search page (1)

30. If there is something related to the search, a page appears including all the evaluation related to the search.

   - We take as an example a customer searching for "with a little " as we can see in figure 56 , all the services including "with a little " appears as we can see in figure 57.

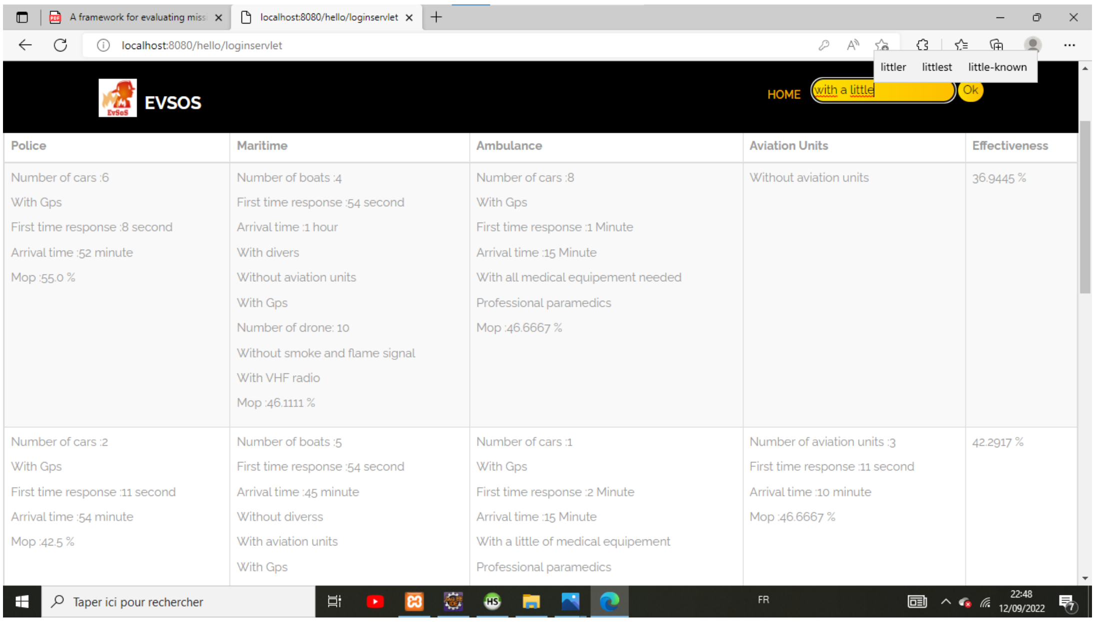

Figure 56: Customer searching page (2)

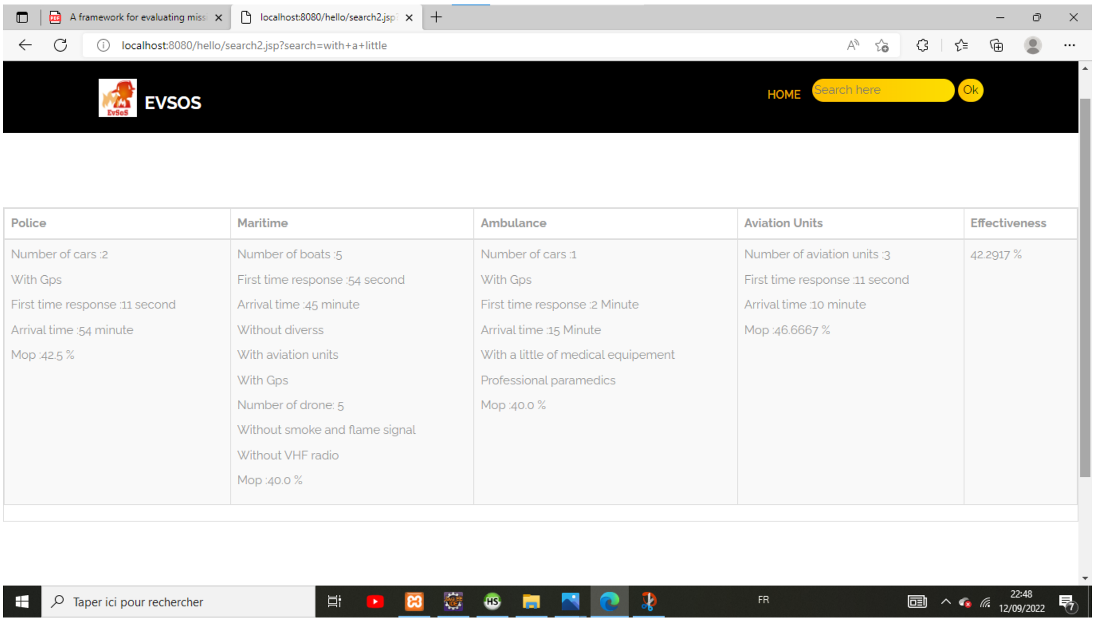

Figure 57: Search page (2)

4.4 Conclusion

In this chapter, we discussed the implementation of our SoS model. We were able to evaluate ERSoS architectures in several scenarios for instance road accidents, people lost at sea, fire, intentional violence and harm. The evaluation process is based on the measure of effectiveness (MoE) and measures of performance (MoPs), where MoEs are based on MoPs. Also, our application gives the user the ability to discover existing evaluated architectures to have inspiration and facilitate taking the best decisions.

CONCLUSION

The main goal of this work is to propose **a framework for evaluating mission-oriented SoS architectures in the emergency response context**.

We proposed a conceptual model and procedures that identify the different evaluation metrics for mission in the SoS context. We used two metrics MoPs in section 2.6.2 and MoEs in section 2.6.1, where MoPs allow us to choose between two constituent systems that perform the same role, while MoEs allow deciding about the effectiveness of the mission. We made our meta-model based on CHERFA's approach, which carries out an emergency response event. Then we created a web application using Eclipse IDE that carried out multiple scenarios to help experts to understand and evaluate their architectures. Our contributions to this work are:

- studying the state of the art in the evaluation of the system of systems and discussing the works interested in the evaluation of the SoS.
- proposition of an extension of the conceptual model proposed by CHERFA[5] using the system modeling language SysML and the tool ECore.
- language SysML and the tool ECore. highlight the structure and the behavior of our SoS using SysML four pillars diagrams (block definition diagram, requirement diagram, activity diagram, and parametric diagram) and the Papyrus tool.
- implementation of the solution by creating a web application to evaluate multiple architectures of the ERSoS.

Limits and future perspectives

In spite of all the contributions cited above, we can point out some limitations of the proposed process for SoS evaluation: the data and scenarios used are not related to a real experience. In addition, the lack of experience and familiarity with the emergency systems makes modeling an emergency scenario a tough mission.

For future improvement, Artificial Intelligence (AI) is an interesting approach to provide a promising way to the prediction of critical properties if joined to modeling and simulation, AI exhibits high efficiency in building automatically predictive models and in improving model performance. It will be interesting to apply AI in the different activities of the SoSE evaluation process.

References

[1] J. H. Brill, “Systems engineering — A retrospective view,” *Systems Engineering*, vol. 1, no. 4, pp. 258–266, 1998.

[2] G. J. Roedler and C. Jones, “Technical measurement INCOSE TP-2003-020-01,” no. December, p. 65, 2005.

[3] D. Webster, N. Looker, D. Russell, L. Liu, and J. Xu, “An Ontology for Evaluation of Network Enabled Capability Architectures,” *Evaluation*, no. April 2015.

[4] Omg and Object Management Group, “OMG Systems Modeling Language (OMG SysML ™) v.1.4,” *Source*, no. June, p. 320, 2015. [Online]. Available: http://www.omg.org/spec/SysML/1.4/PDF/{%}0Ahttp://www.omg.org/spec/SysML/1.2/PDF/

[5] T. D. E. Doctorat, “Faculté des Sciences MISSION ORIENTED PROCESS FOR,” 2021.

[6] S. Liu, “Employing system of systems engineering in China’s emergency management,” *IEEE Systems Journal*, vol. 5, no. 2, pp. 298–308, 2011.

[7] B. S. Blanchard, *SYSTEM ENGINEERING SYSTEM ENGINEERING FIFTH EDITION Virginia Polytechnic Institute and State University*, 2016.

[8] INCOSE, “Definition of System.” [Online]. Available: https://www.incose.org/about-systems-engineering/system-and-se-definition/general-system-definition

[9] A. Gorod, B. Sauser, and J. Boardman, “System-of-systems engineering management: A review of modern history and a path forward,” *IEEE Systems Journal*, vol. 2, no. 4, pp. 484–499, 2008.

[10] INCOSE, “Systems of Systems Primer.” [Online]. Available: https://www.incose.org/products-and-publications/sos-primer{#}:{~}:text=ASystemofSystems(SoS,atopicofincreasinginterest.

[11] M. W. Maier, “Architecting Principles for,” *The Aerospace Corporation; John Wiley & Sons, Inc. Syst Eng*, vol. 1, pp. 267–284, 1999.

[12] J. Boardman and B. Sauser, “System of Systems - The meaning of of,” *Proceedings 2006 IEEE/SMC International Conference on System of Systems Engineering*, vol. 2006, no. April, pp. 118–123, 2006.

[13] R. Abbott, “Open at the top; open at the bottom; and continually (but slowly) evolving,” *Proceedings 2006 IEEE/SMC International Conference on System of Systems Engineering*, vol. 2006, no. April, pp. 41–46, 2006.

[14] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, “Systems of systems engineering: Basic concepts, model-based techniques, and research directions,” *ACM Computing Surveys*, vol. 48, no. 2, pp. 1–41, 2015.

[15] W. C. Baldwin, B. Sauser, and R. Cloutier, “Simulation approaches for system of systems: Events-based versus agent based modeling,” *Procedia Computer Science*, vol. 44, no. C, pp. 363–372, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.03.032

[16] A. Sousa-Poza, S. Kovacic, and C. Keating, "System of systems engineering: An emerging multidiscipline," *International Journal of System of Systems Engineering*, vol. 1, no. 1-2, pp. 1–17, 2008.

[17] D. J. Dahmann, "INCOSE Systems of Systems Working Group Alan Harding." [Online]. Available: https://slidetodoc.com/incose-systems-of-systems-working-group-alan-harding/

[18] D. Soares Santos, B. Oliveira, M. Guessi, F. Oquendo, M. Delamaro, and E. Yumi Nakagawa, "Towards the evaluation of System-of-Systems software architectures," *Congresso Brasileiro de Software: Teoria e Pratica*, no. ii, pp. 53–57, 2014.

[19] Office of the Chairman of the Joint Chiefs of Staff, "DOD dictionary of military and associated terms," *Joint Education and Doctrine Division, J-7*, no. January, p. 382, 2021. [Online]. Available: https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/dictionary.pdf

[20] C. S. Wasson, "System Mission Analysis," *System Analysis, Design, and Development*, pp. 159–166, 2005.

[21] R. D. GARRETT. and R. K., "A Novel Approach to Mission-Level Engineering of Complex Systems of Systems: Addressing Integration and Interoperability Shortfalls by Interrogating the Interstitials. Technical Report 13-MDA-7269, Missile Defense Agency, De- cember 2013." vol. 74, no. 1934, pp. 535–546, 1941.

[22] C. S. Wasson, "System Mission Analysis," *System Analysis, Design, and Development*, pp. 159–166, 2005.

[23] T. Holland, "vdocuments," *Measures, Metrics, and Systems-of-Systems.*

[24] J. YOUNG, "Metrics," 2022. [Online]. Available: https://www.investopedia.com/terms/m/metrics.asp

[25] D. Kaslow, B. Ayres, P. T. Cahill, and L. Hart, "A model-based systems engineering approach for technical measurement with application to a CubeSat," *IEEE Aerospace Conference Proceedings*, vol. 2018-March, pp. 1–10, 2018.

[26] E. Silva, T. Batista, and F. Oquendo, "On the verification of mission-related properties in software-intensive systems-of-systems architectural design," *Science of Computer Programming*, vol. 192, p. 102425, 2020. [Online]. Available: https://doi.org/10.1016/j.scico.2020.102425

[27] E. Silva and T. Batista, "Formal modeling systems-of-systems missions with mKAOS," *Proceedings of the ACM Symposium on Applied Computing*, pp. 1674–1679, 2018.

[28] J. Quilbeuf, E. Cavalcante, L. M. Traonouez, F. Oquendo, T. Batista, and A. Legay, "A logic for the statistical model checking of dynamic software architectures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9952 LNCS, pp. 806–820, 2016.

[29] MITRE, "Test and evaluation of systems of systems," no. Table 1, pp. 1–10, 2021. [Online]. Available: https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/other-se-lifecycle-building-blocks-articles/test-and-evaluation-of-systems-of-systems{#}

[30] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML, Second Edition: The Systems Modeling Language*, 2011. [Online]. Available: http://www.amazon.com/dp/0123852064

[31] L. Delligatti, "SysML Distilled," p. 301, 2014.