UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Requirements analysis for technical interoperability driven by mission engineering

Cristiane Aparecida Lana

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: ______________________

Cristiane Aparecida Lana

Requirements analysis for technical interoperability driven by mission engineering

Thesis submitted to the Institute of Mathematics and Computer Science – ICMC-USP,in accordance with the requirements of the Doctorate of Science – Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Elisa Yumi Nakagawa

USP – São Carlos
June 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

L243r

Lana, Cristiane Aparecida
Requirements analysis for technical interoperability driven by mission engineering / Cristiane Aparecida Lana; orientador Elisa Yumi Nakagawa. -- São Carlos, 2020.
275 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2020.

1. SoS Mission. 2. Interoperability. 3. Architectural Requirements Analysis. 4. Grounded Theory. 5. Mission Engineering. I. Nakagawa, Elisa Yumi, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Cristiane Aparecida Lana

Análise de requisitos para interoperabilidade técnica dirigida por engenharia de missão

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Elisa Yumi Nakagawa

USP – São Carlos
Junho de 2020

To my mother Maria do Carmo, my father Antônio Lana and my little son Leo Lana (in memorian)

Acknowledgements

I have reinvented myself from each attempt to find answers to my afflictions of both a person and a researcher. For those who have shared such moments with me, it seemed an endless and enigmatic task that has become possible only because of people who directly or indirectly participated in the process, although not really knowing the reason for such involvement. No doubt they deserve my due acknowledgement.

To God and my guardian, for having provided me with wisdom and guided my path and each day of my journey.

To my parents, Maria do Carmo and Antonio (in memorian), for giving me life and love - my mother, for always being supportive to my dreams! I share with you the success of this work.

To my brothers, Kleide and Cleber, for always being available to me, and Fernando for the encouragements in all moments of my life. To Victor and Torvictor, who have been light in my life, and to my sisters-in-law, Léa and Lúciola, for always welcoming me and taking care of my mother.

To my nephews and nieces, Samyne, Bernardo, Rafaela, Caio, Max, Felipe, and Yasmim Cristina, for bringing light and joy in difficult days. Aunt loves you!

To my family Silva, who is an example of union and love that I want to take for life.

To my friends Mariana, Ana Beatriz, and Renne, my gratitude for everything and all encouragement given. You are my second family and I want to be always with you.

To my friends Gioana Fargnoli, Áurea, Cidoca, Ritinha, Rúbia, Margareth, Vanderléa, Medianeira, Anderson, Angelica, Cristina, Laís, Isa Muniz, Jaqueline, Chico, Giorgio, Lucas, Gilberto, Drica, Berlizete, Isa, Karma, Carlos, Lili, Ana Chrystina, Milena, Alex, Milton, Sueli, Manuel, and many others (rsrs). You are rich gifts I have been given and that will remain!

To my advisor Prof. Dr. Elisa Yumi Nakagawa, my gratitude for sharing her wisdom with me and for her valuable advice. Thanks for the conversations and laughters, they have made my life and journey lighter. You were essential for my becoming what I am. To my internship supervisor, Prof. Dr. Dieter Rombach, for having me in his research group at Fraunhofer Institute for Experimental Software Engineering (Fraunhofer IESE) and University of Kaiserslautern and sharing his expertise.

To Prof. Dr. José Carlos Maldonado, thanks for all the support and advice throughout these years, and also for sharing your wisdom, industrial experience, and cases with me. You were very important to my evolution.

To my professors, fellow researchers, students and staff from University of São Paulo, in particular to Lucas, Diego, Kamila, Anderson, Danilo, Pedro, Katia, Aline, Carlos, Dio, Silvana, Estevão, Anderson, Aninha, and João Paulo, whose friendship has been a bonus and a joy in my journey, and to Bianch, Milena, Brauner, Frank, and Daniel, who, besides a bonus, have been supportive in my life.

To researchers, colleagues, staff from the Fraunhofer Institute for Experimental Software Engineering (Fraunhofer IESE) and University of Kaiserslautern, in particular Andreas, Raquel, Sonnhild, Rasmus, Mathias Jung, Rodrigo, Karina, and Adeline for all support and sharing of knowledge, and to Dr. Pablo Oliveira Antonino, for receiving me in Franhoufer IESE and sharing his expertise with me.

To my friends Nuquangdala, Aline, Eduardo, and Alexandre Venito, whose friendship is a new gratuity for my life.

My gratitude to all researchers, students, and staff that somehow collaborated with this thesis. In particular, I would like to thank Prof. Dr. Juliana Cobre (ICMC), Prof. Dr. Mariana Cúri (ICMC), Prof. Dr. Luis Gustavo Nonato (ICMC), Prof. Dr. Rafael Capilla (URJC), Prof. Dr. Carlos Eduardo de Freitas (ESALQ/USP), Prof. Fabio Leite (UEPB), Prof. Dr. Kalinka Regina Lucas Jaquie Castelo Branco (ICMC), Prof. Dr. Leandro Fiorini Aurichi, CEMEAI, Rafael (STI) and STI, Dr. Ricardo Yassushi (EMBAPA Instrumentação), MSc Wilson Milani Zambianco (ESALQ/USP), José Jorge Correa (Sygenta), Dr. Andreas Tolk (Mitre), Dr. Saikou Diallo (Mitre), BSc Giovanna Zolin Hayasida (ICMC), Angela Cristina Pregnolato Giampedro (Centro Cultural - USP) for their contributions in several stages of this research.

To my friends Solange Resende, Naiara Resende, Cristal, and Anandsing Dwarkasing, thank you so much for everything and all life teachings and the opportunity for constant growth in the constellation group. You are amazing.

To my father, counselor, and friend José Luiz (Zezito) and his wife Graça; it has been 10 years since our first orientation meeting, and today you are my great friends. Gratitude for all your encouragement for my flying higher and higher, and thank you for the example of the professional you are and for everything you have taught me.

To my adoptive father Manuel (in memorian), who was my inspiration. Gratitude for supporting me since my undergraduate studies and for making me never give up of my dreams and always look at the others with love and benevolence. You always believed in my abilities, and I will take all your teachings with me.

To the São Paulo Research Foundation (FAPESP) and Coordination for the Improvement of Higher Education Personnel (CAPES), for the financial support to this research (grant N.2015/06195-3 and grant N.2017/15354-3) and the CAPES (grant N.PROEX-9259572/D).

"The capacity to learn is a gift; The ability to learn is a skill; The willingness to learn is a choice".

(Brian Herbert)

Abstract

LANA APARECIDA, C. **Requirements analysis for technical interoperability driven by mission engineering**. 2020. 275 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Context: Systems-of-Systems (SoS) have drawn the interest of both industry and academia owing to the continuous growth of business applications. Constituent systems (CSs) can redirect their resources and capabilities towards achieving the SoS mission with new and enhanced functionalities, which would not be feasible by a single CS working independently. The mission is at the core of SoS design, which build upon the reliable analysis of technical interoperability to obtain cohesive SoS bevavior. SoS exhibit emergent behavior, resultant from the interoperation of their CSs, leading to emergence of needs, capabilities, and requirements at runtime. **Aims:** this research aims at the development of a substantive theory supporting the analysis of technical interoperability requirements drive by mission for the design and development of SoS. **Method:** The research was guided by the grounded theory tailored for the construction of substantive theories based on empirical data systematically analyzed and compared with the literature. The data were collected through of practitioners and researchers developing industrial projects in Brazil, USA, Germany, and Spain, from November/2017 to November/2019, using a survey, interviews. Data were analyzed with the content analysis method by the Atlas/Ti 8.0. **Results:** We analyzed 51 survey responses and 17 interviews for designing a new theoretical framework, named ATLANTA. It is comprised eleven categories and five sets of knowledge that may assist the analyses of technical interoperability requirements bounded time and space, and of the business value towards monitoring changes and perspectives for future evolutionary procedures of the SoS development. ATLANTA covers and does not limit to (i) the SoS modeling and design through of the Triplet model, which integrate mission, requirements, and architectural requirements specification, (ii) the situational analysis for SoS enabling identification and analysis of mission based on OKR, and (iii) the specification of technical interoperability requirements derivative from mission adopting specific transformations properties and quality models at design time and runtime. **Conclusions:** ATLANTA promotes a better understanding of the technical interoperability requirements influence on the SoS mission behavior and shows the importance of enterprise analysis to the SoS level during its lifecycle at design time and runtime. The particular focus on practitioners bridges the state-of-the-art and state-of-the-practice. Mission engineering crosscuts analysis, design, and development of SoS, reflecting on the emergent behavior at runtime. Thereby, this theory is intended to support researchers and practitioners in analysis and maintaining technical interoperability requirements and definition/specification/modeling of mission SoS, through a strategical and agile approach of the industrial environment over time.

Keywords: SoS Mission, Interoperability, Architectural Requirements Analysis, Grounded Theory, Mission Engineering.

Resumo

LANA APARECIDA, C. **Análise de requisitos para interoperabilidade técnica dirigida por engenharia de missão**. 2020. 275 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Contexto: Os Sistemas de Sistemas (SoS) tem atraído o interesse da indústria e da academia devido ao crescimento contínuo dos aplicativos de negócios. Os sistemas constituintes (CSs) podem redirecionar seus recursos e capacidades para alcançar a missão SoS com funcionalidades novas e aprimoradas, o que não seria viável por um único CS trabalhando independentemente. A missão é o núcleo do design de SoS, que se baseia na análise confiável da interoperabilidade técnica para obter um comportamento coeso do SoS. SoS exibem comportamento emergente, resultante da interoperação de seus CSs o que leva ao surgimento de novas necessidades, capacidades e requisitos em tempo de execução. **Objetivo:** Esta pesquisa visa o desenvolvimento de uma teoria substantiva que apóie a análise de requisitos de interoperabilidade técnica dirigido por missão para o design e desenvolvimento de SoS. **Método:** A pesquisa foi conduzida pela teoria fundamentada, adaptada para a construção de teorias substantivas baseadas em dados empíricos sistematicamente analisados e comparados com a literatura. Os dados foram coletados por meio de profissionais e pesquisadores que desenvolvem projetos industriais no Brasil, EUA, Alemanha e Espanha, de Novembro/2017 a Novembro/2019, utilizando survey e entrevistas. Os dados foram analisados pelo método de análise de conteúdo pelo Atlas/Ti 8.0. **Resultados:** Analisamos 51 respostas da survey e 17 entrevistas para design um novo referencial teórico, denominado ATLANTA. Ele é composto por onze categorias e cinco conjuntos de conhecimentos que podem auxiliar na análise de requisitos de interoperabilidade técnica limitados de tempo e espaço, e do valor comercial para monitorar mudanças e perspectivas para futuros procedimentos evolutivos do desenvolvimento de SoS. O ATLANTA cobre e não se limita a (i) modelagem e design de SoS por meio do modelo Triplet, que integra missão, requisitos e especificação de requisitos arquiteturais, (ii) a análise situacional para o SoS, permitindo a identificação e análise da missão com base no OKR, e (iii) a especificação de requisitos de interoperabilidade técnica derivados da missão, adotando propriedades de transformação específicas e modelos de qualidade em tempo de design e tempo de execução. **Conclusões:** O ATLANTA promove uma melhor compreensão da influência dos requisitos de interoperabilidade técnica no comportamento da missão de SoS e mostra a importância da análise corporativa para o nível de SoS durante seu ciclo de vida no tempo de design e tempo de execução. O foco particular nos profissionais conecta o estado da arte e o estado da prática. A engenharia de missão é paralela a análise, o design e o desenvolvimento de SoS, refletindo sobre o comportamento emergente em tempo de execução. Assim, essa teoria visa apoiar pesquisadores e profissionais na análise e manutenção de requisitos de interoperabilidade técnica e definição/especificação/modelagem de SoS de missão, por meio de

uma abordagem estratégica e ágil do ambiente industrial ao longo do tempo.

Palavras-chave: Missão de Missão, Interoperabilidade, Análise de Requisitos Arquitetural, Engenharia de Missão.

List of Figures

List of Tables

List of abbreviations and acronyms

ACRE	Approach for Context-based RE
ADL	Architecture Description Languages
ASR	Architectural Significant Requirements
ATAM	Architecture Trade-off Analysis Method
ATLANTA	De**A**ling wi**T**h Mu**L**tilevel of Technic**A**l I**N**teroperabili**T**y **A**wareness
BPMN	Business Process Model and Notation
C4IF	Connection, Communication, Consolidation, Collaboration Interoperability Framework
CMMI-DEV	Capability Maturity Model Integration for Development
CMU-SEI	Carnegie-Mellon University Software Engineering Institute
ConMisOsp	Concept of SoS Mission Operations
ConOps	Concept of Operation
CSs	constituent systems
DoD	US Department of Defense
DoDAF	Department of Defense Architecture Framework
EA	Enterprise Analysis
EARS	Easy Approach to Requirements Syntax
f-DL	fuzzy Description Logic
f-UML	fuzzy UML
FOL	First Logical Order
FSM	Finite State Machine
GORE	Goal-oriented requirements engineering
i-Score	Layered Interoperability Score
IAM	Interoperability Assessment Methodology
ICT	Information and Communication Technology
IRD	Interelement Requirements Diagram
ISIMM	Information Systems Interoperability Maturity Model
IT	Information Technology
KAOS	Knowledge Acquisition in Automated Specification
KPI	Key Performance Indicator
LaRC	Langley Research Center

LCI	Layers of Coalition Interoperability
LCIM	Levels of Conceptual Interoperability Model
LISI	Levels of Information System Interoperability Model
MA	Mission Architecture
MACCS	Marine Air Command and Control System
MAL	Modal Action Logic
MBO	Management by Objective
MBRE	Model-Based Requirements Engineering
MCISI	Military Communications and Information Systems Interoperability
MDA	Model Driven Architecture
MisCon	Mission Concept
MO	mission objectives
NEA	Núcleo de Estatística Aplicada
NFR	Non-functional Requirements
NMI	NATO C3 Technical Architecture Reference Model for Interoperability
NTI	Non-Technical Interoperability Framework
OIAM	Organizational Interoperability Agility Model
OIM	Organizational Interoperability Maturity Model for C2
OpsCon	Operational Concept
OSI	Open System Interconnection
OT	Operational Technology
PVS	Prototype Verification System
QoIM	Quantification of Interoperability Methodology
RE	Requirements Engineering
RML	Requirements Modeling Language
RSD	Requirements Specification Document
SA	Systems Architecture
SAAM	Scenario-Based Architecture Analysis Method
SAI	SoS Architecting Ilities
SDI	Strategic Defense Initiative
SDL	Specification and Description Language
SEI	Software Engineering Institute
SLA	Service Level Agreement
SoIM	Spectrum of Interoperability Model
SoS	Systems-of-Systems
SoSE	Systems-of-Systems Engineering
SoSI	System-of-Systems Interoperability Model

SyRE	Systems Specification Document
SysML	Systems Modeling Language
TA-FCM	Timed-Automata-based Fuzzy Cognitive Maps
TCP-IP	Transmission Control Protocol-Internet Protocol
UFV	Federal University of Viçosa
UML	Unified Modeling Language
UPDM	Unified Profile for DODAF and MODAF
V&V	Verification and Validation
VDM	Vienna Development Method

List of symbols

StK — Stakeholders

HLN — High-Level Needs

M — Mission

MO — Mission Objective

PM — Partial Mission

IM — Individual Mission

PM — Parent Mission

DM — Descendant Mission

C — Capability

CO — Capability Objective

CM — Parent Capability

DC — Descendant Capability

R — Requirements

RD — Descendants Requirements

PR — Parent Requirements

CSs — Constituents Systems

CS — Constituent System

CSC — Constituent System Capability

SoS — System-of-Systems

EB — Emergent Behavior

EDP — Evolutionary Development Property

RC — Resources

NM — New Mission

NC — New Capability

NR — New requirements

ND — New Needs

T — Tasks

I — Intention

S — Service

Contents

CHAPTER 1

Introduction

1.1 Context and Motivation

THE accelerated development of both industry and society has led several independent software-intensive systems to be able to exchange information and interoperate with each other, thus resulting in more complex systems, called Systems-of-Systems (SoS)[1]) (MAIER, 1998; DAHMANN; BALDWIN, 2008; MAIER, 2014; DERSIN; TRANSPORT, 2014; WANG *et al.*, 2017). Such constituent systems (CSs)[2]) are often developed by different companies and rely on different platforms and technologies. The combined work of the CSs enables the SoS to perform complex functions that cannot be delivered otherwise (MAIER, 1998; HAN; DELAURENTIS, 2013), and their two marked characteristics, called evolutionary development and emergent behavior, distinguish them from other systems (MAIER, 1998; NCUBE *et al.*, 2013; DERSIN; TRANSPORT, 2014; NIELSEN *et al.*, 2015a). *Evolutionary development* refers to the capacity of an SoS to evolve in response to changes in its environment, CSs, or missions[3], functions, and purposes[4]. For instance, an SoS must adapt to changes in the constituents, which may affect its mission to resume its proper functioning. *Emergent behavior* refers to new functions that cannot be performed by a CS separately. Such behaviors (or functionalities) can be displayed only by interactions among constituents over time (KEATING *et al.*, 2015). Emergent behaviors may originate in constituents and trigger new ones at the SoS level, and vice-versa. Examples of SoS can be found in several application domains, such as military, aerospace, transportation, industry 4.0, health care, agro4.0, and Internet of Things has become a trend for

[1] For this Thesis, acronym SoS is used interchangeably to express singular and plural.
[2] For this Thesis, acronym CS is used to express singular and CSs plural.
[3] It expresses the set of services the SoS, must deliver, i.e., the main or global transformation performed by the SoS, together with purpose, which clearly reveal the action to be taken and the rationale (FAISANDIER, 2015; LEGGETT, 2017)
[4] It expresses the relevance of the SoS in its context of use and presents the final aim of the SoS in its environment (FAISANDIER, 2015)

future systems.

The mission concept was first cited in the 1970s, in the military domain on the Apollo program of NASA, which took humans to the moon. Since then, it has been broadly adopted in other SoS (ESARY; ZIEHMS, 1975). Currently, the mission engineering concept has emerged to also support the SoS design, which is a broader process focused on satisfying end-user needs throughout the development lifecycle (VESONDER *et al.*, 2018). It has been long applied in the spatial environment (POCHA, 1987); however, its emergence in the US Department of Defense (DoD) is recent and refers to the understanding and specification of threats and execution of an end-to-end mission (SOUSA-POZA, 2015; DAHMANN *et al.*, 2017). A well-established mission helps the analyses of the operational environment and guides the mapping from stakeholders' needs to capabilities, covering the technical needs (e.g., software, hardware, standards, and network) at design time and at runtime. Missions support the modeling of SoS defining the configurations of a system (i.e., constituents that will be part of the SoS and the way they communicate) (BEALE *et al.*, 2004), and impacting its individual configurations, due to changes or adaptations of the missions at runtime. Therefore, the mission design must be efficiently addressed from an enterprise analysis up to the selection of constituents that help meet the missions, and the necessary changes that impact on SoS operation (MOKHTARPOUR; STRACENER, 2017).

The SoS mission is vulnerable for researchers and industry professionals when technical interoperability requirements (i.e., architectural level) are not clearly identified (ESARY; ZIEHMS, 1975; DAZHI; XIAOZHONG, 1988; JAMSHIDI, 2008; SOUSA-POZA, 2015). Such requirements are derived from associated functional interoperability requirements (i.e., requirements engineering level) and support the *specification of an infrastructure resource, characteristics, ability or constraints of communication with data exchange related to the ability of a collaborator to ensure their collaboration regarding terms of composability, compatibility, integrability, interoperation, autonomy, variability, transparency, resilience, sustainability, and reversibility*. In traditional systems, the low-quality requirements document is responsible for 82% of rework and 44% of cancellations of projects (MARTIN, 1984; SKOKOVIĆ; RAKIĆ-SKOKOVIĆ, 2010). In SoS, although problems related to the low-quality requirements document persist, their consequences occur on a large scale and may cause the loss of human lives, assets, and economics value. Therefore, mission and technical interoperability requirements in SoS must be constantly elicited, analyzed, and monitored at the level of the global system according to the changes made by the CSs level that influence the achievement of the mission (NCUBE; LIM, 2018). Consequences of the failure of technical interoperability requirements must be identified and mitigated in time towards the continuity of the SoS mission.

At the architectural level, analyses of such requirements play a fundamental role in supporting the refining and initial selection of the CSs that led to a desirable emergent behavior at runtime (NCUBE *et al.*, 2013), which usually does not take technical and business stakeholders

(BASS *et al.*, 2012) seriously. Those requirements (e.g., hardware, software, network, protocol, and standards) can be hardly identified in early stages, since they are often associated with other functional requirements and architectural level (KNODEL; NAAB, 2016). Because of the evolving nature, SoS can present a large and complex set of requirements and technical interoperability requirements of several dependencies on each other and on the SoS mission (EASTERBROOK *et al.*, 1998; WANG *et al.*, 2016). The requirements analysis and architectural requirements specification must be continually revisited during other activities in the SoS lifecycle (BOEHM, 1984; BILA *et al.*, 2016); however, approaches of architectural requirements analyses are not appropriate for such interactions and relationship among CSs (i.e., technical interoperability requirements). As a result, practitioners are sometimes limited to the use of traditional approaches to elicit and analyze technical interoperability requirements in the SoS context.

Although technical interoperability and Requirements Engineering (RE) are two topic already researched in the last decade, the literature of these fields and of the field of mission engineering is still emerging in SoS domain and there is lack of theory that discusses them (VESONDER *et al.*, 2018; NCUBE; LIM, 2018). Mission engineering is a more recent trend in the SoS industry and technical interoperability requirements is an important base for inter-operation of SoS (TOLK; MUGUIRA, 2003; (DOD), 2008; DIALLO, 2010). Hence, there is a need to build a common theoretical foundation to advance the corpus of knowledge and research in RE for SoS applied to the business level and also to architectural level with focus on social needs (i.e., stakeholders). In particular, Panetto and Cecil (2013) and Ncube and Lim (2018) reinforce the needs of new RE theories for SoS or a number integrated of theories to provide a more structured, theoretical basis for analysis of technical interoperability requirements of these systems, the relation between mission and interoperability is also addressed in this research (VESONDER *et al.*, 2018).

1.2 Problem Statement

The identification, specification, design, and analysis of the mission and architectural requirements of SoS bring several challenges to the daily life of industry professionals due to their intrinsic characteristics (NIELSEN *et al.*, 2015b; VESONDER *et al.*, 2018). For instance, SoS shows an inherent and continuous evolution, and new missions and requirements can arise at runtime (this includes missions not foreseen at design time due to the evolutionary development of SoS and inevitable consequence of the autonomy of many CSs and their evolving capacity alone) (FISHER, 2006; LUNA *et al.*, 2013b). Another characteristic that contributes to modifications during the SoS operation is the emergent behavior, which results from the collaboration among CSs, whose effects often occur in cascade in a bidirectional way (i.e., from SoS to CSs, or vice versa). Such various and continuous changes directly impact the architectural requirements, specifically the technical interoperability requirements throughout subsequent stages of the SoS development at the operational level. The SoS mission depends on whether technical interoperability requirements are not properly identified and analyzed according to the business goal (JAMSHIDI, 2008; SOUSA-POZA, 2015). It can, therefore, fail if such requirements are inadequately treated, harm people and/or lead to the loss of valuable physical assets in the critical application domains where SoS operates. Despite their importance, the requirements have been analyzed using traditional proposition and theories on requirements not suitable for SoS (NCUBE; LIM, 2018).

In other perspectives, research on the SoS mission has often focused on analyses from architectural to operational levels and the initial problems faced by stakeholders, since both needs and SoS mission are identified at the enterprise level, and RE lasts the entire lifecycle of the SoS (DAHMANN, 2019; SOUSA-POZA, 2015). The evolution of the SoS mission can heavily influence both technical structure and behavior, which are mechanisms that support the derivation of technical requirements from the mission improving their control and evolution for the SoS design at design time and runtime (VESONDER *et al.*, 2018). Therefore, the use of a theory that addresses the architectural requirements analyses driven by mission engineering as a guide to such an approach can facilitate the understanding of the analysis process and its challenges.

1.3 Objectives and Research Question

The aim of this thesis is to build a substantive theory that support the process of technical interoperability requirements analysis driven by mission engineering for the design and development of SoS from the perspective of the industry professional and researchers with partnerships in industrial projects. Motivated by the several challenges addressed in the previous sections, we directed this work towards investigating the following research question, and specific objectives were established split into the following three situations:

Research Question: *How can we specify a theory to support the analysis of technical interoperability requirements with the use of SoS mission engineering?*

Situation 1: Since mission engineering is a broader process for satisfying users' needs reflecting multiple interactions among individual systems to achieve the capability that cannot be offered by them alone, an answer must be given to specific objectives (i) and (ii)

(i) *Investigate and map the relationship between mission, requirements, and architecture for the SoS field*

(ii) *Investigate and describe the way technical interoperability requirements are specified and analyzed for the SoS field*

Situation 2: SoS with central control are composed of individual systems that collaborate towards the accomplishment of the overall mission; some of them can evolve to withdraw knowledge from the central control. The situation must be analyzed by objective (iii).

(iii) *Identify and describe the relationship between mission and requirements, and the way requirements are specified and analyzed by mission engineering*

Situation 3: Since the SoS behavior can be foreseen at design time and is exhibited at runtime, and an unforeseen behavior normally emerges, an answer is also required for specific objectives (iv) and (v):

(iv) *Provide a comprehensive explanation of challenges and design of issues to support the analysis of technical interoperability requirements*

(v) *Analyze and describe the barriers and sustainability perspectives for technical interoperability requirements based on the industry*

1.4 Contribution

This research may contribute to the areas of mission engineering, requirements engineering, interoperability, software architecture, and systems-of-systems. ATLANTA Theoretical Framework is based on industrial practitioners and academics that develop projects in collaboration with the industry. Such theory supports analysis by identifying and connecting 10 categories to each other through knowledge areas related to a central category, namely technical interoperability awareness. The ATLANTA is generic and divided into two sets, namely non-crosscutting knowledge categories (i.e, domain, application, and infrastructure) and crosscutting knowledge categories (i.e., technical interoperability awareness, communication, and design decision) with specific solutions or discussions elaborated for each category, according to challenges faced by interviewees and/or compared with the literature (cf. Chapter 4). Therefore, ATLANTA enables analyses and description of the cycle of enterprise needs at design time and operational needs at runtime, since such requirements are essential to achieve the SoS mission.

Figure 1 shows these specific solutions organized as a conceptual model that illustrates the relationship between them and how the interaction evolves from an SoS organization to multilevel mission management ($ATLANTA_{M3}$) in mission engineering at runtime. In short, we first clearly identified the basic concepts of the **interoperability requirements and interoperation** (cf. Section 4), which is property of the **organization of an SoS**. Such concepts are associated with the blended category, which was established based on empirical data, and they are associated with ATLANTA and are part of the $Triplet_T$. To deign and model the SoS, we use the **Triple Model**, which aligns of the specification from mission, requirements, and architecture to both levels, business and SoS. The business level supports the understanding of the environment threats and opportunities and intentions and needs of stakeholders through the enterprise analysis

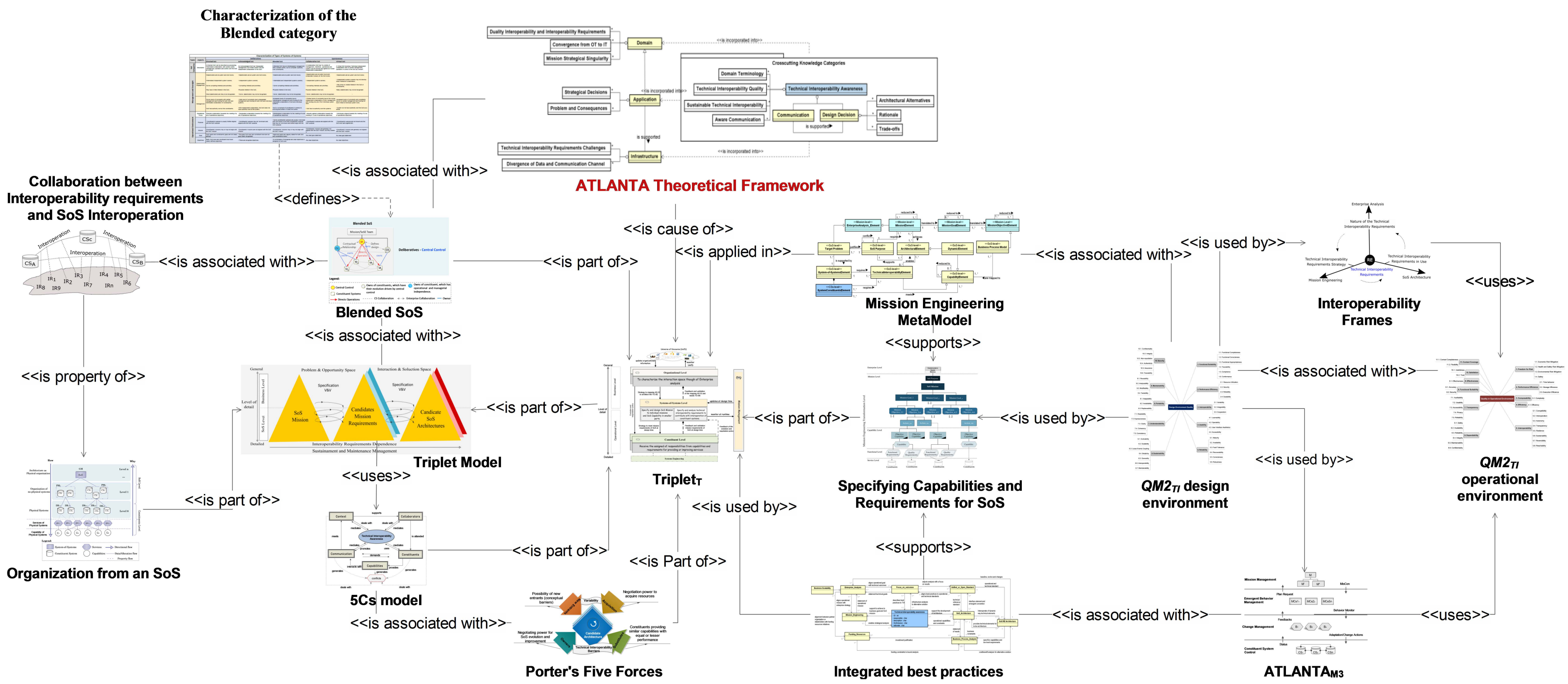


Figure 1 – Contributions and its relationships

Source: Elaborated by the author.

performed with the **5Cs model** and/or **Poter's Five Forces**. Based on this and by knowing the main stakeholders and their roles and activities, the mission planning can be performed and the initial acquisition of infrastructure is elaborated, and a proper SoS mission is defined. To the SoS level, the capabilities and requirements can be specified towards identifying the technical interoperability requirements at design time, i.e., $QM2_{TI}$ **design environment** or/and at runtime, i.e., $QM2_{TI}$ **operational environment**. Mission can change at runtime and management is performed by $ATLANTA_{M3}$ and new functions emerge as needs for the SoS to adapt or behave according to new scenarios. As SoS evolve, new requirements may arise with the structural growth and without adaptation of these new requirements or behaviors and then all are conducted at design time for analysis and management.

1.5 Research Method

Figure 2 illustrates the general structure of the thesis and the way the research questions and objectives were achieved. The thesis was divided into four parts, namely planning and foundations, methodology, development, and results.

1. **Planning and Foundations:** refers to Chapter 1, which addresses the importance and motivation of the research and defines the aim and the research question. Foundations regards Chapter 2 and is a combination of background and state-of-the-art through the application of systematic mapping and manual search in some areas, as in software architecture. Four systematic mappings were performed. Two were the **SoS development process** and **SoS RE** following the guidelines by Kitchenham and Charters (2007) and Petersen *et al.* (2015), whereas **SoS mission** and **SoS interoperability** were performed by snowballing, according to guidelines by Wohlin (2014a). The mappings were performed due to literature-driven analysis carried out in grounded theory method, in which the empirical data are compared with literature data and complemented by manual search, when necessary. The description of the conduction of each mapping is addressed in Chapter 3, as well as the results from a literature comparison of the grounded theory method.

2. **Methodology:** comprises a qualitative method applied in an exploratory way and using the grounded theory method (STRAUSS; CORBIN, 1990; CHARMAZ, 2005). The data acquisition form was chosen according to needs and population. In general, a literature-driven analysis was performed through systematic mappings (i.e., technical literature analysis and non-technical literature analysis), empirical data analysis with industry practitioners and industrial projects, data register in code, and discussions with empirical or academic experts at all stages towards improving the findings. Codification followed an adapted process proposed by Pandit (1996) and detailed in Chapter 3. 51 survey respondents and 27 interviews were involved in the coding process conducted in the development stage.

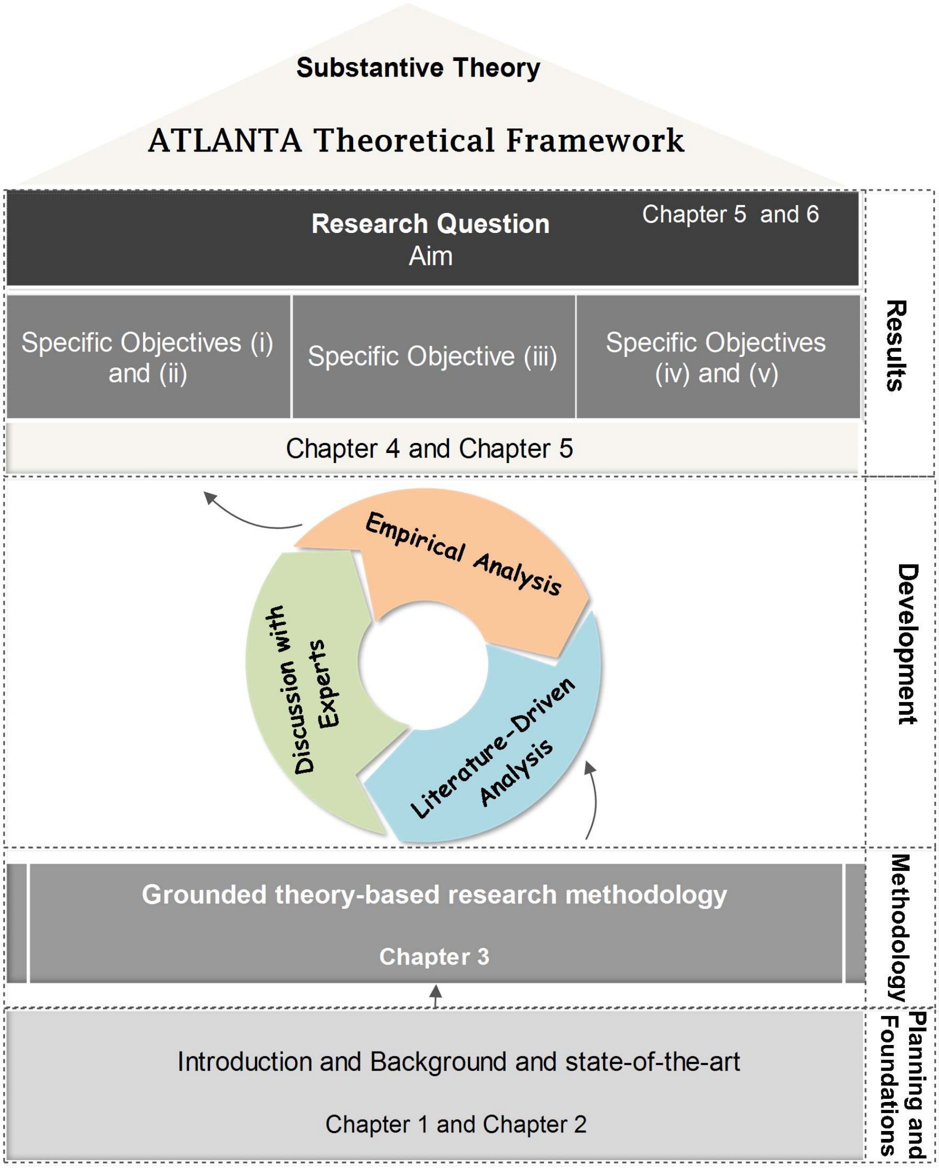


Figure 2 – Overview of the structure of the thesis

Source: Elaborated by the author.

3. **Development:** followed the four phases (i.e., research design planning, data acquisition, analysis conduction, and literature comparison) described in Chapter 3. As displayed in the continuous process (Figure 2), the three steps were performed until the theoretical saturation of the codes was achieved, and lead to the identification of the 11 categories (i.e., 10 category and central category) of theory and the knowledge areas. Figure 43 of Appendix A shows the way each systematic mapping contributed to both code and final comparison, and the empirical analyses performed until update of the substantive theory. Each of the 10 categories and the contributions of the literature comparison are described in Chapter 2 and the core category is addressed in Chapter 5.

4. **Results:** In summary, they address the research question and objectives proposed in

Chapter 1. The research question was answered in Chapter 5 with the use of information discussed in Chapter 4, as shown in Table 1 and Section 1.4. To address the research question, we performed one systematic mapping and one survey. Our goal was to investigate as has been interoperability requirements analyzed or specified in industry or academy, with academics involved in industrial projects. The preliminary analysis of findings revealed

Table 1 – The relation between specific objectives and sections

ID	Objective	Section
(i)	Investigate and map the relationship between mission, requirements, and architecture for the SoS field	(cf. Section 4.2.6)
(ii)	Investigate and describe the way technical interoperability requirements are specified and analyzed for the SoS field	(cf. Sections 4.2.1, 4.2.4, and 5.4.2.3)
(iii)	Identify and describe the relationship between mission and requirements, and the way requirements are specified and analyzed by mission engineering	(cf. Sections 4.2.10 and 5.4
(iv)	Provide a comprehensive explanation of challenges and design of issues to support the analysis of technical interoperability requirements	(cf. Sections 4.2.3, 4.2.7, 4.2.8, 4.2.9, and 5.5)
(v)	Analyze and describe the barriers and sustainability perspectives for technical interoperability requirements based on the industry	(cf. Sections 4.2.9, and 4.2.5)

that there are doubts about concepts (e.g., interoperability requirements x interoperation; quality attribute x technical requirements), beyond the interoperability requirements analysis, it is not performed by many of the respondents even though considered extremely important by them. Such information led us to deepen investigation through interviews and new systematic mappings.

The consolidated set of findings obtained from this new group of investigation combined with a previous one was used to build the substantive theory represented in form of a framework, called De**A**ling wi**T**h Mu**L**tilevel of Technic**A**l I**N**teroperabili**T**y **A**wareness (ATLANTA). This theory explores the analysis of technical interoperability requirements driven by mission engineering, focusing raising the challenges faced by practitioners and academics, and facilitating the understanding as well as the application of the concepts to minimize or solve such challenges. For instance, a challenge to practitioners was to identify the main barriers that impact the interoperation process. To solve this challenge, we identified which were the main technical interoperability barriers that could affect interpretation towards supporting its identification.

Technical interoperability requirements affect the success or failure of the mission of an SoS; therefore, the resultant theory enables the description of the behavior of these systems by considering important steps of analysis and models as the $QM2_{TI}$ quality model. ATLANTA is useful lens for researchers to explore mission engineering, SoS mission, requirements engineering, architectural analysis, and the interaction among them. From the perspective of practitioners or academics involved in industrial projects, it is a valuable tool for enterprises to have insights of how to initiate, maintain or evolve the analysis technical interoperability requirements, or only to identify an SoS mission, using strategical and agile approach at the industrial environment of a sustainable way. However, knowledge of technical interoperability awareness is required to implement actions towards solutions for the next generation of SoS, which is more challenging in its design and development.

1.6 Research Group and collaborations

This PhD research is a contribution to science; it was developed in the Laboratory of Software Engineering (LabES) at ICMC/USP, and focused on SoS, software architecture, mission engineering, requirements engineering, and interoperability. It involved the SofTware ARchitecture Team (START), one of the research groups of LabES that develops larger projects (e.g., "SASoS: Supporting Development of Software Architecture for Software-intensive Systems-of-Systems" (supported by Fapesp, grant 2014/02244-7), which has created a process that systematizes the design of SoS software architectures, and "RASoS: Construction of Reference Architecture for Systems-of-Systems" (supported by FAPESP, grant 2017/06195-9), which established the construction of reference architectures for SoS.

This PhD project is an integral part of SASoS and RASoS. It has been conducted in partnership with researcher Dr. Ricardo Yassushi Inamasu from Embrapa Instrumentation, Prof. Dr. Carlos Eduardo de Freitas from ESALQ/USP and MSc Wilson Milani Zambianco, both from Piracicaba campus, Prof. Dr. José Luiz Braga, retired teacher from the Federal University of Viçosa (UFV), and Prof. Dr. José Carlos Maldonado, retired teacher from the ICMC/USP, which has supported all activities of this PhD, as well as this PhD candidate since the early stages. Another important partnership was established with the embedded systems engineering group of Fraunhofer Institute for Experimental Software Engineering (Fraunhofer IESE) under the supervision of Prof. Dr. Dieter Rombach, during an internship in Kaiserslautern, Germany. The excellence and uniqueness in software engineering of the above-mentioned groups and researchers have opened the frontiers of knowledge regarding this research.

1.7 Thesis Outline

This chapter shows an overview of the context and motivation for the development of this doctoral project. It addresses the problem statement, research question, and main contributions. The remaining chapters are organized as follows: Chapter 2 is devoted to the background and state-of-the-art, and Chapter 3 describes the Methodological Procedures adopted; Chapter 4 describes the categories for the construction of the substantive theory, and Chapter 5 details the construction of the ATLANTA Theoretical Framework: dealing with multilevel technical interoperability awareness; finally, Chapter 6 provides the conclusions and revisits the Thesis contributions.

The other material accompanying this Thesis encompasses: Appendix A, shows the development process of the substantive theory in Business Process Model and Notation (BPMN) model and all interactions. Appendix B, describes the difference between system engineering and SoS and Appendix C illustrates the grounded theory proposed by Pandit (1996). Appendix D shows the protocols of the systematic mapping, whereas Appendix E displays the primary studies selected for the SoS development process. Appendices F and G describe the protocol and the primary studies selected for the systematic mapping in formal and semi-formal requirements, respectively.

Appendix H introduces the protocol for interoperability and mission systematic mapping, and Appendices I and J show the selected primary studies. Appendix K details the protocols and the question raised during the survey, and Appendix L describes the script that guided the interviews. Appendix M displays the code, sub-categories, and category identified in the coding process by Atlas.ti tool for the construction of the theoretical framework. Appendix N provides information on the 27 interviewees consulted. Appendix O describes each quality attribute considered in $Q2M_{TI}$, and Appendix P shows the way the selection process of each quality attribute was performed. Appendix Q describes ISO/IEC-25010 (2011), which was not considered in our model, and Appendix R provides the list of publications.

CHAPTER 2

Background and State-of-the-art

2.1 Initial Remarks

The increasing complexity of most systems has placed pressure on traditional design methodologies[1] (DECHER, 2010a). In a globalized environment, legacy and single systems have been connected towards the construction of more modern and robust ones that can meet new requirements (NCUBE, 2011). Such a change has resulted in more complex systems, called SoS, which can improve collaborations among CSs. (NIELSEN *et al.*, 2015b). SoS are usually the outcome of the interoperability among heterogeneous and independent CSs that work together to carry out a mission (MAIER, 1998; DELAURENTIS, 2005), can be found in several application domains, including critical ones (e.g., industry 4.0, agro4.0, energy, aerospace, telecommunications, healthcare, disaster management, and military environment) (DELAURENTIS, 2005; MITTAL; RAINEY, 2015).

On the other hand, requirements analysis is the process that determines, specifies, models, negotiates, and refines requirements from a high-level strategical goal to low-level technical requirements. Its success involves the understanding of users', customers', and other stakeholders' needs, as well as the operational contexts of the developed software. The document of capability and requirements must be validated during and after a negotiation process for avoiding the insertion of inconsistent, incomplete, incorrect, and ambiguous information in a specification (CHENG; ATLEE, 2007; ALSHAZLY *et al.*, 2014). An analysis not correctly performed cannot detect errors in the early stages of the development, which persist in the entire lifecycle; moreover, on a large scale, it may cause economic and human lives losses (ZOWGHI; COULIN, 2005; CHAKRABORTY *et al.*, 2012). Several initiatives, such as Goal-oriented requirements engineering (GORE) (YU; MYLOPOULOS, 2017) and Model-Based Requirements Engineering (MBRE) (RIBEIRO FABÍOLA *et al.*, 2016) have emerged for capabilities and requirements

[1] For this Thesis, we consider traditional design methodologies as methodologies used to develop systems not classified as SoS.

modeling and their specification with the use of scenarios (BASS *et al.*, 2012) and structured languages (e.g. Easy Approach to Requirements Syntax (EARS) (MAVIN *et al.*, 2009))

This chapter is organized as follows: Section 2.2 provides an overview of SoS and the evolution of the term from 1950 to 2018, and Section 2.2.1 describes its essential characteristics. Section 2.2.2 discusses the common types of SoS; Section 2.3 is devoted to the mission-driven SoS development, and Section 2.3.1 addresses the main types of coalitions and mission analysis and the impacts on the development of the SoS architecture. Section 2.3.2 focuses on the emergent discipline of mission engineering and the most relevant types for this research. Sections 2.4 and 2.4.1 describe the requirements engineering for SoS, and the specification and modeling languages, respectively. Section 2.4.2 reviews languages and techniques for SoS requirements modeling, and Section 2.4.3 discusses the importance of requirements analysis in the SoS domain.

Section 2.5 provides an overview of software architecture; Section 2.5.1 describes interoperability and its models of evaluation and measurement, and Section 2.5.2 focuses on technical interoperability. Section 2.6 discusses some related work; finally, Section 2.7 is devoted to the final considerations on the topics covered in the chapter.

2.2 Systems-of-Systems

A system is usually understood as a set of elements (e.g., software, hardware, processes, humans, and data) that work together for generating a behavior or function not available from individual elements (LEVESON, 2013; ISO/IEC/IEEE-24765, 2017). Under this definition, the concept of a system can be applied to SoS, subsystems, product lines and families, whole enterprises, and any other aggregations of interest (ISO/IEC/IEEE-42010, 2011). The first studies on SoS were conducted by Boulding (1956), followed by Ackoff (1971), Jacob (1974), and Jackson and Key (1984). Boulding (1956) pictured SoS as a "spectrum of theories" that might be a theoretical construction larger than the sum of its parts. Ackoff (1971) considered SoS a unified and integrated set of systems concepts, whilst Jacob (1974) analyzed the concept in relation to biology. Jackson and Key (1984) considered the application of problem-solving methodology an interrelationship among independent systems in an operation domain (e.g., SoS methodology). However, term SoS was applied as an engineered technology system only in 1989, in the military domain, by the Strategic Defense Initiative (SDI) (CONGRESS; SERVICES, 1989).

Term SoS as currently known was introduced by Eisner *et al.* (1991) and Eisner (1993) and has been researched since then. Figure 3 shows a summary of the academic progress on research on SoS from 1950 to 2018. The red color font was extracted from Gorod *et al.* (2008), and the purple color was obtained from a previous mapping (LANA *et al.*, 2016) updated[2].

[2] The systematic mapping was conducted April/2015 to October/2015 and updated from September/2018 to December/2018

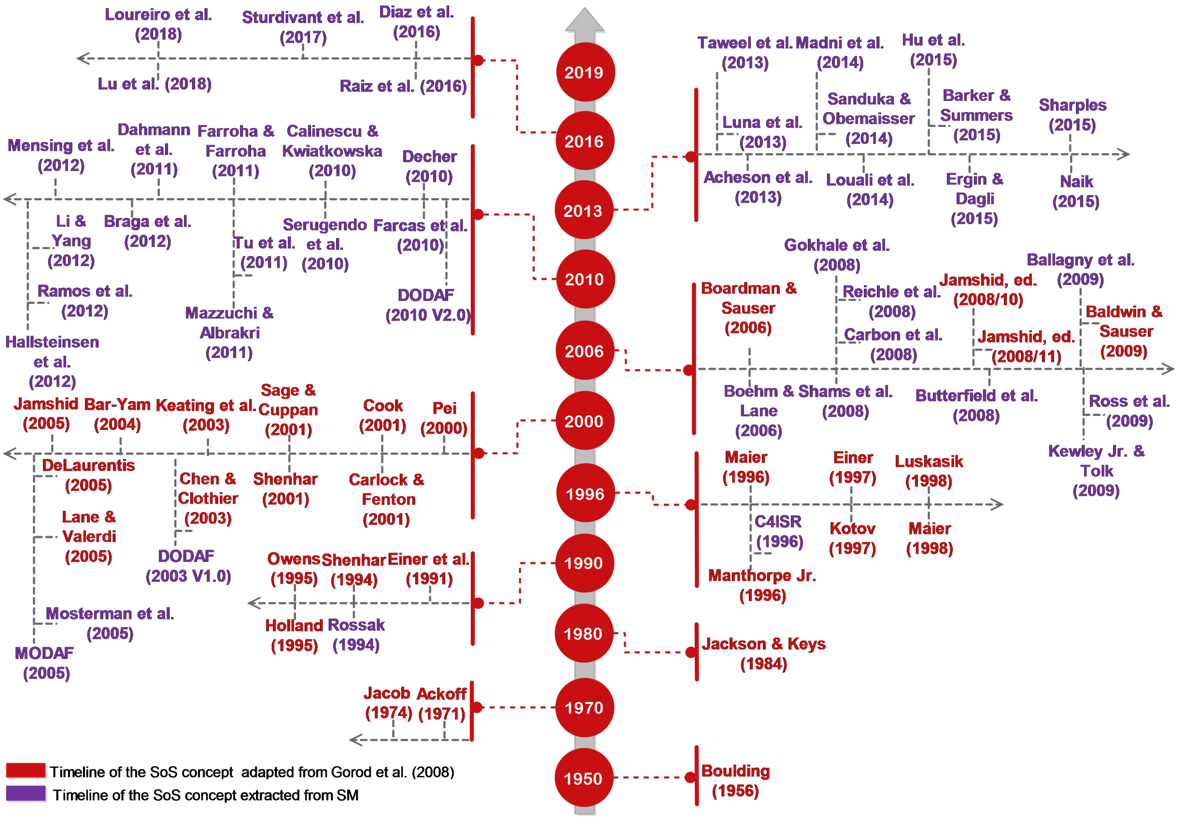


Figure 3 – Timeline of the history of systems-of-systems

Source: Elaborated by the author.

Maier (1996), one of the most influential contributors in the SoS domain, proposed a set of fundamental properties to characterize an SoS and differentiate it from other complex and large-scale monolithic systems. Such properties encompass operational independence, managerial independence, evolutionary development, emergent behavior, and geographic distribution. They were revisited twice (MAIER, 1998; MAIER, 2014) and, in 2014, Maier (2014) concluded only three (i.e., operational and managerial independence of CSs and emergence behavior of the SoS) were essential for the definition of an SoS (the other two, i.e., evolutionary development and geographic distribution, might be present, however, they are not absolute). The properties have been used for distinguishing systems and SoS (cf. Appendix B), as well as the characteristics of stakeholders, objectives, management, and lifecycle. Ownership, described by Maier (1996), Boardman and Sauser (2006), recognized similar principles and the characteristics shown in Table 21.

Although the literature reports several SoS definitions, none has been universally accepted, and many have been compiled by researchers, such as Jamshidi (2008), Nielsen *et al.* (2015a), and FREng (2016). The present study has adopted the essential Maier's properties and an SoS was defined as *a concept that represents an organization of heterogeneous and independent constituents, which have their own mission (i.e., individual mission) and that interoperate to* *carry out a greater mission, also known as a global mission.* SoS are present in different domains, and failures in subsystems of a CS, or in the own CS, can directly affect the performance of the interoperation of those systems and cause large losses of assets, economic, environmental, and, especially of human lives, whenever they reach the fundamental structure of SoS (AVIZIENIS *et al.*, 2004; KUMAR; MERZOUKI R.AND BOUAMAMA, 2017).

2.2.1 Characteristics of Systems-of-Systems

SoS has been recognized as the biggest trend according to which both industry and government, as DoD, shift from platform-centric to focused on capabilities and broader users' needs and supply of value (LUBAS; COMMAND, 2017). Its inherent set of properties differentiates it from monolithic systems (i.e., traditional systems), originally identified by Maier (1998). Since then, SoS has been expanded (e.g., dynamic architecture and self-management) and rewritten towards fitting specific application domains (BOARDMAN; SAUSER, 2006; FIRESMITH, 2010; NIELSEN *et al.*, 2015a). The constituents of an SoS are (i) *operationally independent*, i.e., they have their own functionality even when not cooperating with other constituents, and (ii) *managerially independent*, i.e., they are managed independently by their owners. SoS display an *Emergent behavior*, i.e., new functions cannot be performed by a CS separately. Such features are characterized only through interactions among constituents over time (KEATING *et al.*, 2015). Emergent behaviors may originate in constituents and trigger new behaviors at the SoS level, and vice-versa. Moreover, SoS involves two additional properties, called *Evolutionary development* and *geographical distribution*. The former is related to the SoS capacity for evolving in response to changes in its environment, CSs, or missions, functions, and purposes. For instance, an SoS must absorb the constituents' changes, which might affect the achievement of its global mission and proper functioning. The latter refers to the properties of CSs, i.e., they interact among themselves exclusively in terms of information exchange, and are distributed over different locations.

Some observations on Maier's properties can lead to consequences in the development practices for SoS (AXELSSON, 2015b; ALBERS *et al.*, 2018):

- *life cycle*: CSs have different and unsynchronized life cycles, which directly impacts on the SoS evolution with changes in its fundamental structure, including its CSs and their relationships (HAN; DELAURENTIS, 2013; AXELSSON, 2015b; ALBERS *et al.*, 2018). Below are the consequences arisen:
  - The architecture of the SoS must be open to changes and evolve over time for encompassing new situations. The architecture of potential CSs should be targeted at flexibility and resilience, especially regarding their interfaces;
  - The design environment of SoS must be (ABBOTT, 2006): (i) *Open at the top*, i.e., an SoS is continually open to new constituents, with no top-level system defining it;

(ii) *Open at the bottom*, i.e., the lowest level of the SoS, (e.g., a specific communication stack) can be changed at any time; and

(iii) *Continually evolving, but slowly*: an SoS is never complete, since it evolves with changes in the surrounding environment and in at least three ways, called standards and interfaces adjustment, technological changes, and feature and needs modification;

- Managerial principles must ensure the SoS purpose is maintained while the system is changing;
- Methodologies must support the identification of the main stakeholders in a context of multiple stakeholders distributed in different organizations and the way they impact the entire SoS;
- Approaches must support the identification and management of SoS missions, their changes over time, and the way requirements are derived from those missions to provide value to the main stakeholders; and
- Traditional project-oriented management models will most likely not work. Each CS is subject to its own ongoing change projects, which must interconnect with the SoS evolution.

- *ownership*: it is related to decisions on the SoS design, which, in most cases, result in negotiations across organizational borders. Solutions must restrict the autonomy of any individual system, but still, the SoS mission must be accomplished. Furthermore, the SoS liability is shared among the organizations behind its CSs (NCUBE *et al.*, 2013; AXELSSON, 2015b).

- *value*: although CSs keep their autonomy to be part of the system, they can modify their behavior towards gaining benefits, maintaining the original proposal, inside or outside the SoS. Moreover, the requirements trade-off between the CSs mission and the SoS mission (AXELSSON, 2015b) requires a clear traceability of the constituents' mission and description of their base and the way they can contribute to the accomplishment of the mission (ALBERS *et al.*, 2018).

- *emergence*: it fulfills the SoS mission through the CSs interoperation. Therefore, the principles that control the constituent's behavior must be understood to be aligned with the SoS mission (HAN; DELAURENTIS, 2013). Appropriate mechanisms, including regulating ones that minimize inappropriate behaviors, and awarding ones that encourage a desirable conduct, must be identified (HAN; DELAURENTIS, 2013; NCUBE *et al.*, 2013; AXELSSON, 2015b).

2.2.2 Classification of Systems-of-Systems

SoS can be classified and compared in relation to governance and managerial levels. Governance and managerial control are akin, however, they work at different organizational levels. The former is concerned with a high level of decision-making processes that will impact the entire organization, including the SoS and achievement of its mission (e.g., policies, regulation, and laws), whereas the latter is associated with operational activities and the way CSs and/or SoS can be changed over time (SOMMERVILLE, 2016). The classification of each SoS affects the whole lifecycle, from requirements to maintenance and evolution of the organizations and systems, therefore it is required to support the identification of architectural principles that better assist the organizational decision-making and SoS architecture, (e.g., regulations, opportunities, adaptations, cooperation, among them, technologies, among others) (MAIER, 1998; LANE, 2013; NIELSEN *et al.*, 2015a).

The organization of the categories considered (i) the existence of a central control that manages the interoperation of constituent systems ensuring the fulfillment of the global mission- only decision-making processes related to the high level of SoS are included in this category; and (ii) the managerial control of each constituent together with its owner (see Figure 4) (MAIER, 1996; DAHMANN; BALDWIN, 2008; HENSHAW, 2013). Spontaneous organizations

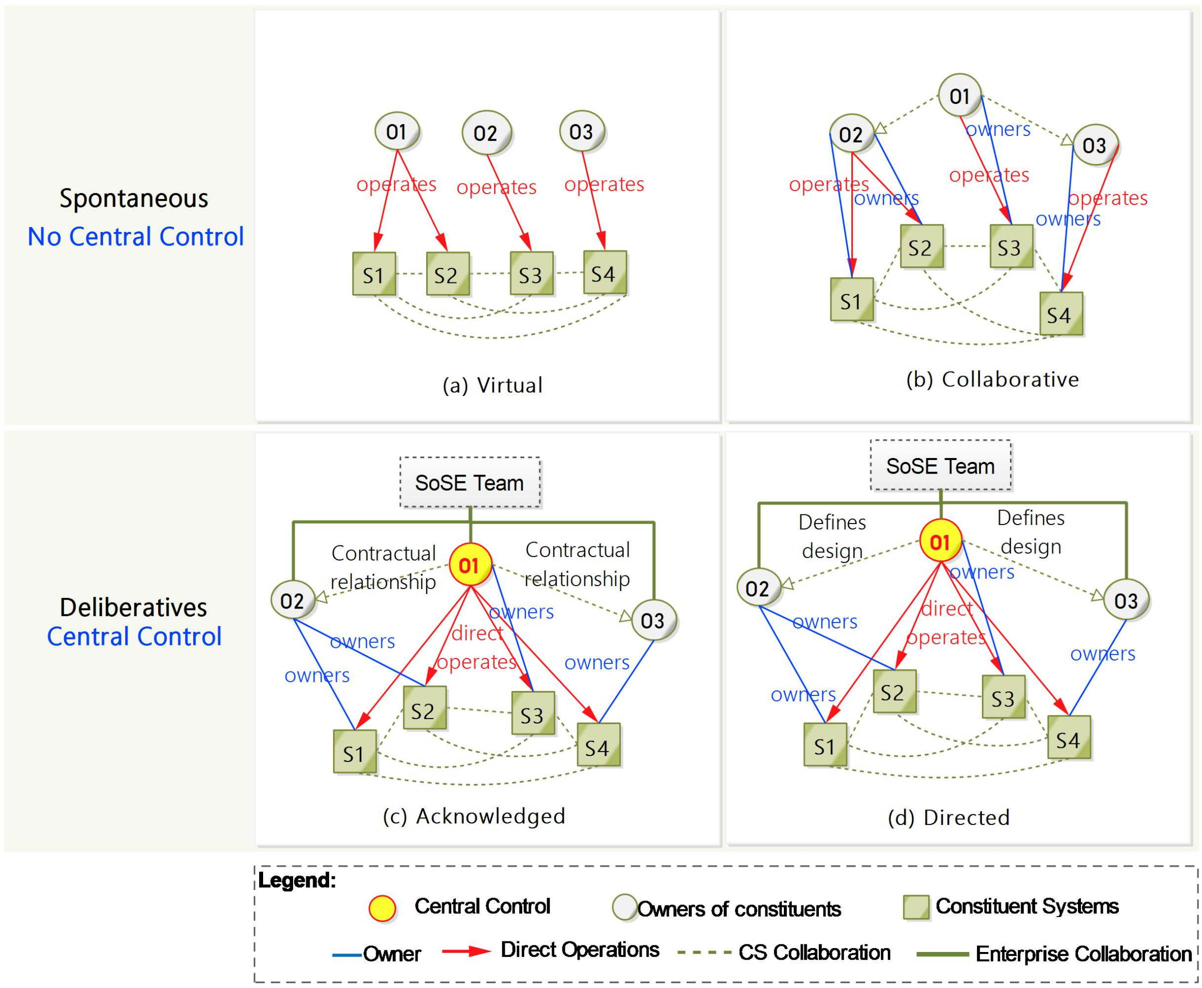


Figure 4 – Categories of systems-of-systems

Source: Adapted from Henshaw (2013).

are formed by SoS without a central control, thus increasing the organization from virtual to collaboration categories. These systems are established in a "voluntary" manner, depending primarily on the ability of each constituent to interoperate with each other. The virtual category (i.e., Figure 4a) has neither the central authority to manage its constituents' activity, nor a clear purpose. Interoperation is accomplished by recognized protocols, or standards, rather than through agreements between pairs of constituents. Therefore, constituent owners (i.e., O1, O2, and O3) access other constituents through their systems towards acquiring individually sought benefits, although a high-level emergent behavior (i.e., global mission) may be still exhibited (HENSHAW, 2013). Whereas the constituents of Collaborative category (i.e., Figure 4b) work together and more or less voluntarily to fulfill agreed central purposes, constituent owners (O1, O2, and O3) operate their systems and collaborate with others towards accomplishing some shared mission.

On the other hand, deliberative organizations are composed of SoS with a central control, thus increasing the managerial control from acknowledged to directed categories. Such SoS are defined by the negotiation of capability needs among CSs, covering SoS needs emerged and not met by dynamic adaptation at runtime, and by the control of the constituents evolution that affects the global mission. A deliberative organization comprehends two types of systems, called acknowledge and directed (HENSHAW, 2013). An acknowledged SoS is comprised of constituents with independent ownership, objectives, funding, and development approaches, but no complete authority over them. Figure 4c shows a general organization - owner O1 (i.e., central control) together with the SoSE team direct the choice of constituents and operation, whereas owners O2 and O3 have a contractual relationship (e.g., Service Level Agreement (SLA) or smart contract) with the owner's central control. The central control (O1) has less control over the constituents owned by O2 and O3 (i.e., S1, S2, and S4) and must rely more strongly on the influence and negotiation between them. Figure 4d) shows the directed SoS that has specific purposes and complete authority over the evolution of its constituents managed by a central control and engineering team. The SoS categories might be geographically distributed and CSs must be in accordance with different national laws and regulations. Therefore, O2 and O3 are given directions by O1 on both specification and operation of their constituents (i.e., O2 - constituents S1 and S2, and O3 - constituent S4).

2.3 Development of Mission-Driven Systems-of-Systems

Important concerns about the development of systems are the understanding of their mission, the capabilities provided, their attending, and the values delivered for end-user and customers. No question can be answered if the mission is not understood, since it drives the systems development and evolution, including the SoS. Therefore, a system's mission heavily draws the attention of both industry (ESARY; ZIEHMS, 1975) and academia (DAZHI; XIAOZHONG, 1988). Mission has been researched since 1970s and defined as an activity divided into consecutive time periods (or behaviors), in which essential tasks must be performed through reliable software and hardware (ESARY; ZIEHMS, 1975) for its achievement. According to Dazhi and Xiaozhong (1988), mission is a task performed when the system is altered, such that the logic model changes at a specified time.

The concept of SoS mission is analogous to the traditional systems; it supports the definition of the operations and guides the business necessities mapped for the capabilities. A well-established mission is related to collaboration among different stakeholders (e.g., project team, technical team, and main SoS stakeholders) and application of the multidisciplinary knowledge. It facilitates the identification of a capability required for the acquisition of CSs and identification of SoS requirements to be refined in the design phase (DOD, 2018). Moreover, it provides the SoS development team (e.g., SoS engineers, analysts, and/or SoS architects) with a broader understanding of its context and the main capabilities necessary for its success.

The SoS mission is classified at least into the following three types: (i) *global mission*, i.e., attributed to the high level of an SoS; (ii) *partial mission*, i.e., a mission of a non-physical organization of SoS met through the collaboration of various constituents, and (iii) *individual mission*, i.e., mission of each constituent that will interoperate towards accomplishing the a global mission. It refers to the overall operational capability of further mission objectives (MO) of an SoS and shows its transformation actions that are often vaguely formulated, although important for the assessment of both value (relative to priority) and cost of the objectives. Therefore, an SoS mission can be a statement in a short sentence, or a paragraph with the needs of business stakeholders arranged in the SMART[3] methodology (DORAN, 1981) together with Kipling's Questions (also known as 5W1H[4] problem-solving method). However, this statement must be different for each SoS organization. SMART might be used with 5W1H for deliberative organizations, since their main stakeholders can be identified. Several constituents are not known in spontaneous organizations, therefore, only SMART can be applied, and the interoperation of CSs have been supported by various coalitions types (KAZMAN *et al.*, 2013), such as:

- (i) Greenfield: no architectural constraints besides those that are a consequence of the SoS Mission and CSs will be developed to operate for a long time. An example of a greenfield SoS development would consider all constituents of autonomous vehicles that have been newly designed, or the CS of the eHealth domain required to be newly designed for the establishment of an SoS.
- (ii) Brownfield: in principle, the CS can implement new requirements for providing new capabilities or improving the existing ones, and middleware or mediators can be introduced or replaced towards improving the adaptation process. Brownfield is focused on acknowledged and directed SoS, thus facilitating the negotiation of changes with the CS

[3] SMART is a mnemonic device that defines Specific, Measurable, Achievable, Results-oriented and Traceable

[4] 5W1H is a mnemonic device that defines “What”, “When”, “Where”, “Why”, “Who”, and “How”.

owner (e.g., Global Earth Observation[5] (GEOSS) and Flood Monitoring SoS (HORITA *et al.*, 2014)).

- (iii) Closed Source: a constituent has limited access for changing or negotiating with the owner. Interoperation can be established by a facade with the use of mediators (e.g., collaborative and virtual SoS), or with a short time frame for a specific mission. For instance, SoS mission “Missing Malaysia Airlines Flight MH370/MAS370”, which started in March, 2014, and finished in January, 2017, involved over 25 countries and more than 100 capabilities provided by different ship, aircraft, helicopters, vessels, and satellites (ATSB, 2017).

The SoS Architecture must be open to changes and evolve over time towards encompassing new situations and playing a fundamental role in both development and quality of the systems. The CS independence must be maintained, since such constituents can evolve, or even meet a second SoS (i.e., SoS_2), but must accomplish the SoS global mission of SoS_1. Decisions made at the architectural level influence the achievement of business goals and the SoS global mission, and their capability and quality requirements (JAMSHIDI, 2009). Such a mission can be refined to more specific levels, and addressed by architectural elements in the architecture. Refinements enable SoS requirements engineers, systems architects, or SoS architects to evaluate the compliance from high-level needs for architecture, including mission, capabilities requirement, specification, and element architecture.

2.3.1 Mission Analysis of Systems-of-Systems

Mission analysis is a critical step in the SoS development, since it supports the understanding of the business or mission problem and the opportunities and initial characterization of solution space; its omission generally results in risks to other analyses (JAMSHIDI, 2008; SEBOK, 2014; ISO/IEC/IEEE-15288, 2015) related to needs, capability gaps, and solutions to be applied for the evolution of the business objectives and satisfaction of the needs, solution space, and constraint space. No effective architecture can be designed if such constraints are not considered (JAMSHIDI, 2008; SEBOK, 2014; WALDEN *et al.*, 2015).

Different concepts and documents have been structured and applied towards helping the mission analysis process (e.g., Concept of Operation (ConOps), Operational Concept (OpsCon), acquisition, deployment, support, and retirement concept (WALDEN *et al.*, 2015)). ConOps has been developed for enterprise-level and describes the organization’s assumptions or intents regarding an overall operation or series of operations for the SoS development, existing and future CSs. It defines, but is not limited to, constraints that affect the SoS organization, business strategies, interactions among stakeholders, clear statement of responsibilities, and delegation of authorities (ISO/IEC/IEEE-29148, 2011). It is applied for the description of the mission,

[5] <https://www.earthobservations.org>

understanding of the project and users, and preparation of the preliminary OpsCon, which refers to "what" the system will do and "why" (rationale) from the user's viewpoint. It summarizes the business needs from an operational viewpoint for the solution space. A final version of OpsCon should be iteratively established with business and stakeholders' needs refined through a feedback from the RE, capability engineering, and architectural design processes in the SoS development (ISO/IEC/IEEE-29148, 2011; WALDEN *et al.*, 2015). Acquisition describes the way the CS will be acquired, i.e., it is related to the way SoS is developed, whereas deployment concerns the SoS validation delivered and introduced into operations. Support is related to the infrastructure required for supporting the SoS after it has been deployed, and retirement with the SoS and/or CS is removed from the operation (WALDEN *et al.*, 2015). The verification of the SoS mission is an essential activity; however, the literature lacks empirical or academic studies that report it; most of them are concerned with the validation of SoS mainly through simulations (DIALLO *et al.*, 2018; MITTAL S.AND DIALLO; TOLK, 2018). Kumar and Merzouki R.and Bouamama (2017) and Ayala (2015) evaluated fault tolerance through the supervision of the SoS and their CSs. Therefore, when a CS makes a change that impacts the SoS global mission, or a CS has a fault, this change or fault can be quickly identified and corrected by a managerial decision-making process.

Such concepts and documents have been used in industrial sectors (e.g., aviation administration, aeronautic transportation, healthcare system, and space) with adapted definitions, such as operational, usage and/or technological concepts. For instance, mission analysis in the space domain describes the mathematical analysis of satellite orbits performed for achieving the objectives of a space mission (SEBOK, 2014). The results of the analyses support the synthesis that all competing needs and vexing constraints are merged into a solution (JAMSHIDI, 2008). Therefore, the SoS and mission team play a key role, i.e., the refinement of the SoS global mission into mission objectives, partitioned into smaller ones and used by SoS architects or engineers for decision-making in the architectural design of SoS (JAMSHIDI, 2008; CROWDER *et al.*, 2016). Requisitions are made by information sources even when those systems running have changed the SoS behavior and generated new demands that impact the structure of the architecture. Such demands must be analyzed and treated for maintaining compliance, and partitioned into smaller units to be met by CSs. If they are not well performed and structured, they lead to negative outcomes, which must be overcome in the development phase, which increases rework and redesign of the SoS architecture. The success of the SoS architectural design is the systematic specification of the global mission up to SoS requirements through mechanisms of management of the decomposition cycle, i.e., derivations and traceability forward the architectural elements to the SoS global mission (i.e., backward traceability).

2.3.2 Mission Engineering of Systems-of-Systems

Mission engineering is an emerging discipline designed as a suitable area to integrate the systems engineering, operations, and missions (SOUSA-POZA, 2015). It aims at translating stakeholders' mission needs into implementable software-intensive system requirements (CARLOCK; LANE, 2006). The concept has been used in several ways, and two examples of its applications are found at NASA and DoD. A compendium of its use by US organizations outside DoD and non-US organizations is available in a survey conducted by Vesonder *et al.* (2018), with 32 mission engineers. A mission engineering competency framework was proposed with six key competences for a mission engineer that worked with planning, architecting, and designing of SoS, called Discipline & domain foundations, Mission concept, System engineering skills, system mindset, interpersonal skills, and technical leadership.

NASA mission engineering is design-driven, and provides a highest level of assurance of accomplishment of a mission established. The tasks of space mission are technical, with few adaptations after analysis and design; therefore, mission engineering was defined as the refinement of mission parameters and requirements towards meeting broad and usually ill-defined objectives of a space mission at reduced cost and risk (LARSON; WERTZ, 1991; WERTZ *et al.*, 2015). However, the procedure is related to the integration of new technologies and resources to solutions already implemented, thus requiring consistent decision-making for improving the SoS architecture without affecting the mission results.

On the other hand, DoD's mission engineering is more related to the acquisition and expansion of new capabilities. Since 1990s, DoD has invested in development approaches (e.g., Systems-of-Systems Engineering (SoSE) and Department of Defense Architecture Framework (DoDAF)) for dealing with challenges imposed by SoS. Although substantial advances have been made, mission problems have not been consistently supported or completely satisfied (SOUSA-POZA, 2015). The mission success demands the development of complex and progressive processes for its definition, RE, architecture, design, verification, validation, evaluation, and testing, as well as operational analysis of the mission, SoS, and constituent levels for effective outcomes (ESTEFANIA, 2010). Towards a minimizing the effects, the planning, analysis, organization and integration of operational resources and legacy must be deliberated, so that current and emerging systems can achieve the desired effects of the operational mission (DAHMANN, 2019). A well-engineered mission architecture promotes the design of more resilient and adaptive systems to which new capabilities and technologies can be added.

Mission engineering supplies a visual definition of capability and its linkage to requirements, supporting the identification of the mission and business process and strategies. Moreover, a design decision becomes a channel for collaboration among SoS elements (CARLOCK; LANE, 2006). Table 2 shows the three elements considered in the acquisition of mission, called: (i) **Proactive** - an initial process of mission engineering, in which primary analyses are performed to preclude future problems across mission outcomes; (ii) **Reactive** - refers to the activities

Table 2 – Mission engineering classification

Source: Adapted from Dahmann (2019).

Mission Engineering		
Proactive	**Reactive**	**Opportunistic**
Its initiation is based on the recognition of the primary importance of a mission or an enterprise outcome	Is triggered by issues or gaps identified in the mission performance or an element supporting the mission	Responds to a potential of new technology or anoother change that offers potential mission advantage technology
Addresses the " health" of the "end to end mission" to identify gaps, issues or opportunities for maintaining or enhancing the mission outcomes	Identifies the sources of mission gaps or the effects of problems with systems or other elements on mission outcomes	Addresses the question of the impact on mission outcomes by introducing new technology, systems or processes
Can lead to the identification of gaps or issues that may be affecting the mission outcomes or may do so in the future (risks)	Assesses the impact of possible changes to address issues or gaps on other elements or systems supporting the mission	

triggered in the execution of the mission (e.g., gap, threats, and mission supporting elements) that must be evaluated, and feasible modification proposals; and (iii) **Opportunistic** - responds to environmental changes with modifications or introduction of capability, process, and technology, which impact the mission outcomes.

2.4 Requirements Engineering for Systems-of-Systems

The critical role of requirements for the success (or not) of the software development has been recognized over the past three decades (JARKE *et al.*, 2011). Traditional RE processes were defined to support the development of new systems whose key requirements are defined before the system is architected and implemented by a single authority or customer who controls their development (HALLERSTEDE *et al.*, 2012). SoS requirements are defined as either SoS requirements that are properties of the overall SoS, or CS level, allocated for a particular CS (NCUBE; LIM, 2018) (Appendix B). Table 19 shows the drivers of systems engineering and SoSE and the problem domains that modify the traditional way of thinking of complex systems constructions and their requirements (KEATING *et al.*, 2015), and Table 20 displays the characteristics between traditional engineering and SoSE, highlighting the multiple levels (e.g., stakeholders, lifecycles, and managers) of an SoS. Table 21 shows the differentiation between the attribute of systems and SoS, also discussed in Maier (1996). Although traditional RE has been extensively researched and applied at distinct degrees of success, it is not sufficient for SoS development, which makes it even more challenging (DAHMANN; BALDWIN, 2008; LEWIS *et al.*, 2009; KEATING *et al.*, 2015; NCUBE; LIM, 2018)

Due to their characteristics (e.g., emergent behavior and evolutionary development), the definition of SoS requirements is limited and no methodology is adequate for real-world SoS problems (WALKER, 2014). Therefore, SoSE practitioners are narrowly use traditional techniques, methods, and tools for identifying requirements of complex problems. New techniques are necessary for dynamically fulfilling such unstable and fragmented requirements, which are continually evolving and changing (NCUBE, 2011; WALKER, 2014). Techniques should consider problems associated with requirements, but also deal with the partitioning of SoS problems into several CSs, which evolve independently (NCUBE, 2011). Particular attention should be given to the RE process by its community towards the understanding of stakeholders' demands, interoperability of SoS, interoperable architecture, and dynamic evolution of SoS, and the impacts of the emergent behaviors on the requirements stability (NCUBE, 2011; KEATING *et al.*, 2015).

Walker (2014) proposed an SoS Requirements Definition Method for offering SoS practitioners a systemic and flexible way of defining, unifying, and measuring SoS requirements. It is a system-based method that uses discoverers' induction coupled with coding techniques from the grounded theory method, and can be applied to all SoS that have a recognized central purpose (i.e., collaborative, acknowledged, and directed). A case study conducted by Walker (2014) with Marine Air Command and Control System (MACCS) provided satisfactory results. MBSE was used in an Approach for Context-based RE (ACRE) (HOLT *et al.*, 2012) for both SoS and CSs. However, it has been successfully applied at the CSs level, and ACRE was modified for satisfying the SoS (HOLT *et al.*, 2015). Figure 5 shows the requirements modeling for SoS in a Use Case Diagram composed of actors (e.g., requirement engineer and standard (e.g., (ISO/IEC/IEEE-15288, 2015)))) and the description of a scenario that presents the ACRE features.

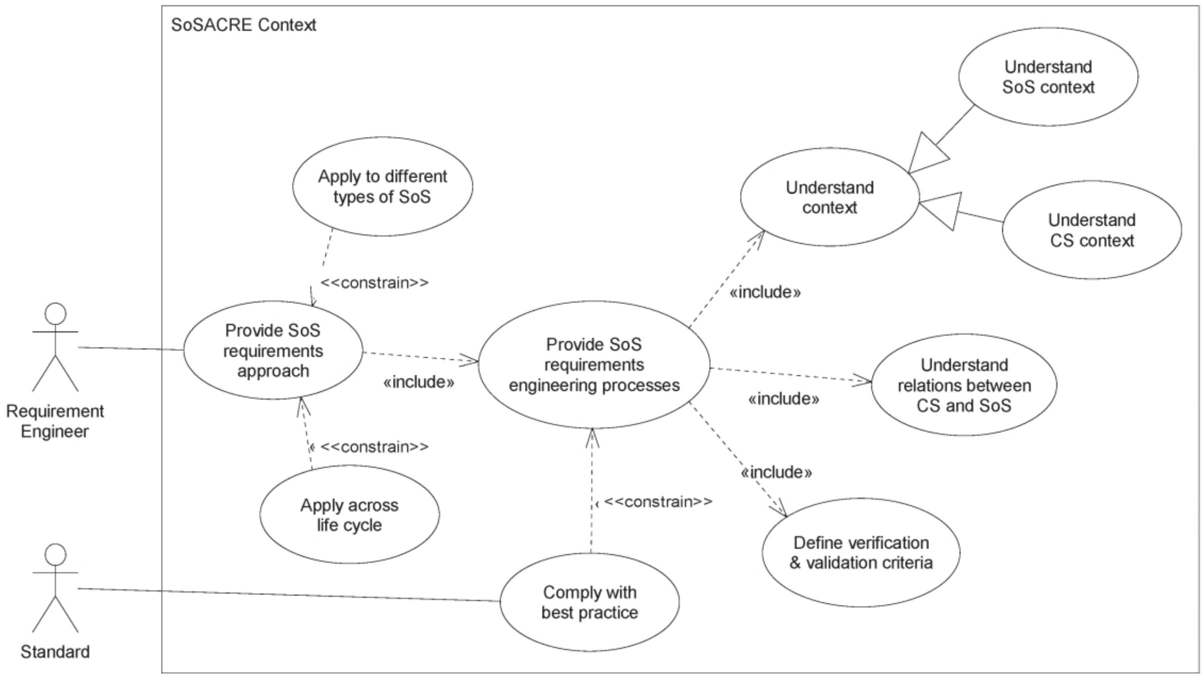


Figure 5 – Requirements modeling in the SoS context

Source: Holt *et al.* (2015).

The success of the SoS RE process requires a combination of top-down and bottom-up approaches. A top-down approach used alone does not consider the needs of the CSs, and a bottom-up one alone cannot capture the key expectations for the SoS as a whole (LEWIS *et al.*, 2009). Top-down aspects include the understanding of the SoS topology, the environments associated with the SoS, and their CSs, and the identification of business goals and interactions among SoS, whereas bottom-up ones aim at understanding the capabilities provided by the CSs and the internal and external constraints imposed on those systems (LEWIS *et al.*, 2009). Such an approach comprehends aspects of two properties of SoS, called, emergent behavior and evolutionary development, and promotes a view of the requirements in the perspectives of SoS and their constituents, thus improving the vision of the impacts caused by changes in both i.e., CSs and SoS (HONOUR, 2013).

Several challenges still remain in RE for SoS, and some were discussed by Lewis *et al.* (2009), Ncube (2011), Hallsteinsen *et al.* (2012a), Ncube *et al.* (2013), Keating *et al.* (2015), and Axelsson (2015b). Lewis *et al.* (2009) addressed the impacts of the RE process, such as scale, multi-domain, varied operational contexts, decentralized control, rapidly-evolving environments, continuous and often disconnected execution of multiple lifecycle phases, and opportunistic needs for collaboration and integration. Similar topics were also discussed by Axelsson (2015a). Keating *et al.* (2015) highlighted some implications for requirements (e.g., holism, complementary, and context dominance) from a SoSE perspective.

2.4.1 Requirements Specification and Modeling Languages of Systems-of-Systems

Towards meeting the increasing complexity of the systems, several techniques have modelled and specified[6] requirements, including their Verification and Validation (V&V)[7], since 1970s. Costs for the correction of errors[8] originated in requirements and that persisted throughout different development phases can scale up to 100 times their original costs (BOEHM, 1981; MAALEM; ZAROUR, 2016). Therefore, RE has adopted informal, semi-formal, and formal techniques, methods, and languages (notations) for requirements modeling and V&V (CHAPURLAT *et al.*, 2006; ARTS *et al.*, 2015). Informal notations are usually more expressive and flexible; however, they also rely on human expertise (WANG *et al.*, 2016). On the other hand, semi-formal languages[9], e.g., i*, Unified Modeling Language (UML), and Systems Modeling Language

[6] The terms modeling and specification are used interchangeably in this article.

[7] We consider V&V of requirements as defined in ISO/IEC/IEEE 24765:2017 (ISO/IEC/IEEE-24765, 2017). **Validation**: confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. **Verification**: the process of ensuring that the software requirements specification complies with the system requirements, conforms to document standards of the requirements phase, and is an adequate basis for the architectural (preliminary) design phase

[8] We consider an error as a human action that produces an inconsistent requirements document.

[9] In the context of this thesis, UML is a language and each of its diagrams is a technique. Therefore, a technique enables the development of a specific structural/behavioral model of a system, whereas a

(SysML)[10], provide a defined syntax, but lack complete semantics to support communication among stakeholders (AHMED; ROBINSON, 2007).

Alternatively, formal languages, such as Vienna Development Method (VDM) and Larch, provide both a well-defined syntax and semantics; they are a set of finite strings of symbols from a finite alphabet (ROZENBERG; SALOMAA, 2004), but they also require considerably more training than semi-formal languages (GUTTAG; HORNING, 1993; WANG *et al.*, 2016). An advantage of defined semantics is that it enables the development of automated tools that can efficiently find problems in requirements (WANG *et al.*, 2016). Several formal languages have been tailored to express requirements (e.g., Requirements Modeling Language (RML) and Knowledge Acquisition in Automated Specification (KAOS)), and enable a precise definition of the objectives of software systems. In this sense, formal languages can support the systematic analysis of formalized statements and their associated impact (NIELSEN *et al.*, 2015a), which reveal missing requirements and inconsistencies, predict behaviors, check the accuracy of requirements, and promote stakeholders' understanding by means of clear semantics (JARKE *et al.*, 2011).

Towards dealing with requirements related to SoS and CSs, RE must change its focus to the right composition of constituents within the SoS, which yields desirable emergent behaviors at runtime (NCUBE *et al.*, 2013). Due to the dynamic nature of SoS, RE is a permanent activity that must be frequently revisited during the SoS life cycle (BOEHM, 1984; BILA *et al.*, 2016). As a result, SoS have a large and complex set of requirements of several interdependencies (EASTERBROOK *et al.*, 1998; WANG *et al.*, 2016), and informal notations are insufficient for modeling such requirements (WANG *et al.*, 2016). Although formal techniques can model requirements related to SoS autonomy, evolution, and emergent behaviors, they usually rely on approaches that lack support for SoS openness and unpredictability (NIELSEN *et al.*, 2015a). The combination of semi-formal and formal techniques can balance the limitations of each approach, e.g., formal techniques model critical parts of the system, and semi-formal ones model non-critical parts (PONSARD *et al.*, 2005; ZHANG, 2014).

2.4.2 Languages and Techniques for Modeling SoS Requirements

Formal and semi-formal languages and techniques that model SoS requirements (LANA *et al.*, 2018) were identified trough a literature review (KITCHENHAM *et al.*, 2010; PETERSEN *et al.*, 2015). Current practices related to specification style and paradigm, as well as their advantages and limitations (e.g., UML, SysML, Prototype Verification System (PVS), and Finite State Machine (FSM)) have been contextualized. Although several SoS belong to critical domains, only a few languages and techniques for the modeling of SoS requirements (e.g., (ATKINSON;

language expresses systems in a structure defined by a consistent set of rules.

[10] SysML is a general-purpose graphical modeling language for representing systems that may include combinations of hardware, software, data, people, facilities, and natural objects (FRIEDENTHAL *et al.*, 2008; Object Management Group, 2016)

CUNNINGHAM, 1991; PONSARD *et al.*, 2005; YUAN *et al.*, 2011; WANG *et al.*, 2017)) are currently supported by automated tools, which probably impacts the accuracy, correctness, and consistency of the requirements.

Such languages and techniques were classified and compared in terms of specification styles, paradigms, and executable syntax, according to the styles described in (MISIC; VELASEVIC, 1997; NASA, 1997; SRIVAS; MILLER, 1995; LAMSWEERDE, 2000; GHEZZI *et al.*, 2002). The most important characteristic of a specification language is based on its mathematical foundation (MISIC; VELASEVIC, 1997). However, different terminologies are used in each study for the same style. For example, the technical report of NASA (NASA, 1997; SRIVAS; MILLER, 1995) classifies languages and techniques as either model-oriented, or property-oriented. The former can be considered a constructive (or prescriptive) style, whereas the latter can be considered a declarative (or descriptive) one, which is also referred to as axiom-based, rule-based, or goal-based (LAMSWEERDE, 2000; GHEZZI *et al.*, 2002). Figure 6 shows the conceptual model created to clarify the classification of specification styles used for requirements modeling and referenced column "Specification Style" (Table 3).

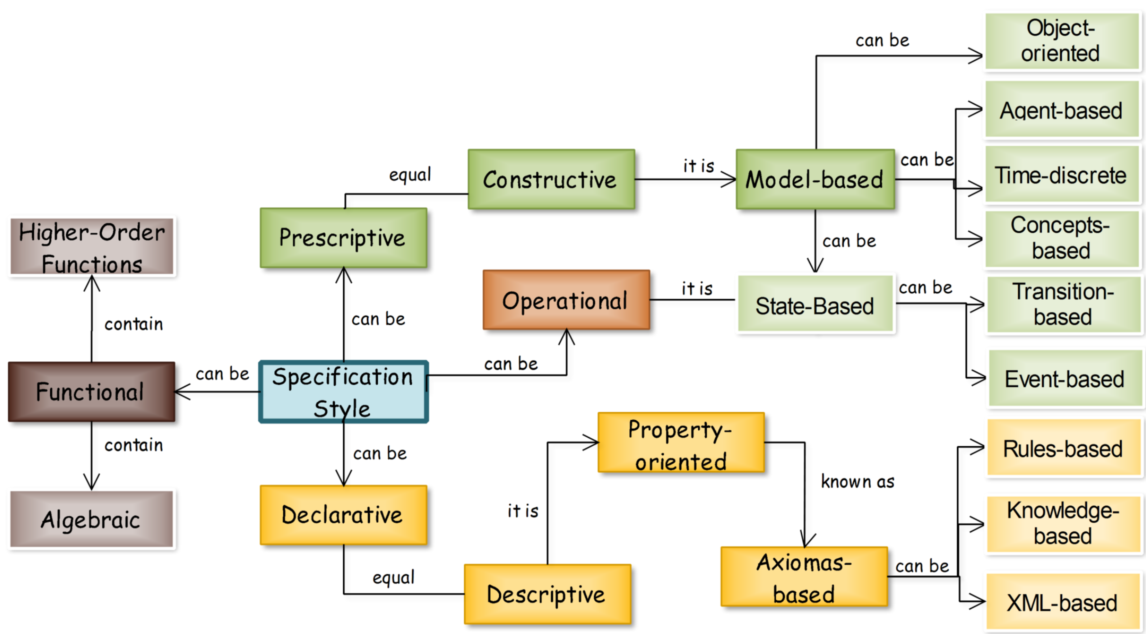


Figure 6 – Conceptual model of specification style terminologies

Source: Elaborated by the author.

A model-based specification, i.e., constructive and prescriptive style, describes the desired behavior for an intended system through detailed models (GHEZZI *et al.*, 2002), but an excess of information can lead to bias in both design and implementation (SRIVAS; MILLER, 1995). Such specifications are often easier for the understanding of non-technical users than property-oriented specifications, which describe desired properties of a system at a higher abstraction level

Table 3 – Classification of formal and semi-formal languages and techniques

Target: DSMSE - Domain-Specific Modeling for Systems Engineering, **LSCS/SoS** - Large-Scale Complex Systems/Systems-of-Systems, **LRTS** - Large Real-Time Systems, **CRTS** - Concurrent and Real-Time Systems, **SoS/LSCCPCS** - Systems-of-Systems/Large Scale Complex Cyber Physical Control System, **LCS** - Large and Complex Systems, **SCS** - Safety-Critical Systems, **SoSSAS** - SoS/Self-Adaptive Systems, **RTES**- Real-Time Embedded Systems, and **LSCS/SoS** - Large-Scale Complex Systems/Systems-of-Systems.

Source: Elaborated by the author.

		Description Languages/Techniques	Specification Style	Paradigm	Graphical Model	Formalism	Language ◎/ Technique ⊙	Executable	Target	Documentation
Model-Based Specification	Object-Oriented	SysML Requirements Diagram - SysML-RD	Prescriptive/Logic-based	Object-oriented	√	S	⊙		DSMSE	√
		UML Activity Diagram - UML-AD				S	⊙		General	√
		UML Class Diagram - UML-CD				S	⊙		General	√
		UML Sequence Diagram - UML-SD				S	⊙		General	√
		SysML Block Definition Diagram - SysML-BDD				S	⊙		DSMSE	√
		SysML Internal Block Diagram - SysML-IBD				S	⊙		DSMSE	√
		Fuzzy Unified Modeling Language - f-UML				S	◎	√	LSCS/SoS	
		SysML				S	◎		DSMSE	√
		Unified Modeling Language - UML				S	◎		General	√
	State-Based	Specification and Description Language - SDL	Prescriptive	State-based	√	F	◎	√	LRTS	√
		Requirements State Machine Language - RSML			√	F	◎	√	Safety-Critical Systems	√
		Z language				F	◎		General	√
		Prototype Verification System - PVS	Prescriptive/Descriptive /High-order Functional			F	◎	√	CRTS	√
	Event-Based	Hybrid Communication Sequential Processes - HCSP	Prescriptive/Operational	Event-based		F	◎	√	Hybrid Systems	√
		Timed Communication Sequential Processes - Timed CSP				F	◎	√	Cyber Physical Systems	√
		Time Petri Nets - TPN			√	F	⊙	√	Real-Time Systems	
	Agent-Based	KAOS Methodology	Prescriptive	Agent-based	√	F/S	◎	√	General	√
		i* Framework			√	S	◎		General	√
	Hybrid	ModelicaML	Prescriptive	Object-oriented/Time-discrete or Event-based	√	F	◎	√	SoS/LSCCPCS	√
		Timed-Automata-Based Fuzzy Cognitive Maps - TA-FCM		Concepts-based/State-based	√	F	⊙	√	Systems-of-Systems	
		Abstract State Machine - ASM		Object-oriented/State-based	√	F	⊙	√	LCS	√
		Finite State Machine - FSM				F	⊙	√	SCS	√
		Statecharts				F	⊙	√	Real-Time Systems	√
Property-Based Specification	Rules-Based	Interelement Requirements Diagram - IRD	Declarative	Rules-based	√	S	⊙		General	
		Fuzzy Logic				F	⊙		SoSSAS	
		eXtensible Markup Language - XML				F	◎	Interpreted	General	√
		Formal Language Maude				F	◎	√	General	√
		Logic First Order - LFO				F	◎	√	General	√
		Modal Action Logic - MAL				F	◎	√	RTES	
	XML-Based	Web service definition language - WSDL	Declarative	XML-based		F	◎	√	General	√
		Web Service Business Process Execution Language - WSBPEL		XML-based	√	F	◎	√	General	√
	Knowledge-Based	Fuzzy Description Logic - f-DL	Declarative	Knowledge-based		F	◎	√	LSCS/SoS	

and perform specifications with fewer details (NASA, 1997; GHEZZI *et al.*, 2002). However, inconsistencies are more easily introduced to this style, and require advanced knowledge to be read and understood by users (NASA, 1997; SRIVAS; MILLER, 1995; GHEZZI *et al.*, 2002). A detailed discussion on the trade-offs between the two specification styles can be found in NASA (1997) and Srivas and Miller (1995).

Table 3 shows other characteristics analyzed (LANA *et al.*, 2018), namely (i) *Paradigm*: classifies a language or technique according to modeling features; (ii) *Graphical Model*: indicates whether a language or technique supports a graphical notation; (iii) *Formalism*: indicates whether a language or technique is formal (F) or semi-formal (S); (iv) *Language/Technique*: indicates whether it is a language or technique; (v) *Executable*: indicates whether models can be executed by the language/technique for requirements V&V towards minimizing time and costs (SAMMI *et al.*, 2010); (vi) *Target*: indicates the types of systems described by the language or technique (e.g., large real-time systems or systems-of-systems); and (vii) *Documentation*: indicates the existence of external information for the language or technique. For example, f-UML is described only in the included studies, whereas Z language is also described by the standard ISO/IEC-13568 (2002).

2.4.2.1 Model-based Specification

The following five model-based specification paradigms were identified: object-oriented, state-based, event-based, agent-based (also known as goal-based), and hybrid i.e.,languages and techniques can be classified into more than one paradigm (e.g., language ModelicaML, and four techniques of statecharts, FSM, abstract state machines, and Timed-Automata-based Fuzzy Cognitive Maps (TA-FCM)). The object-oriented paradigm impacts on systems development, hence, on requirements modeling, and the use of UML and its variants, such as SysML, has been identified. Zhang (2014) combined UML and SysML for ModelicaML, which is derived from UML and SysML, to model the requirements of a cyber-physical SoS. Linhares *et al.* (2007) combined Requirements Diagram, Block Definition Diagram, and Internal Block Diagram of SysML with Time Petri Nets for modeling and model-checking the behavior requirements of a factory plant. UML techniques, specifically Activity Diagram, Class Diagram, and Sequence Diagram, were used by Linhares *et al.* (2007),Ghazel and Koursi (2007), Tang *et al.* (2010a),Tang *et al.* (2010b), and Cimatti *et al.* (2011) for representing requirements. Languages UML and SysML were identified as recurrent for the development of object-oriented systems in the industry. Since these languages and their variants (e.g., executable UML (xUML[11])) lack formal execution semantics for supporting formal modeling, they were also combined with formal approaches (HAIMES, 2012).

Two or more languages and techniques have been combined to model SoS requirements. Leveson *et al.* (1994), Heimdahl and Leveson (1996), Crow and Vito (1998), and Yuan *et al.*

[11] <https://xtuml.org/>

(2011) combined formal languages and formal techniques; Ponsard *et al.* (2005) and Krishna *et al.* (2009) combined formal languages and semi-formal techniques; Linhares *et al.* (2007) and Tang *et al.* (2010a) combined formal and semi-formal techniques; and Zhang (2012), Zhang (2014), and Zou *et al.* (2014) combined formal and semi-formal languages. Yuan *et al.* (2011) employed a formal language (Specification and Description Language (SDL)) and a formal technique (FSM) in a method for the modeling and verification of systems requirements of a train control system. Therefore, we can infer SoS requirements modeling is not completed with the use of a single formal language or technique; instead, a set of languages and techniques is required for properly dealing with the modeling challenges posed by each domain. More studies are necessary for investigations on SoS requirements modeling in other domains, due to the different challenges that might be faced, and evaluations of the formal and/or semi-formal languages and techniques that provide better support for SoS inherent characteristics.

Two languages, called i* framework and KAOS methodology support the agent-based paradigms in industrial environments (GOLDSBY *et al.*, 2008; PICCOLO *et al.*, 2015). Agents can be defined as active components that represent people, devices, legacy software, or software-to-be, and fulfill specific requirements and expectations (RESPECT-IT, 2007). i* framework is a conceptual modeling language adopted by Goldsby *et al.* (2008), Krishna *et al.* (2009), and Zhang (2014) for modeling a system-to-be and critical modeling decisions, such as identification of the main system's goals, representation of multiple stakeholders and their interdependence, and possibilities for exploring the use of such relationships (KRISHNA *et al.*, 2009; YU, 2011). These characteristics can be interesting for SoS requirements modeling, since SoS are formed by interacting independent constituents that individually address a particular set of stakeholders and goals (BENDOV, 2009; MAIER, 1998; HAN; DELAURENTIS, 2013). KAOS methodology, adopted by Ponsard *et al.* (2005), represents goals that express system properties stated by stakeholders and met by agents (RESPECT-IT, 2007). In general, the language supports the following four techniques: (i) goal model, which forms a set of interrelated goal diagrams that, together, address a particular problem; (ii) object model, which describes objects (e.g., agents, entities, and relationships); (iii) responsibility model, which describes the requirements and expectations an agent is assigned to; and (iv) operation model, which describes all behaviors agents must exhibit to fulfill their requirements (PONSARD *et al.*, 2005; RESPECT-IT, 2007). KAOS assists in the establishment of both formal and semi-formal models. Semi-formal models are based on texts and include graphical representations, whereas formal models are built on top of semi-formal models, either partially, or entirely (RESPECT-IT, 2007).

The state-based paradigm can be considered in the behavioral paradigm, since it represents a system's behavior by transitioning among different states. In critical systems, models depicting behavior diagrams, such as FSM, and sequence diagrams are employed not only in the modeling of requirements for safety-critical systems, but also in their verification through simulations. Simulations traditionally evaluate execution scenarios at design time and, more recently, SoS (HONOUR, 2013; NETO *et al.*, 2014). SoS simulation is a difficult task, due to a combination of factors, such as performance issues, conflicting goals, standards, and emergent behaviors (which might be known or unknown at design time) (HONOUR, 2013; ZEIGLER; NUTARO, 2015). However, it offers important benefits, such as early identification of errors and problems, which can be corrected prior to the actual realization of the SoS (NETO *et al.*, 2014). Linhares *et al.* (2007), used PVS, a language with a tool and theorem-proof integrated with decision procedures for different theories, including real and integer arithmetic, in conjunction with an abstract state machine and TA-FCM. The authors investigated the formalization of the subsystem modeling of NASA's Space Shuttle using PVS to explore and document the feasibility and utility of formalizing critical Shuttle software requirements that represent a spectrum of maturity levels. PVS has still been used at NASA's Langley Research Center (LaRC) for the modeling of requirements of aerospace applications, such as pilot flying specification (COFER *et al.*, 2014), and aerospace verification tool (WAGNER *et al.*, 2017).

Although PVS has been classified as a model-based style (i.e., state-based paradigm), it can also be considered a property-based one (i.e., an axiom-based paradigm), since a set of properties is described to ensure consistency of the specification (SRIVAS; MILLER, 1995; NASA, 1997). PVS considers higher-order logic or higher-order function, classified as a functional style (LAMSWEERDE, 2000). The same author classified Time Petri Net as an operational style, while others classified it as a model-based style (SRIVAS; MILLER, 1995; NASA, 1997; GHEZZI *et al.*, 2002).

2.4.2.2 Property-based Specification

The following three property-based paradigms were identified: (i) rule-based, which uses formal and semi-formal languages and techniques, (ii) XML-based, which uses formal languages, and (iii) knowledge-based, which uses formal languages. Despite the smaller number of property-based paradigms, they have been used in SoS requirements modeling (CORDES; CARVER, 1988; NGUYEN *et al.*, 2014; ABDALLA *et al.*, 2015; COFER *et al.*, 2014). All property-based languages identified, i.e., XML, WSDL, WS-BPEL, FOL, Maude,Modal Action Logic (MAL), and fuzzy Description Logic (f-DL), and property-based techniques, called Interelement Requirements Diagram (IRD) and Fuzzy Logic were occasionally applied separately by each study. Wang *et al.* (2017) adopted the Description Logic ontology, which is a logical reconstruction of a frame-based knowledge representation language for the description of quality requirements. Towards the specification of functional and non-functional SoS requirements considered fuzzy and vague at the mission level, an extension of f-DL, called f-SHIN, described the necessary quality of the requirements, due to its strong abilities for representation and decidability. f-DL can formalize the UML model and provide an algorithm that converts a fuzzy UML (f-UML) model into the f-DL ontology and automates verification.

Han *et al.* (2016) focused on the automation and verification of safety requirements based on pattern-based specification. The requirements were specified by First Logical Order

(FOL), a rule-based paradigm, for the verification of safety requirements in the railway domain automatically through model checking. Due to the characteristics of SoS, the creation of an environment for requirements verification is a challenging task. Each constituent has its own verification responsibility; therefore, changes in constituents may result in verification events that might affect the entire SoS (HONOUR, 2013). Moreover, the managerial independence of the constituents often does not enable the synchronization of multiple life-cycles, and the SoS requirements and the SoS itself are occasionally checked without the presence of all constituents of an SoS (i.e., without all capabilities) (LEWIS *et al.*, 2009; LUNA *et al.*, 2013b). Due to the evolutionary development of SoS, requirements modeling and verification should be performed continually, and include the evaluation of the system's capability regarding its missions (DAHMANN *et al.*, 2010; HONOUR, 2013).

2.4.3 Requirements Analysis for Systems-of-Systems

Requirements are descriptions of the capabilities required for solving a problem towards the achievement of a mission goal. They may be considered a high-level abstract statement of a system's functions or operational constraints and reflect the needs and expectations of stakeholders in designing the system (MOHAPATRA, 2010). Requirements Analysis involves the decomposition or refinement of the needs of multiple stakeholders in capabilities and low- level requirements that are often met by the constituents. The use of different views is an old practice of requirements engineers for ensuring all stakeholders can be understood and examine the requirements to find errors, omissions, ambiguities, conflicts, and other deficiencies (ANTHONY, 2016). Validation improves the quality of the requirements and solves issues that emerge in the analysis phase. For instance, requirements can be raised, since stakeholders do not clearly declare their intentions at the beginning of a project. Undetected mistakes at this stage may require time and other resources and are more difficult to be corrected (BOEHM, 1981; MAALEM; ZAROUR, 2016).

The requirements analysis of complex systems (e.g., SoS) is tightly intertwined with their architecture. Requirements and architecture thinking turns into further fuzzy at runtime than other types of large software projects (WIEGERS; BEATTY, 2013). Therefore, capability and requirements must be specified for the SoS architecture element(s) in the architecture projected at design time, or adapted and evolved at runtime. The analysis activity must be rationally adopted and used in a crosscutting way for the correct identification and exploration of each level of mission necessary for decomposition, ensuring the iterative and incremental process between problem space and solution space can be solved (SCHMIDT, 2013; WIEGERS; BEATTY, 2013). The literature on monolithic systems (i.e., traditional systems) has discussed the interleaving between requirements activities and software design (NUSEIBEH, 2001; CLELAND-HUANG *et al.*, 2013; CHEN *et al.*, 2013; GALSTER *et al.*, 2014) - this interaction was addressed for SoS in Chapter 4.

Among the challenges of analyses of SoS requirements faced by the mission and the SoS team is the sharing of a final vision of the system with the sponsor and other stakeholders. Towards minimizing it and dealing with the increasing complexity of the systems, approaches have been built since the 1970s for modeling and specifying requirements from the conceptual entity-relationship model to structured, object-oriented, use case and goal-oriented approaches (AMYOT *et al.*, 2010). The current use of concepts, such as agents, goals, objectives, and actions rather than functionality, methods, and procedures, makes the software modeling more flexible for dealing with the inherent complexity of the applications (LEWIS *et al.*, 2009). Examples of object-oriented approaches are SysML and UML (Object Management Group, 2018a; Object Management Group, 2018b), scenario (BASS *et al.*, 2012), and GORE, which has been widely adopted by academics and industry for goals modeling (RESPECT-IT, 2007; YU *et al.*, 2013).

GORE considers alternative solutions, enhances the analysis of the requirements completeness,[12] and guides traceability from an organizational context (YU, 2011). The use of metaphors and goals reflects the intention of organizational actors and approximates the real world of organizations where the software is inserted. Goals support the elaboration of requirements providing a rationale for a refinement from high-level strategic goals to low-level technical goals applying AND/OR, so that quality attributes can be addressed and treated in several dimensions (LI, 2016). They facilitate the comprehension of the mission by the SoS team and mission team towards meeting the stakeholders and organizational needs, in the early phases of SoS development (YU, 1995; DAHMANN, 2019).

2.5 Software Architecture of Systems-of-Systems

Software architecture is a topic of growing concern within academic and industrial communities, considered the backbone of the success of software-intensive systems and plays a fundamental role in their quality (KAZMAN *et al.*, 1994; CLEMENTS *et al.*, 2011; JAMSHIDI, 2009). It is a system abstraction related to the fundamental organization of the software and specifies its components, relationships, and environmental principles that involve a project and its evolution (BASS *et al.*, 2012). Therefore, architectural decisions directly impact on the achievement of business goals and functional and quality requirements (BASS *et al.*, 2012), and regarding the use of the architecture, models describe and analyze a system's behavior (CROWDER *et al.*, 2016). The architecture of systems has been created through a body of knowledge that meets specific missions or business needs according to a set of given requirements or capabilities (CROWDER *et al.*, 2016).

Systems have become larger and their capabilities have increased; however, the development of an architecture that meets and manages the SoS needs (e.g., mission, requirements, performance, metrics, and quality attributes) has been a big challenge (CROWDER *et al.*, 2016).

[12] A requirement specification is complete if all the goals can be accomplished from the specification.

SoS have multiple stakeholders with different and conflicting intentions that require negotiation and trade-off analysis. Figure 7 shows a relation of dependence (i.e., effects) between each architecture driver and the SoS architecture. `SoS architecture` is the core and covers the characteristics addressed by Crowder *et al.* (2016). `Quality attribute` can be affected by `system complexity` and affects the `functional behavior`, whereas the `emergent behavior` helps the definition of `systems complexity` and is directed by `functional behavior`.

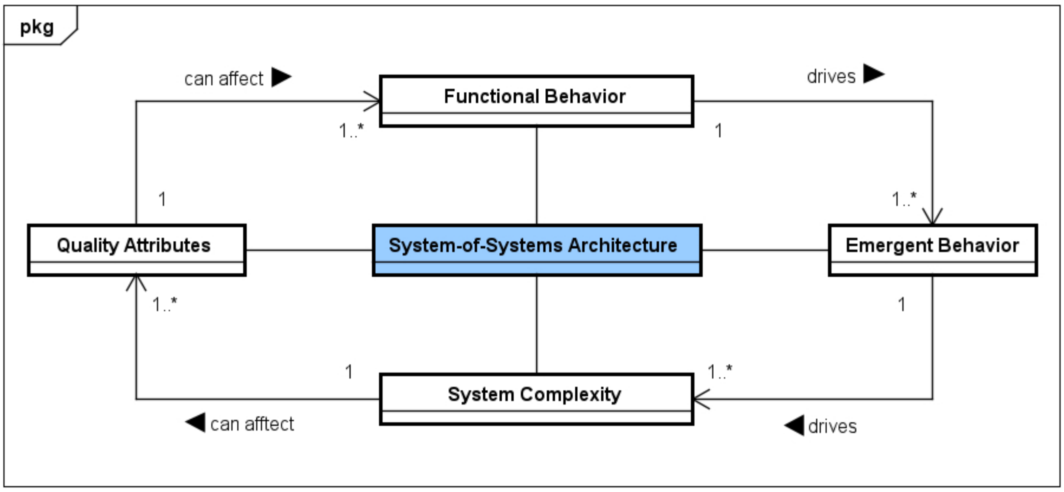


Figure 7 – Systems-of-Systems architectural characteristics dependence

Source: Adapted from Crowder *et al.* (2016).

Few initiatives have been designed to support the development of an SoS architecture that addresses their main characteristics. Acheson (2010) proposed an object-oriented approach based on DODAF framework with a focus on flexibility and evolution of the SoS, and reuse of its objects (systems and subsystems), and Chigani and Balci (2012) developed a process of architecting a software-based network-centric SoS based on DODAF and Capability Maturity Model Integration for Development (CMMI-DEV) at maturity level 3. It covers the main phases of the SoS architectural development. Cavalcante (2016) designed a formal framework to model the SoS architecture considering its property of dynamic architecture and the necessity of its automatized verification, since the adaptation occurs at runtime. Guessi (2017) presented a dynamic SoS architecture considering the constraint-satisfaction problem that is automatically analyzed into initial properties that can be revisited and refined over time. Other topics regarding the development of SoS involve Model Driven Architecture (MDA) (GOKHALE *et al.*, 2008a; NETO *et al.*, 2014), agile methods (FARROHA; FARROHA, 2011a), and reference architecture (RODRIGUES, 2018). International standard ISO/IEC/IEEE-42010 (2011) was proposed for supporting the architectural description, and comprehends specified requirements on Architecture Description Languages (ADL) (e.g., ACME (MEDVIDOVIC; TAYLOR, 2000), π-ADL (CAVALCANTE *et al.*, 2014), UML (Object Management Group, 2018b), and SysML (Object Management Group, 2018a)). The use of formal languages is interesting for SoS, since most

SoS have been built for critical domains (e.g., military, aerospace, and automotive); however, informal descriptions still have been widely found (CLEMENTS *et al.*, 2011). Another support is architectural frameworks (e.g., DODAF (OFFICER, 2010) and MODAF (PARTNERS, 2005)), which establish common practices for the creation, interpretation, analysis, and use of architecture descriptions in a particular domain of application or stakeholder community (ISO/IEC/IEEE-42010, 2011).

Shanmugapriya and Suresh (2012) compared several methods of architectural evaluation discussed by Santos *et al.* (2014) in four main groups, namely (i) simulation-based methods, which evaluate the performance and accuracy of architectures, (ii) descriptive modeling methods, which adopt mathematical proofs for evaluating operational quality requirements (e.g., performance, interoperability, and adaptability), (iii) experience-based methods, which rely on previous experience and domain knowledge of consultants or developers, and (iv) scenario-based methods, which evaluate the quality requirements for describing specific scenarios that help risk analysis and consequences in a scalable way. Due to constants changes, SoS architecture must be frequently evaluated; the most applied methods for such an activity are simulation (NASA, 1997; COFER *et al.*, 2014; MAIER, 2014; MITTAL; RAINEY, 2015; TOLK; RAINEY, 2015; DIALLO *et al.*, 2018), Scenario-Based Architecture Analysis Method (SAAM) (KAZMAN *et al.*, 1994), and Architecture Trade-off Analysis Method (ATAM) (KAZMAN *et al.*, 1998; URWIN *et al.*, 2010). Such changes are usually environmental and developers have little or no control over them by either expanding functionality, or improving performance, associated with two of the main SoS characteristics, called evolutionary development and emergent behavior (SELBERG; AUSTIN, 2008; MENS *et al.*, 2010; MITTAL S.AND DIALLO; TOLK, 2018). Therefore, SoS architectures must be prepared to support dynamic evolution and incorporate new functionalities, or remove those that affect the achievement of the SoS mission (JAMSHIDI, 2009; NAKAGAWA *et al.*, 2013). Selberg and Austin (2008), Mens *et al.* (2010) and Cavalcante (2016) discussed not only the architecture's dynamics and evolution, but also the evolution of architectural artifacts, especially description, preserving its initial purpose and criticality concerns. Despite the research conducted so far, the SoS architecture has not been deeply investigated, and important topics, such as complete and integrated processes that analyze, design, represent, evaluate, and evolve the SoS software architectures still require special attention.

2.5.1 Interoperability of Systems-of-Systems

Interoperability is a critical requirement in the SoS development, considered a crosscutting one at different levels and degrees (CHEN; VERNADAT, 2002; FORD *et al.*, 2007a; REZAI *et al.*, 2014). The most common types of interoperability are semantic, syntactic, technical, and organizational (KUBICEK *et al.*, 2011). Semantic interoperability is regarded as the capability of the CSs to comprehend the meaning of exchanging information (COMMUNITIES, 2004). Problems at this level emerge when a message of requisitions between two or more CSs is described through different conceptualizations or representations of the entity types, properties, and values from their subject domains (MACIEL *et al.*, 2017). For instance, CSs may diverge on similar concepts, but different definitions are exchanged. Syntactic interoperability deals with the format of data facilitating their exchange between two or more CSs for fulfilling an activity (KUBICEK *et al.*, 2011; MACIEL *et al.*, 2017); it concerns the compatibility of the format of data exchanged between CSs towards promoting their communication (COMMUNITIES, 2004). The data are described with syntactic elements and rules previously established by a grammar. Thereby, problems are related to the incompatibility of the rules used by the CS transmitter and another CS that receives the data (MACIEL *et al.*, 2017). Technical interoperability ensures the infrastructure of hardware/software, systems, and platforms that link the computer systems and services. Such an interoperability type is usually associated with communication protocol, interfaces, mediators, architectural styles, and all infrastructure necessary for machine-to-machine communication (KUBICEK *et al.*, 2011; REZAI *et al.*, 2014). Finally, organization interoperability is related to the communication of organizations and collaboration within and between them, thus ensuring business collaboration (e.g., contracts, ownership, and market structures), and also legal collaboration (e.g., regulatory structures and requirements, and protection of physical and intellectual property); it requires the establishment of the three collaboration types (WESTHEALTH, 2010). Faster technological advances facilitate deficiencies in interoperability that cause problems of incompatibilities for independent systems working together for fulfilling a greater mission (KASUNIC, 2001). The need of a quick access to information and improvements in both speed and accuracy of prioritization and transferring data have demanded efforts towards reducing interoperability problems (KASUNIC, 2001). Frameworks, methodologies, and models have been designed since 1980 for improving the measurement and evaluation of interoperability (KASUNIC, 2001; CHEN; VERNADAT, 2002; FORD *et al.*, 2007a) and quantifying results in a visible way, while maintaining the focus and setting right priorities. Changes must be managed and their impact and the whole SoS must be measured (KASUNIC, 2001; FORD *et al.*, 2007b; OFFICER, 2010).

Over 20 interoperability models are available to support the measurement and evaluation of interoperability. They have specific aims often separated by levels of interoperability and/or maturity. The first model, Spectrum of Interoperability Model (SoIM), was designed by La Vean, in 1980. It comprehends seven levels, which range from different systems at the lowest level, to the same system at the highest level of interoperability (VEAN, 1980; WYATT, 2014) for evaluating three mission areas, called surveillance, over-the-horizon targeting, and electronic warfare and quantifying seven interoperability components i.e., media, languages, environment, requirements, human factors, procedures, and standards. Mensh *et al.* (1989) developed the Quantification of Interoperability Methodology (QoIM) for correlating interoperability to measurements of its effectiveness through simulation. In 1996, Military Communications and Information Systems Interoperability (MCISI) (AMANOWICZ; GAJEWSKI, 1996) was proposed, followed by Levels of Information System Interoperability Model (LISI) (OFFICER, 1998), which, despite resembling SoIM, defines, evaluates, measures, and assesses information systems' interoperability. In the same year, (LEITE, 1998) designed Interoperability Assessment Methodology (IAM), which resembles QoIM, while Organizational Interoperability Maturity Model for C2 (OIM) (CLARK; JONES, 1999) is an extension of the LISI model. The importance of identifying compatibility among the legacy system and other systems was also treated in the interoperability model in 2002 in the military domain, when the Stoplight model (HAMILTON *et al.*, 2002) was developed to support decision-makers to understand whether or not their legacy systems could meet operational and acquisition interoperability requirements of the SoS (e.g., C4ISR systems). Similarly to LISI and OIM, the Levels of Conceptual Interoperability Model (LCIM) (TOLK; MUGUIRA, 2003) was proposed as an alternative for connecting the conceptual and technical designs for implementation, integration, or federation (TOLK; MUGUIRA, 2003). It contributes to the achievement of meaningful interoperability and improves the standardization of V&V activities even as the description process of systems developed to be federated. Tolk (2003) also designed the Layers of Coalition Interoperability (LCI); it is comprised of nine interoperability layers and shows, through a reference model, the existence of an unbroken line between organization interoperability and technical interoperability. The highest layer is "political objectives layer" of the organization interoperability, which has an interface with technical interoperability through the application of the "knowledge/awareness layer", and the lowest layer is physical interoperability, which belongs to operational interoperability. NATO C3 Technical Architecture Reference Model for Interoperability (NMI) was proposed in 2003 and revised in 2017 (NATO, 2003; NATO, 2017). It is similar to LISI, but with only four layers. Non-Technical Interoperability Framework (NTI) (STEWART *et al.*, 2004) and Organizational Interoperability Agility Model (OIAM) (KINGSTO *et al.*, 2005) are based on the OIM organizational model, whereas the Layered Interoperability Score (i-Score) (FORD *et al.*, 2007b) is based on mathematical methods that measure the interoperability of all types of systems for a very specific operational scenario/thread (BILLAUD *et al.*, 2015). The System-of-Systems Interoperability Model (SoSI) proposed by Morris *et al.* (2004) and Connection, Communication, Consolidation, Collaboration Interoperability Framework (C4IF) do not contain specific metrics to quantify interoperability within an SoS. SoSI (MORRIS *et al.*, 2004), developed by Carnegie-Mellon University Software Engineering Institute (CMU-SEI) to support SoS interoperability research, introduced three types of interoperability (i.e., programmatic, constructive, and operational), and C4IF proposed a typology for interoperability information systems (PERISTERAS; TARABANIS, 2006). Information Systems Interoperability Maturity Model (ISIMM) was designed for standardizing and evaluating technical interoperability in governmental environment (STEFANUS; JAMESON, 2012).

2.5.2 Technical Interoperability

Technical interoperability refers to the interoperability of infrastructure and enables the exchange of data among components or individual systems translated into information to be used by another one or end-user (MICHENI *et al.*, 2014). Its aspects are associated with hardware, software, network, communication protocol, standards, and equipment for connectivity among machines (KUBICEK *et al.*, 2011), including requirements, such as open interface, connectivity, data integration, middleware, data presentation, data exchange, accessibility and security issues (VEER H; WILES, 2006). Regarding data in the SoS paradigm, configuration data, online data, and run-time data must be differentiated (MONOSTORI *et al.*, 2016). Data configuration is usually performed during the design of the systems and describes their physical parts. Online data are generated between data configuration and runtime data, since the necessary data are available, but not accessible in real-time for entire systems. Runtime data are established in the operational phase and describe the current status of the mission. A better synergy is achieved if each data type can share the same models, which requires the use of standardization at design and operational time.

The maturity level of technical interoperability is the first to be guaranteed for the processing of interoperation of individual systems (TOLK; MUGUIRA, 2003; KUBICEK *et al.*, 2011). Many users are familiar with agreements at the technical level of the interfaces, where physical and logical connections are established among systems, and communication and data are transported to multiple systems and networks. However, human information and descriptions are inserted in entities related to data and information or the way they are related to similar entities across different mission domains (MATER, 2009). Moreover, layout and individual systems are interconnected, i.e., the way each entity is related to others regarding computation and decision-making. The traditional control system is hierarchical and deterministic, and operates with centralized decision-making and data flow only vertically, between the child and parent nodes (MONOSTORI *et al.*, 2016). The SoS paradigm of the new generation has assumed a combination of decentralized and centralized approaches, in which data can flow vertically and horizontally between platforms (e.g., SoS based on the Internet of things (IoT) (HERSHEY; TALBOT, 2019)).

Interoperation is driven by the need for a mission that supports information sharing. The business process enables the essential exchange of information. At a higher level (e.g., pragmatic level), interoperability demands agreement on the business process interoperation that is expected to consider an interface, and on the interoperation of several individual systems. Such an agreement would express the service or microservice request, minimal requirements to such services to be met, and responses that support a larger process, which is shared by the organizations in collaboration with the operational environment (MATER, 2009). However, those processes must evolve and be consistent with both tactical issues of running the business and strategical issues related to the political and regulatory environment, besides sharing the mission of the involved parties.

2.6 Related Work

To the best of our knowledge, no study on the analysis of technical interoperability requirements has addressed mission engineering for the design of the substantive theory. Nonetheless, this section reports some studies on SoS mission, mission engineering, and SoS interoperability.

Mokhtarpour and Stracener (2017)) proposed a conceptual methodology for the selection of the right coalition of constituents towards the formation of an SoS. It starts with acquisition and mission requirements, and then the preferred candidate is selected. The methodology aims at obtaining the appropriate SoS configuration identifying the amount and type of constituents to be selected from existing or proposed ones. Garcés and Nakagawa (2017) designed a systematic process that establishes, models, and validates SoS missions, and incorporates them into reference architectures. The identification of the models was based on Silva *et al.* (2015a), and provides more abstraction to the reference architecture. The process was evaluated in an e-health system of a chronic disease. Silva *et al.* (2015a) proposed a mission modeling language, called mKAOS, which is based on a conceptual model for SoS missions and supports the specification of missions and their relationships, as well as several other aspects of the SoS, regardless of its implementation details. Silva *et al.* (2015b), designed an open-source tool for modeling SoS missions using mKAOS language. The tool was integrated with Eclipse and developed through open-source modeling frameworks. Kumar and Merzouki R.and Bouamama (2017) also developed a generic modeling of an SoS class, called mechatronic systems, using a bond graph modeling approach to evaluate behavioral and organizational aspects of systems. The generic model was simulated for intelligent transportation systems.

DANSE project (ETZIEN, 2014) addressed the formal SoS modeling based on agents through the establishment of an ontology with key concepts formalized by the graph theory and then mapped to the Unified Profile for DODAF and MODAF (UPDM), a framework that supports the modeling of SoS architecture. Similarly, Zhu *et al.* (2017) discussed key aspects related to SoS mission modeling and analysis using an ontology. They proposed a mission decomposition method composed of patterns, principles, and algorithms and translated into the OWL DL ontology for the analysis and validation of the rules. The method was evaluated by a case study in the application domain of an airport.

From the perspective of mission engineering, Sousa-Poza (2015) discussed the difficulties in managing and governing complex systems, besides the needs of process and management for work in complex situations. New knowledge related to mission engineering was generated towards maintaining coherence among systems engineering, operations, and the mission. Ricci *et al.* (2014) presented the SoS Architecting Ilities (SAI) Method, which enables systems architects to develop for ilities considering the conceptual design phase until the final selection of the SoS architecture, and the fostering of value sustainment throughout the systems' lifecycles. Wertz *et al.* (2015) created a process of modeling space mission engineering that supports the entire phase of analysis of mission end-to-end and design of mission architecture, and understands the need of both field and project. The acquisition for each external field and its station according to the mission project is an essential step for the identification of the necessary resources. Vesonder *et al.* (2018) conducted a survey with mission engineers to understand the way they worked with the mission in different domains and the difference between mission engineering and SoSE for other disciplines. They proposed a mission engineering competency framework, however, the competencies overlapped systems engineering competencies. The competences of mission framework are discipline & domains foundations, mission concept, system engineering skills, systems mindset, interpersonal skills, and technical leadership, and support mission and systems engineers towards a better understanding of the development process of both systems and personal skills. Dahmann (2019) adapted the wave model (DAHMANN *et al.*, 2017) and designed general steps for mission engineering to be addressed in SoSE, since it is considered more dynamic than mission analysis in SoSE. Therefore, the SoSE process in the initial phase must be changed and more attention is required during the whole SoS lifecycle for the identification of mission threats.

Mallek *et al.* (2012) analyzed the interoperability requirements at the enterprise level and proposed a model-based approach for the detection of interoperability problems through several formal techniques according to the abstraction level of the requirement. The needs were identified and, after formalization, organized into a repository of interoperability requirements where verification is performed towards quality improvements. Hogie *et al.* (2005) discussed approaches for standard Internet technologies to meet the communication needs of future space missions. The aim was to support the selection of appropriate internet protocols that assist communication in space and maintain interoperability with the terrestrial Internet. The approach developed used a layered architecture, initiated with low-level physical data link, and data routing issues. After the low-level had been described with basic communications, the transport protocols and applications were selected towards achieving various mission data delivery needs. The protocols were implemented and the spacecraft and ground system were assessed. Chen and Daclin (2006) described the framework for enterprise interoperability, which was designed in the INTEROP NOE project. It supports the identification of the main dimension of interoperability, and has three dimensions,i.e, enterprise levels, which address business, processes, services, and data, interoperability barriers, and interoperability measurement (i.e., compatibility measures and performance measures), which support the identification of interoperability knowledge improving the understanding of the enterprise's interoperability problems. Particularly, barriers (i.e., conceptual, organizational, and technological) hinder interoperation among enterprises. Qureshi *et al.* (2010) developed a framework for eliciting and analyzed interoperability requirements based on user's needs for variability dimensions using the goal approach, which supports the framework with intentional and process views. The former analyzes the user's goals and preferences, whereas the latter analyzes the operationalization of the goals. Both views promote a comprehensive analysis at different levels and elicit relevant requirements.

This research differers from the aforementioned studies in regarding the following aspects: (i) it establishes a theory and concepts based on empirical information, (ii) presents a triplet model that gathers technical requirements from the activities of mission engineering, (iii) uses an industrial approach to identify the SoS mission and decomposes it with important benefits, followed by flexibility, transparence, engagement, alignment, focus, and accountability, and (iv) it addresses the way technical interoperability can impact the SoS mission.

2.7 Final Remarks

This chapter presented the main concepts related to SoS and its mission-driven development, RE, and Software Architecture field relevant to this research. The SoS concept was addressed together with its evolution, and their relevant characteristics and properties were described. The mission-driven SoS development was discussed in two dimensions: (i) mission analysis and (ii) mission engineering. Issues related to RE for SoS and the way the requirements were specified and modeled and languages and techniques were applied in the requirements modeling was discussed. The challenges faced by SoS and mission team were also addressed. Moreover, SoS architecture issues and the main concepts and challenges of interoperability (particularly those related to technical interoperability) were described, and some related work on mission, mission engineering, and SoS interoperability was discussed. The next chapter presents the methodological procedures applied to this research.

CHAPTER 3

Methodological Procedures

3.1 Initial Remarks

SINCE the 2000s, several studies have focused on the application of qualitative methods to software engineering (SEAMAN, 2008; DYBå *et al.*, 2011; MILES *et al.*, 2014). Among such methods, the grounded theory has excelled (CARVALHO *et al.*, 2003; CHAKRABORTY; DEHLINGER, 2009; HALAWEH, 2012; MILES *et al.*, 2014; WüRFEL *et al.*, 2016; GIDEY *et al.*, 2017). According to it, data can be obtained from several sources (e.g., interviews, surveys, case studies, documents, observations, and others), and their collection and analyses are performed in a systematic, interactive, and incremental fashion, with knowledge generated from a joint construction between interviewer and interviewee of a studied domain (GLASER; STRAUSS, 1967; STRAUSS; CORBIN, 1990).

Grounded requirements engineering is a qualitative method that supports the acquisition, analysis, and constant validation of requirements by stakeholders (URQUHART, 1997). Therefore, requirements from the architectural design phase are more valuable to user-end and customers (HALAWEH, 2012). Due to the increasing complexity of software systems, a success requirements engineering end-to-end goes beyond the understanding of the needs of users, customers, and other stakeholders, and the contexts in which the developed software-intensive systems will be used (CHENG; ATLEE, 2007). Consequently, the understanding of the alignment and misalignment from strategical to technological layers and between SoS mission and the software system architecture level that satisfies the requirements is mandatory.

This chapter describes the methodological procedures for the identification of the categories of the development of substantive theory, and addresses the motivation for the choice of grounded theory as a qualitative analysis method, as well as the steps of empirical and literature-driven analyses. Section 3.2 focuses on the research characterization, and Section 3.3 explains and discusses the grounded theory method. Section 3.4 describes the research design planning, and the data acquisition procedure is addressed in Section 3.5. Sections 3.6 and 3.7 are devoted to the development of the research and a comparison with relevant literature, respectively. Section 3.8 introduces the research assessment criteria, and Section 3.9 provides the final considerations.

3.2 Research Classification

This research uses an exploratory-descriptive investigation planned from outlines of qualitative research and applied and non-applied research types employing procedures that involve bibliographic, documentary and field research for the creation of the substantive theory. Exploratory research aims at discoveries in an area of little accumulated and systematized knowledge towards supplying greater affinity with the problem for representing a situation or phenomenon (GIL, 2010; YIN, 2018). Regarding the type applied, we investigated interoperability requirements over time for finding practical problems and contributing to the progress of the SoS design (BICKMAN; ROG, 2009). The use of such a type reduces the distance between the empirical knowledge and the approaches established by software engineers (ADOLPH *et al.*, 2011). Regarding the non-applied type, we addressed the mission engineering discipline due to the few applications related to it in the SoS domain. Qualitative research regards the construction of knowledge through constant interactions between interviewer and interviewee towards solutions to problems (MILES *et al.*, 2014; BAIMYRZAEVA, 2018). It is related to a systematic data acquisition and provides new insights for further ones. Therefore, our aim is to produce practical knowledge through the identification and analysis of interests and needs of SoS stakeholders that influence the decision-making process.

Current software engineering research on grounded theory is associated with positivism and interpretivism approaches (STOL *et al.*, 2016). Nevertheless, we have maintained the core pragmatism with multiple data, such as empirical analyses, literature review, and debates with experienced specialists (REMENYI, 2014), and used this methodology as a base for achieving the thesis' objective. The core pragmatism also complies with the statements of Wohlin *et al.* (2015) and Hazzan (2010), according to which learning about a new context, situation, and process cannot be fully analyzed only by quantitative methods. Figure 8 illustrates the characteristics of the three types of *Grounded Theory* methods, i.e., the original one, proposed by Glaser and Strauss (1967), its variance, established by Strauss and Corbin (1990), and the one designed by Charmaz (2005). The light green color refers to the theory adopted for the development of this research, which initiates with a question and a preliminary literature review. The grey color shows the approach does not comply with this research. The reason for this choice is consistent with considerations of Benbasat *et al.* (1987), Coleman and O'Connor (2007), and Polacsek *et al.* (2018). Grounded Theory is suitable for research in non formally studied domains, or those where a new viewpoint might be beneficial (ADOLPH *et al.*, 2011). Its use enables representations of the studied field, facilitating the comprehension of those involved (CHARMAZ, 2014). Therefore, this approach was selected due to our interest in understanding issues relevant to the analysis

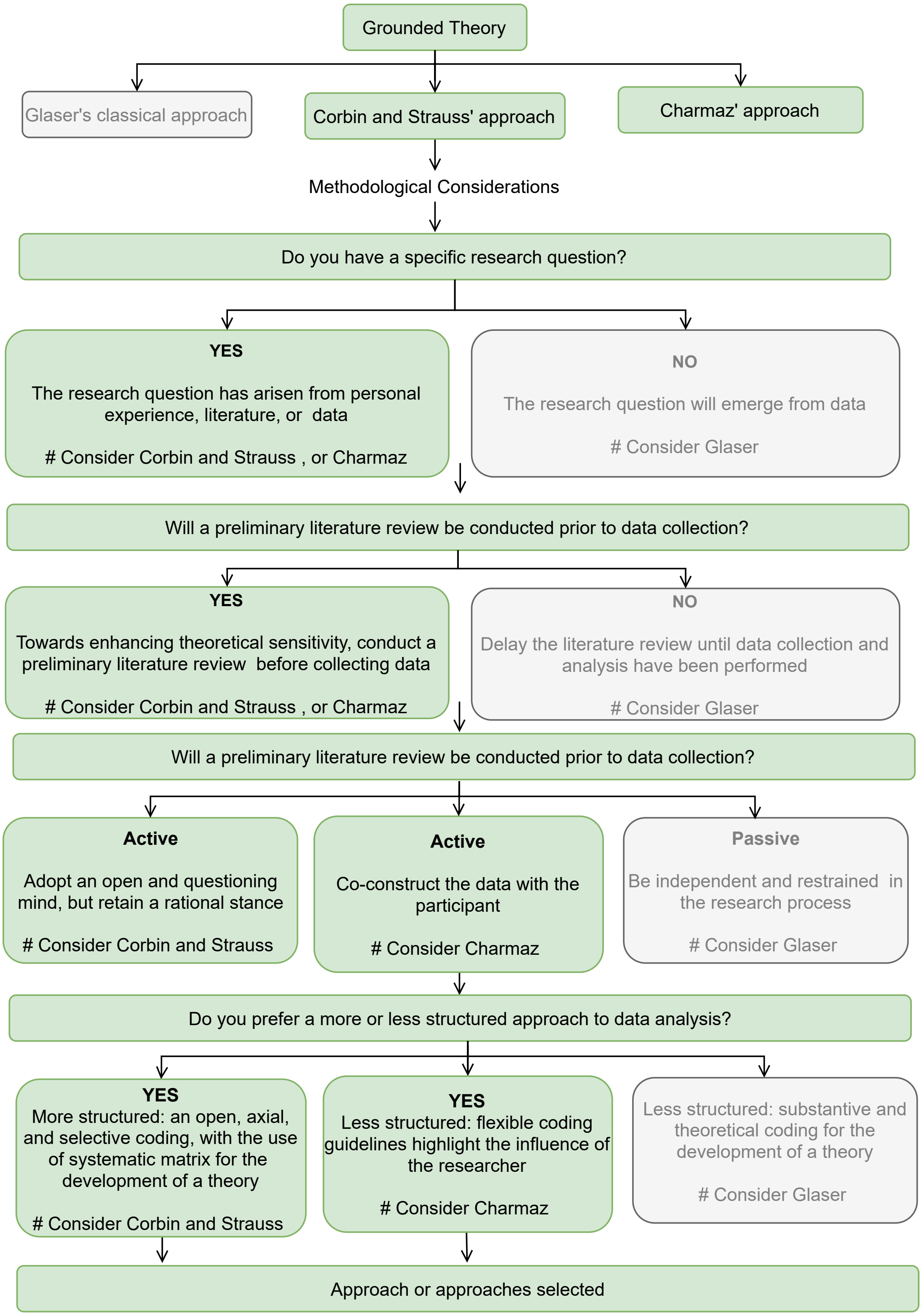


Figure 8 – Guidelines for the selection of grounded theory approaches

Source: Adapted from Polacsek *et al.* (2018).

of technical interoperability requirements and its relation to SoS mission from the perspective the industry practitioners and academics developing industrial projects. The theory projection uses codification and constant questions, such as "how can interoperability requirements of an SoS Mission be specified?" or "How can an SoS Mission of an organizational mission be identified?". The answers support the understanding of concerns related to stakeholders' needs and specification of a systematic and interactive fashion for the creation of strategies that identify the mission and analyze technical interoperability requirements. Ground theory enables flexible and open thoughts that help criticism and design of better actions that focus on the sample during data acquisition. The process is performed in an iterative and incremental fashion; it returns to a previous step whenever a critical analysis of the situation is required for the avoidance of bias. The testing of the theory does not need to be performed by its designer, but by other researchers (STRAUSS; CORBIN, 1998).

3.3 Grounded Theory Method

The grounded theory method involves several activities encompassed in the classical approach designed by Glaser and Strauss (1967), or in a merge of the approaches developed by Strauss and Corbin (1990) and Charmaz (2005). The activities proposed by Strauss and Corbin (1990) and Charmaz (2005) were incorporated by Pandit (1996) in a process with five iterative phases (cf. Appendix C). Our research used empirical data and was also based on literature-driven analyses; therefore, some adaptations were made, as showed in blue color in Table 4. Pandit's activities were reorganized into four phases. In the original process (PANDIT, 1996), the data protocol is established in `Phase 2`, however, the benefits of the establishment of a protocol comprehend specification of activities, research questions, and evaluation of the results (BRERETON *et al.*, 2008). A protocol supports the recognition of gaps in the literature and research questions, as well as reduction in research bias (KITCHENHAM; CHARTERS, 2007). Therefore, our protocol was anticipated to phase 1.

Pandit (1996) also suggested identifying the first case (i.e., step 2) and extending the analysis to other cases, always returning to step 2 towards theoretically saturating the researched question. We selected the first case during the establishment of the protocol, i.e., step 1, applying criteria specific for selection regarding industry and the way it develops the SoS and similar systems, and performs interoperability requirements analyses. In the first case selected, `step 2` was not adopted as described in Pandit (1996); however, when `step 7`, had been reached, both steps, i.e., `step 2` and `step 7`), interacted for coding the data until achievement of data saturation. Data were acquired through the interaction of the following four sets of data units: (i) technical literature analysis (e.g., articles, books, technical reports, and theses), (ii) non-technical literature analysis (e.g., documents, laws, reports, and catalogs), (iii) empirical data analysis (e.g., survey, workshops, and interviews), and (iv) meetings and discussions with empirical or academic experts at all stages. `Data register` crosswise supported coding and analysis (Figure 9), and consists of notes, observations, and memos[1] (CHARMAZ, 2005) for the registration

1 Memos are theorizing write-ups for written or drawn incidents when it is field (GLASER; STRAUSS, 1967)

Table 4 – Pandit's grounded theory design process

Source: Adapted from Pandit (1996).

PHASES	ACTIVITIES
PHASE 1 - RESEARCH DESIGN PLANNING	
Step 1 - Development of a rigorous protocol	Planning of entire conduction, research question(s), data acquisition and analysis
Step 2 - Selection of cases	Theoretical, not random, sampling
Step 3 - Review of technical and non-technical literature	Definition of research questions and data analysis by multiple data collection methods
PHASE 2 - DATA ACQUISITION	
Step 4 - Entering the field	Overlap of data collection and analysis by flexible data collection methods
Step 5 - Data ordering	Chronologically arraying of collected data
PHASE 3 -ANALYSIS CONDUCTION	
Step 6 - Analysis of data on the first case	Unification of coding (open, color code, axial, and selected coding)
Step 7 - Theoretical sampling	Theoretical replication across cases (go to step 2 until theoretical saturation)
Step 8 - Reaching closure	Theoretical saturation when possible
PHASE 4 - LITERATURE COMPARISON	
Step 9 - Comparison between emergent theory and literature	Comparisons with similar and conflicting frameworks

of ideas and insights in the field and during the mapping. The coding process, addressed in `step 6`, was divided into four categories, called: (i) *open coding*, which involves decomposition into smaller parts through line-by-line examination and comparison of data for the creation of concepts or codes (STRAUSS; CORBIN, 1998). Codes are considered keywords and short sentences, usually assigned to statements in an interview, open questions of a survey, or parts of a literature analysis. Concepts are interconnections between codes, whereas categories are classified as interrelationship of concepts; (ii) *color-coding* method, which is based on the idea of Alon *et al.* (1995) that each keyword or sentence receives a specific color according to a specific code. Color code was adopted in our study as an intermediate method to be applied between open and axial coding, and helped the identification of each code for multiple levels and relations through them; (iii) *axial coding*, which refers to evaluations of codes identified during the open and color coding phases for the reformulation of the categories and subcategories. It aims to synthesize and organize large amounts of data and present them to a new organization after open and color analyses (CORBIN; STRAUSS, 2008; CHARMAZ, 2005); and (iv) *selective coding*, which refers to the integration of categories towards the identification of a core variable

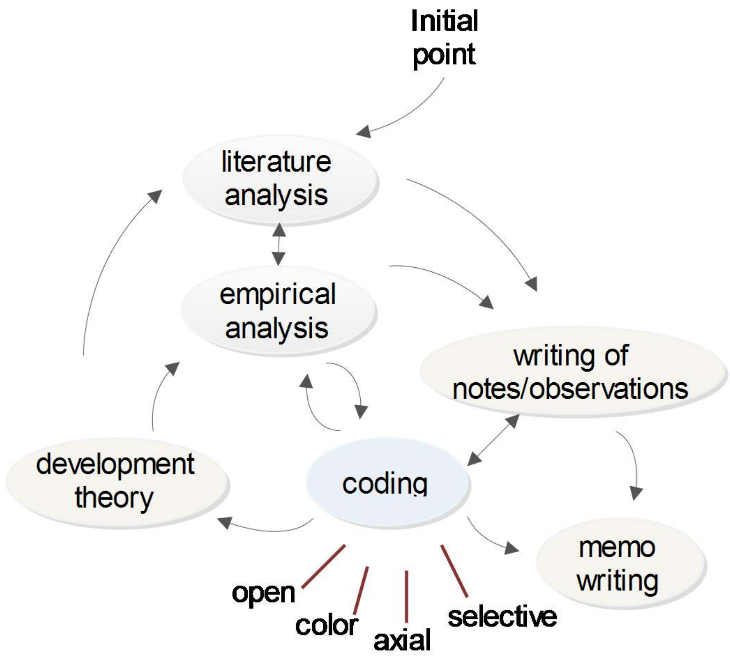


Figure 9 – Coding and analysis process

Source: Adapted from Strauss and Corbin (1990), Charmaz (2005).

or central theme. It keeps the analysis of only data connected to the emerging theory and related concepts for the design of an abstract theoretical schema in a final emergent theory (GLASER; STRAUSS, 1967; STRAUSS; CORBIN, 1990).

Although the activities are displayed in a sequential manner in the table, their use in the theory construction must be dynamic towards maintaining an evolutionary process of data analysis and comparisons with the literature. The aim of grounded theory is the establishment of a theory based on data gathered over a specific study, rather than the testing of a theory to be built. The testing and/or refining of the theory can be performed by other scientists (GLASER; STRAUSS, 1967; STRAUSS; CORBIN, 1990; CHARMAZ, 2005).

3.4 Phase 1: Research Design Planning

The protocol, including the research questions (cf. Chapter 1), data collection strategy, and analysis criteria was established, assessed and validated by practitioners and experts in SoS, SoS interoperability, requirements engineering, software architecture, agroindustry economics, economic sociology, and institutional economics. A free and clarified consent statement was assigned for maintaining the confidentiality of the population. The participants, however, were allowed to withdraw at any time.

3.4.1 Sample Selection

The population was incrementally selected during the research, so that the research question could be answered. An initial set of cases was chosen in the design of the protocol according to the concept of cases of Miles *et al.* (2014). Further cases were also selected during the research, therefore, the corpus was built according to its evolution. The objective of a constant data identification is to offer theoretical help towards the achievement of highest possible saturation (STRAUSS; CORBIN, 1990; CHARMAZ, 2014; JOHNSON *et al.*, 2018) and increase in the support to the theory.

The sample selections must be intentional to better support the research domain (GLASER; STRAUSS, 1967; STRAUSS; CORBIN, 1998). For an emergent domain, such as the mission engineering of SoS, those samples may not be available for industrial applications, or still are insufficient; therefore, more than one data sample type are used for the construction of a consistent theory. The following six best practices are essential and help in the evolution process and establishment of the theory: (i) researchers must constantly and systematically ask questions to themselves and their sample; (ii) comparisons among concepts must be made towards a clear understanding and creation of propositions; (iii) memos can be written for the recording and refinement of ideas and insights towards evolving the methodology; (iv) field notes can be written when a long registration cannot record insights and main information for the evolution of the methodology; (v) a baseline must be established for promoting discussions with experts on the evolution of the methodology; and (vi) the practical applicability (e.g., usability of the industry) of the methodology must be constantly checked. The first three best practices were proposed by Strauss and Corbin (1990). The first is related to the identification of a problem or concept, whereas the second regards the semantics of each concept, i.e., all possible meanings known by the author and research involved, their interpretation and relationship. The third and fourth best practices help the researcher to not lose information during the coding and interaction with the individuals of the sample and discussion with experts. Memos should be registered at the moment they appear, because they are fragile and ephemeral, whereas data are always available (GLASER; STRAUSS, 1967). The fifth practice enables interventions of experts of the academia and/or industry practitioners towards increasing effectiveness for the application of the methodology during the development process of the emergent theory. Six practices support the verification and simplify it towards meeting the business goals of the organization. All practices were applied in the design of the theory. However, we considered the existence of ontologies (DOLLING, 1995; MAYK; MADNI, 2006; TJONG, 2008; UNGER; CIMIANO, 2011; BENALI *et al.*, 2014; ABDALLA, 2017; HAYMAN *et al.*, 2018) to address the diversity of concepts. Such ontologies were applied mainly for solving ambiguities in the requirements analysis level and identification of examples of technical interoperability requirements.

Our systematic mapping showed few studies considered the term "interoperability requirements" and the way it collaborates to the success of the SoS particularly, when the requirements are derivatives of the SoS Mission. Therefore, our aim was to verify, in practice, the way those requirements were used and considered by industry professionals or academics that work with industrial projects. The survey sample started with the analysis of industrial or academic authors of main international events[2] and journals [3] in the SoS context and similar systems [4]. We performed the analysis from the current year to previous events, considering the last ten years [5], and visited SoS projects (e.g.,DANSE[6], T-AREA-SoS[7], AMADEOS[8], ROAD2 SoS[9], and CPS-SoS[10], COMPASS[11]) governmental organizations (e.g., U.S. Department of Defense (DoD), UK Ministry of Defence (MoD), NASA, and US Department of Transport), non-profit organizations (e.g., Mitre Corporate, INCOSE, and SRI International), and universities (e.g., Maastricht, Software Engineering Institute, and Naval Postgraduate School) for identifying members from our target audience. Industries, such as Fraunhofer-Gesellschaft, IBM, Siemens AG, Cientive Group, Inc., GM Corporate, Jaguar Land Rover, John Deere, and CAISE IH were introduced as members of our survey sample. 658 individuals were identified, and duplicate entries of samples that neither were industry, nor had evidence of work with industrial projects were excluded, totaling 432 individuals. Although interoperability has been long researched, few professionals work in this area (e.g., interoperability engineers). Therefore, the respondents were also professionals of SoS architecture, SoS requirements or SoS design.

A new sample of cases was selected (i.e., return to step 2), since after the analysis of survey data and documents, the theoretical saturation[12] had not been reached. In `Phase 2`, two rounds of semi-structured or unstructured interviews were conducted (SEAMAN, 2008; LAWRENCE, 2000). The former involved a combination of predefined and unplanned questions, i.e., combined structured and unstructured interviews. The structured questions provided

---

2 (e.g., IEEE International Conference on System of Systems Engineering, International Council on Systems Engineering (INCOSE), International Workshop on Software Engineering for Systems-of-Systems (SESOS), ACM/IEEE International Conference on Cyber-Physical Systems and International Conference on the Internet of Things)

3 (e.g., International Journal of System of Systems Engineering (IJSSE), Journal of Systems and Software (JSS), Journal of Systems Science and Systems Engineering (JSCSE), INCOSE Systems Engineering Journal, IEEE Internet of Things Journal, IEEE Systems Journal, International Journal of Adaptive and Innovative Systems, Complex Adaptive Systems Modeling, and Requirements Engineering Journal)

4 (i.e., that have characteristics of SoS, such as self-adaptive, autonomic systems, collaborative systems, complex systems, dynamic systems, distributed systems, integrated systems, cyber-physical systems, ultra-large systems, context-awareness systems)

5 Some journals such as "IEEE Internet of Things Journal" have less ten years, thus we researched only the issues available

6 <http://danse-ip.eu/home/index.html>

7 <https://www.tareasos.eu/>

8 <http://amadeos-project.eu/>

9 <http://road2sos-project.eu/cms/front_content.php>

10 <http://www.cpsos.eu/>

11 <http://www.compass-research.eu/>

12 Theoretical saturation is reached when a new case does not add any code to the existing categories; therefore, conduction can be finished and `Phase 4` is started (STRAUSS; CORBIN, 1998; ADOLPH *et al.*, 2011).

consistency, while unstructured ones provided the interviewees with more freedom to share new ideas and experiences, hence, greater interaction between interviewer and interviewee and discussion of novel issues. The lack of predefined questions causes interviewers to lose focus, spend considerable time, and find results difficult to be generalized. The objective of this thesis was explained to the participants, and interviews were conducted with 27 different subjects distributed among German, the USA, Spain, and Brazil. 13 were industrial practitioners and the other 14 were researchers developing industrial projects on different topics related to the object of the analysis.

3.4.2 Literature Review

The literature review does not need to be completed in advance. During step 1, the researcher should have in mind the initial scope of the research and delineate borders for the identification of the problem and formulation of general research questions (CHARMAZ, 2005). We initially identified the key concepts (CHARMAZ, 2014; STRAUSS; CORBIN, 1998) applied to the establishment of our search strings or snowballing method (WOHLIN, 2014a) for performing systematic mappings (KITCHENHAM; CHARTERS, 2007; PERTENSEN *et al.*, 2015). Four systematic mappings were conducted and the research question was refined with new empirical data and literature analysis. The latter enables extension for the design of new theories and widening of the boundaries of understanding (REMENYI, 2014). We compared or aligned literature with empirical data towards a better comprehension of the way technical interoperability requirements are derived from mission. The industry has a manager that solves technical threats and barriers for maintaining a sustainable process (e.g., software, hardware, standards, protocols).

3.5 Phase 2: Data Acquisition

Data are acquired according to the protocol established in Section 3.4. The process was built through data and methodology triangulation, which is one of the criteria proposed by Strauss and Corbin (1990) for the assessment of the research quality. Triangulation often depends on the topic of analysis and easiness of access to data, and influences findings (CHARMAZ, 2005). As described in Section 3.3 and illustrated in Figure 9, the techniques adopted in this process were systematic mapping, including analysis of technical and non-technical literature (i.e. gray literature), and snowballing, also used for surveys, and interviews. Non-technical literature, such as dissertations and technical reports, was also analyzed, and notes, observations, and memos were made in the field and during the systematic mapping and survey analysis.

3.5.1 Technical and Non-technical Literature Data Collection

The systematic mapping was used for gathering data from technical and non-technical literature according to a specific protocol and a design strategy. The topics of four mappings conducted are presented in the following sections: Section 3.5.1.1: software process development in SoS context; Section 3.5.1.2: formal and semi-formal modeling for SoSRE; Section 3.5.1.3: SoS mission performed by snowballing; and Section 3.5.1.4: interoperability of SoS. All mappings were used in a literature-driven analysis for refining the theory.

3.5.1.1 Software Development Process in the System-of-Systems Context

The first mapping was performed in the software development processes context for SoS from April/2015 to October/2015 and updated from September/2018 to December/2018, with six researchers, of whom four were experts. In 2015, 493 primary studies were obtained from all sources of data (i.e., engine search, snowballing, related work, and experts' opinions (see details in Appendix D). After the application of the selection criteria to title and abstract, 88 primary studies were selected for full reading. The criteria were applied again, thus resulting in the inclusion of 33 primary studies. 225 studies were recovered in the update, which followed the same process adopted in the first conduction. 42 duplicate studies were removed, 183 metadata of primary studies were analyzed by the selection criteria, and 55 primary studies were selected for complete reading, resulting in other 12 studies included. The process from April/2015 to December/2018 included 45 primary studies (cf. Appendix E), of which 33% (i.e., 15 studies) had been published in the last five years and 73.% (i.e., 33 studies) had been published in the last ten years, which points to a growing interest in the research area over the past decade,

During the complete reading, memo registrations were performed, so that no important information or ideas would be lost. Such descriptions were made as statements and questions and used in the latter phase for supporting our theory and discussion. Below is an example of a memo based on questions:

> **Memo 1**
> May 10, 2015
> Summary Memo
>
> *The SoS development initiatives have proposed ways for the establishment of architectures, but how have they met the requirements in such architectures? How have the requirements been identified, specified, analyzed, verified, and validated? How do they identify a mission to be performed? Is there a trade-off analysis for different requirements that considers the complexity of SoS? . How do the requirements behave at design time and runtime for multiple systems? Are the requirements handled similarly for all types of SoS and throughout the development chain? Is there a difference in the treatment according to the domain?*

Writing successive memos throughout this process kept us engaged with different focuses important for increasing the level of critical analysis and abstraction of our ideas and quickening our analytic work and productivity (CHARMAZ, 2005). Another feature used was mapping notes, whose, function is similar to noting and registering the field. We registered data considered important during the reading or a brief description of the most relevant information.

The data analysis was finished and discussed with experts, who emphasized the need for evaluations of the requirements analysis and specification process. Therefore, we moved towards understanding how to support the activities of requirements at design time and runtime, since analysis provides more information about requirements instead of only counting on information of the elicitation and validation activities. A second systematic mapping was then performed for observations on the way SoS have been modeled or specified. Since SoS can be found in several domains, the domain was not delimited to SoS and similar systems. The selection process was conducted manually by the author of this thesis.

3.5.1.2 Formal and Semi-Formal Languages and Techniques for Systems-of-Systems Requirements Modeling

This systematic mapping was conducted from April/2017 to August/2017 following the protocol described in Appendix F. The selection process was divided into two phases and eight steps, according to recommendations of Magdaleno *et al.* (2012): (i) selection and preliminary organization of primary studies based on title, abstract and keywords, with applications of inclusion criterion (IC) and exclusion criterion (EC); (ii) similarity comparison; (iii) reading of introduction, proposal, and conclusion; (iv) analysis by two groups of experts; (v) selection of primary studies based on the reading of full studies; (vi) evaluation of the studies quality; (vii) extraction of data included studies; and (viii) synthesis of data. The first four activities correspond to the first analysis phase, whereas the others correspond to the second one.

The procedure obtained 4,751 studies, of which 3993 primary studies were retrieved from six publication databases, and 758 were selected manually in publication venues. After the exclusion of duplicate studies (839), 3,912 remained for selection. The title, abstract, and, when necessary, introduction sections were read and the selection criteria were applied. 781 primary studies were selected for a similarity analysis. Primary studies not considered duplicate by the support tool were compared for ensuring they were single in our database. For instance, the study published by (PEPER *et al.*, 1997) was returned by Scopus and ISI web of Science database; however, it was not considered duplicate by parsif.al tool, due to a small difference between the records of those studies in each database[13]. In this case, one of them was manually rejected according to the criteria of duplicity. Other similarities found were related to studies that have part of equal abstract (e.g., context) and no author in common, and those in different venues and years from the same authors and with same abstract. Two strategies were adopted for the assessment of such similarities. The studies were ordered by abstract and compared with authors, publication year and venue - closer attention was given to studies with similar abstracts,

[13] The title recorded in Scopus was *Generic approach to the formal specification of requirements* and in ISI Web of Science *A generic approach to the formal specification of requirements*.

which were ordered by authors and related to title, year, and abstract - the focus was on studies conducted by the same author and with similar abstracts, but from different years. 73 duplicate studies were excluded. In the final process, 96 studies had been selected and the full reading and selection criteria were applied again. A set of 25 primary studies was selected for data extraction and one study (ZHANG, 2012) was inserted, as suggested by an expert, thus totaling 26 studies (cf. Appendix G).

3.5.1.3 Mission of system-of-systems

The previous systematic mappings used several databases, and one of them followed only the snowballing guidelines proposed by Wohlin (2014b). It was carried out from December/2017 to May/2018, and used the protocol described in Appendix H for the selection of studies that addressed the mission, business analysis, and business requirements in SoSE, and whether the mission requirements have been addressed by them. The first search was made in Scopus database, and retrieved 80 single studies. After the application of the selection criteria, only nine were accepted for a full reading, of which three were included (BHASIN *et al.*, 2008; MOKHTARPOUR; STRACENER, 2015; GARCÉS; NAKAGAWA, 2017). Backward snowballing (i.e., check of the references of the studies) was iteratively applied and 14 primary studies were included. Although the conduction period finished in May/2018, all citations (i.e., forward snowballing) of studies identified until January/2019 were monitored and five new ones were selected for data analysis, thus resulting in 20 studies (cf. Appendix I). Such primary studies are related to mission engineering (SOUSA-POZA, 2015).

Writing memos and notes enabled the relation of concepts and differentiation of the application of mission analysis and mission engineering in SoSE. The concepts of mission analysis are most specific and have been used in SoSE since the 1990s, with the development of engineering by DoD. They are also used by the system engineering of individual and complex systems, and the understanding of their application has become easier (SEBOK, 2014). Term mission engineering for SoS is an emergent field of analysis with few technical studies available.

3.5.1.4 Interoperability of system-of-systems

The last systematic mapping was conducted from June/2018 to November/2018. It followed the guidelines proposed by Wohlin (2014b) and used Scopus database (cf. Protocol in Appendix H). Although a few studies have addressed the requirements of the mission, the survey conducted with a mission engineer from the Software Engineering Institute (SEI) indicated the importance and benefits of approaches that support the elicitation and analysis of mission requirements for mission and interoperability (VESONDER *et al.*, 2018). Among such requirements, interoperability directly influences the accomplishment of the SoS mission. We investigated the way interoperability and technical interoperability had been addressed in the SoS development and whether they were derived from the mission. Technical interoperability is the first level to be met for holistic interoperability to occur in an SoS, however, failures at this level can affect both the other level and the SoS mission.

1050 unique studies were recovered from Scopus database. The Selection criteria were applied to title, abstract, and keywords and 128 primary studies were selected for full reading. The criteria were applied again and 14 studies were included for the application of backward snowballing. The references of the primary studies were analyzed and 22 studies were added, thus resulting in 36 studies for data extraction (cf. Appendix J). Mapping note and memos registry were carried out; they were important as a guide for the questions arisen. For instance, according to visual memo 1 and Figure 10, the aim is to record some areas related to interoperability, e.g., (i) heterogeneous environments that cover systems, interfaces, technologies, organizations, etc., (ii) hardware infrastructure, such as networking, cloud, servers storage, (iii) software (e.g., mediators, data format), and (iv) services (e.g., Application service, data service, network service, and communication service). Each of them was correlated to mission, requirements, and constituents, and the way they are influenced by business rules and standards for achieving interoperability was analyzed.

After data analysis, the results were discussed with experts. Such results suggested a deeper investigation into stakeholders' engagement end-to-end, since it is an important factor for the success of mission-driven SoS, although little addressed in the technical literature. Regarding interoperability, no clear separation is observed between the concepts of interoperability requirement and interoperability coalition (i.e., systems interoperation), which has led to conflicts of understanding and misapplication of the requirement concept that supports the coalition process. This differentiation is addressed in Chapter 4, which also shows the conclusion memo of this mapping.

3.5.2 Empirical Data Collection

The data were gathered through surveys, interviews, observations, video-conferences, workshops, writing of memos, field notes, and annotation approaches. for the development of the substantive theory. The triangulation technique improved the quality of the data collection, maintaining the reliability of information obtained in the field.

The populations of the interviews and surveys received *I_Tnumber* and *IDNumber* identifications, respectively (e.g., interview 10 - I_T10 and survey respondent 25 - ID25). However, in the thesis development, both were referred to as interviewees, except when a citation was made. The survey population answered the questionnaire in English, which was not translated. On the other hand, some recordings in Portuguese were translated by the researcher for the population of the interviews.

Visual Memo 1

July 23, 2018

Summary Visual Memo

The development process of SoS is surrounded by heterogeneous environments that must comply with several business rules and standards and also with external regulations to support the mission. Standards are used in software systems to improve data exchange through the correct application of a hardware infrastructure. Ubiquitous services can be provided by the interoperation of constituents. Business rules and standards generate different requirements, and new constituents might meet the SoS requirements, whereas legacy systems requirements should be met inside an SoS architecture. Requirements of SoS frequently change at runtime, and such changes must be quickly addressed by adaptations. However, some situations, needs and expectations must be directed at design time.

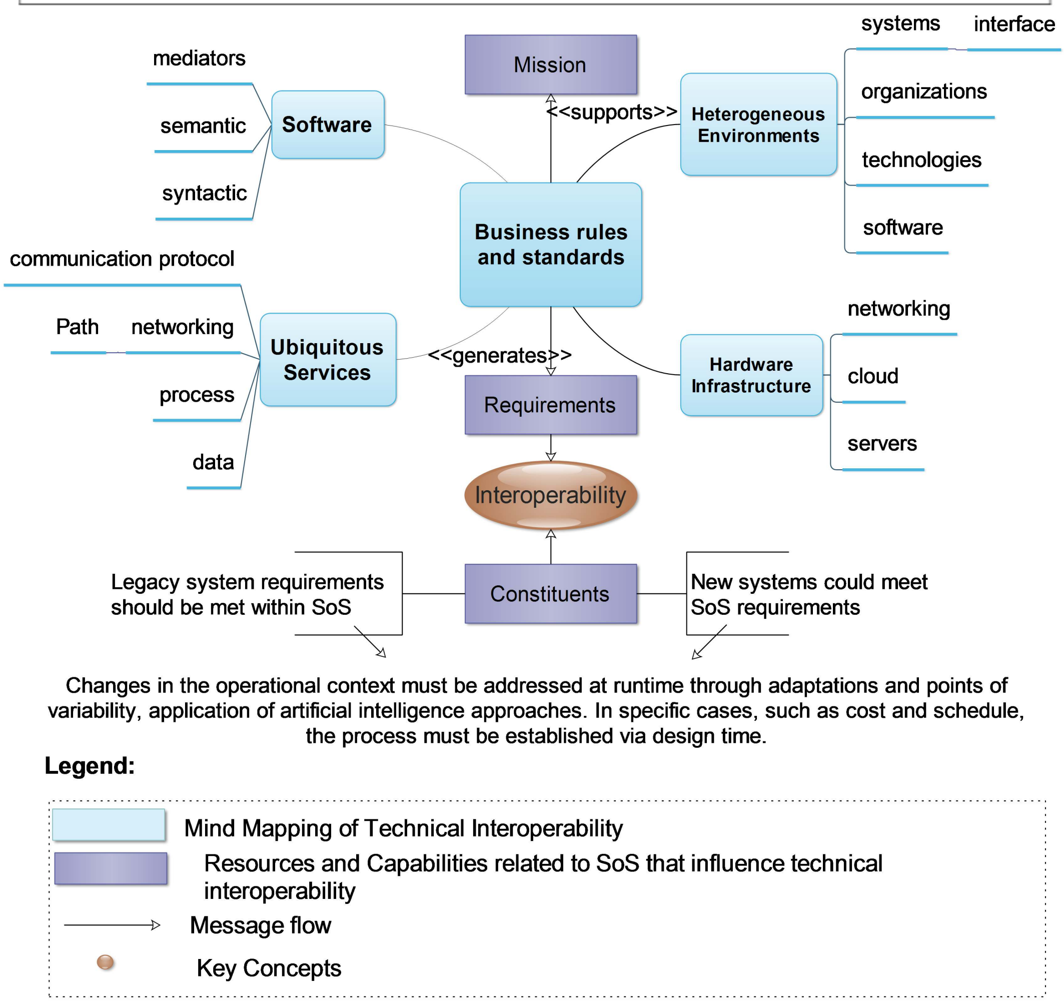


Figure 10 – Visual memo of the areas related to interoperability

Source: Elaborated by the author.

3.5.2.1 Survey Conduction

The survey was conducted from June/2018 to August/2018 as a qualitative survey with Open-ended, closed-ended, and hybrid questions (i.e., open-ended and closed-ended questions) (KASUNIC, 2005; SEAMAN, 2008). Qualitative methods support deeper analyses of problems and questions that involve variables of difficult quantification, such as human experience (SEAMAN, 2008). Open-ended and hybrid questions investigated in-depth relevant issues on the investigation purpose. Closed-ended questions were provided with a graduated scale, such as dichotomous response (yes/no, agree/disagree) and ordinal response (Likert scales), which supported the identification of some possible knowledge of a specific topic.

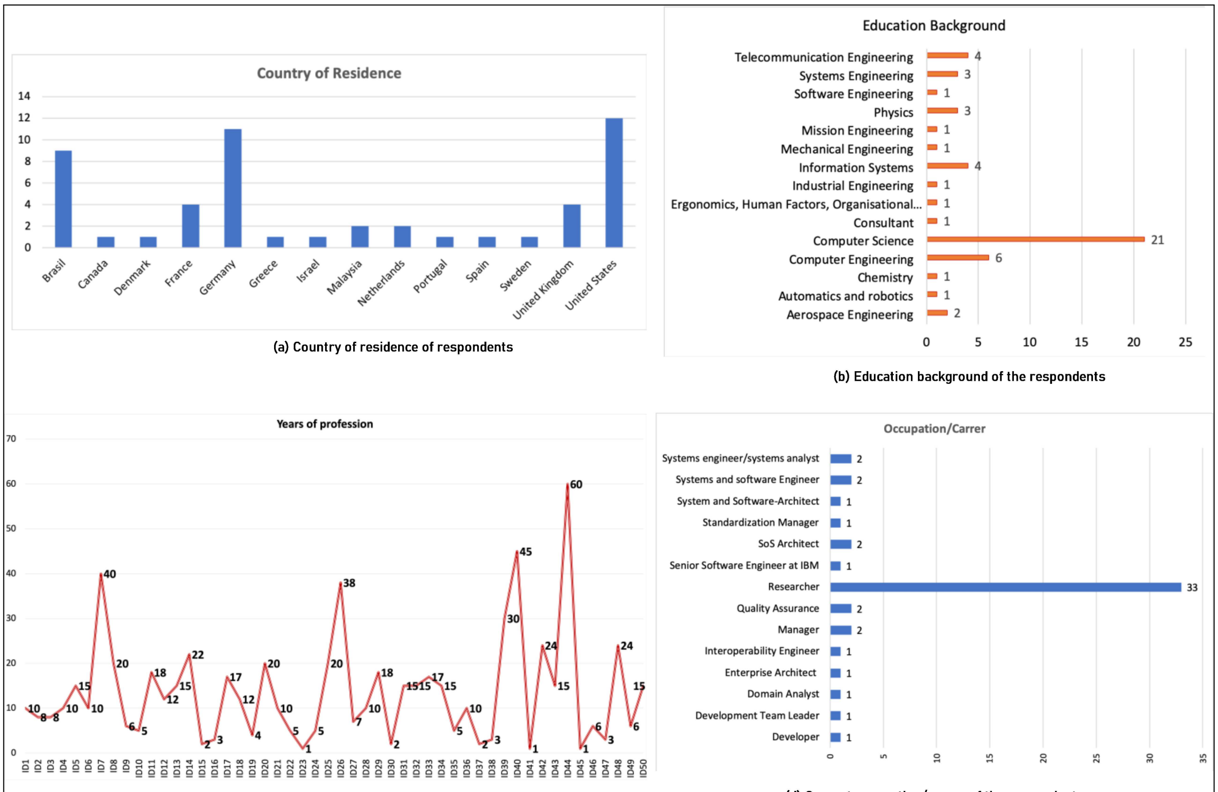
The questionnaire, comprised of three sets of issues, namely (i) architectural analysis practices for SoS interoperability requirements, (ii) landscape of SoS interoperability, and (iii) demographic data, besides initial and final considerations, and was distributed to 432 individuals via e-mail (systems generate link) by Survey Anyplace[14] (cf. Appendix K) online tool. It was answered by 51 individuals from our sample (i.e, 11,81%), and the answers were assessed in two steps, i.e., by us and by the researcher of Núcleo de Estatística Aplicada (NEA)[15] of the University of São Paulo. Although a respondent worked for an organization that does not develop SoS, our results were considered trustworthy and all answers were based on the experience of practitioners or academics involved with industrial projects. The respondent has been a system engineer for over 25 years and currently works with systems analysis. He has experience in the aerospace domain, which we consider an SoS domain. Figure 11 shows the profile of the 51 respondents.

Our sample was comprised of respondents from 14 countries (cf. Figure 11 a) with 1-year minimal experience (Master Students) and 60-years maximal experience (a PhD (cf. Figures 11 c and 11 e)). Their average of experience was approximately 14 years (cf. Figure 11 c). Most of them had an educational background in computer science, however, other 14 areas, among which computer engineering, telecommunication engineering, information systems, mission engineering, physics, and aerospace engineering (cf. Figure 11 b) also collaborated. From the 51 respondents, 33 were researchers working with industrial projects (Figure 11 d) described in (cf. Figure 11 f).

3.5.2.2 Interviews Conduction

The interviews were conducted from November/2017 to November/2019 with 27 participants, of whom 13 were experts developing industrial projects and 14 were industrial practitioners (cf. Appendix L). The first contact with each interviewee was made by email, and others were made in three different forms: (i) online interviews or discussions by skype or hangout, (ii) physical scheduled interviews, and (ii) a combination of (i) and (ii). The first meeting was held online

14 <https://www.surveyanyplace.com/>
15 <http://cemeai.icmc.usp.br/NEA/sobre/>

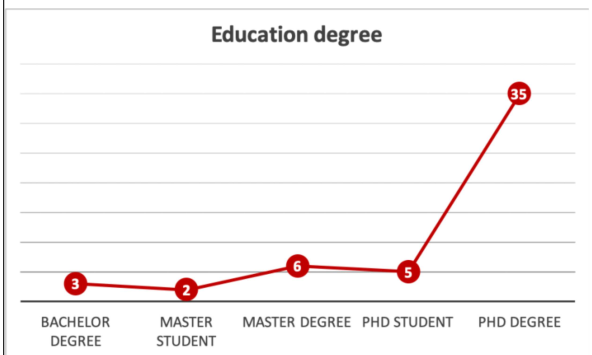


(a) Country of residence of respondents

(b) Education background of the respondents

(c) Time of profession of the respondents

(d) Current occupation/career of the respondents



(e) Education Degree of the respondents

ID	Companies
ID1	IBM
ID2	MITRE (Retired)
ID3	CAU-SES ("Complexity Are Us" - Systems Engineering Strategies)
ID5	Fraunhofer IESE
ID7	University
ID8	Bang & Olufsen
ID9	Acrelec
ID10	UTFPR INPE
ID12	Loughborough University
ID19	AT&T
ID24	University-research Setting
ID26	USN
ID27	Assistant Professor and Researcher
ID29	University of Malaya
ID31	University
ID33	UNINOVA
ID34	Siemens AG
ID37	Fraunhofer IESE
ID38	University of Lille, France
ID39	University of São Paulo - ICMC
ID41	University of Pau and Adour Countries
ID42	Universidad Politécnica de Madrid
ID43	Johns Hopkins University
ID44	General Electric Honeywell Edelbrock Ascent Logic IBM Self
ID45	IBM Research (Retired); Technion
ID47	MITRE
ID48	Universite Clermonte Auvergne
ID50	VMASC

(f) Excerpt from Companies of some respondents

Figure 11 – Excerpt from demographic data of the survey respondents

Source: Elaborated by the author.

between the author of this PhD Project and the interviewee for explanations on the aim of the thesis and ways of collaboration. After acceptance by the interviewees, a "Free and Clarified Consent" was presented to be signed by each of them, who also filled in a demographic profile. The interviews ranged from semi-structured to unstructured ones and were always recorded after the interviewees' authorization.

27 interviews were held and 10 interviewees[16] were consulted several times for different feedback checks, which resulted in 57 hours and 40 minutes of tape recording, besides checks with experts. The interview process started with nine interviewees, as shown in Table 5, which also displays their `IDs`, `subject type`, `experience domain`, and `number of years` of experience in the domain. New cases were added for the achievement of theoretical saturation. The biggest challenge was to find professionals that worked with SoS and interoperability requirements for supporting the data collection process. As the group was limited, we adopted multiple data sources to identify relevant data and information. An adequate theoretical sample can be used in theoretical and conceptual research. Abstract measures, such as representatives of a sample that can be generally judged offer no benefit to the process and, therefore, are not used as representative of a population or for the purpose of generalizations (CHARMAZ, 2005).

Table 5 – Subjects of the first round of interview

Source: Elaborated by the author.

ID	Subject Type	Experience Domain	Number of Year
IT_1	Industrial practitioners	Software Architecture and Safety Engineering	> 11
IT_2	Experts	Software Architecture, Reference Architecture, and SoS	> 11
IT_3	Experts working with industrial projects	Safety-critical System and SoS	> 11
IT_4	Industrial practitioners	Software Systems Engineering	> 7
IT_5	Industrial practitioners	Program Management	> 9
IT_6	Industrial practitioners	Senior Electrial Engineering	> 5
IT_7	Experts working with industrial projects	Software Testing and Agile Development	> 11
IT_8	Experts working with industrial projects	Requirements Engineering	> 40
IT_9	Industrial practitioners	Software Engineering, Project Management and research at Mobile Telecommunication	> 15

3.6 Phase 3: Analysis Conduction

The code process used ATLAS/ti 8.0[17] tool, and the data were extracted by the researcher and disagreements and relationships were discussed with interviewees and experts. The process of codification followed the four steps shown in Figure 9. Codification in the grounded theory method should follow basic guidelines to be always guided by the research question in all stages of activities (CHARMAZ, 2005). Therefore, flexibility and interactivity must focus on the research problem. Figure 12 shows an overview of the codification, which is an essential process

[16] The interviewees selected were from different domains
[17] <https://atlasti.com/>

in theory and can be conducted line-by-line, sentence-by-sentence, paragraph-by-paragraph, etc (i.e., small data are also known as incidents). A low-level analysis ensures truly grounded

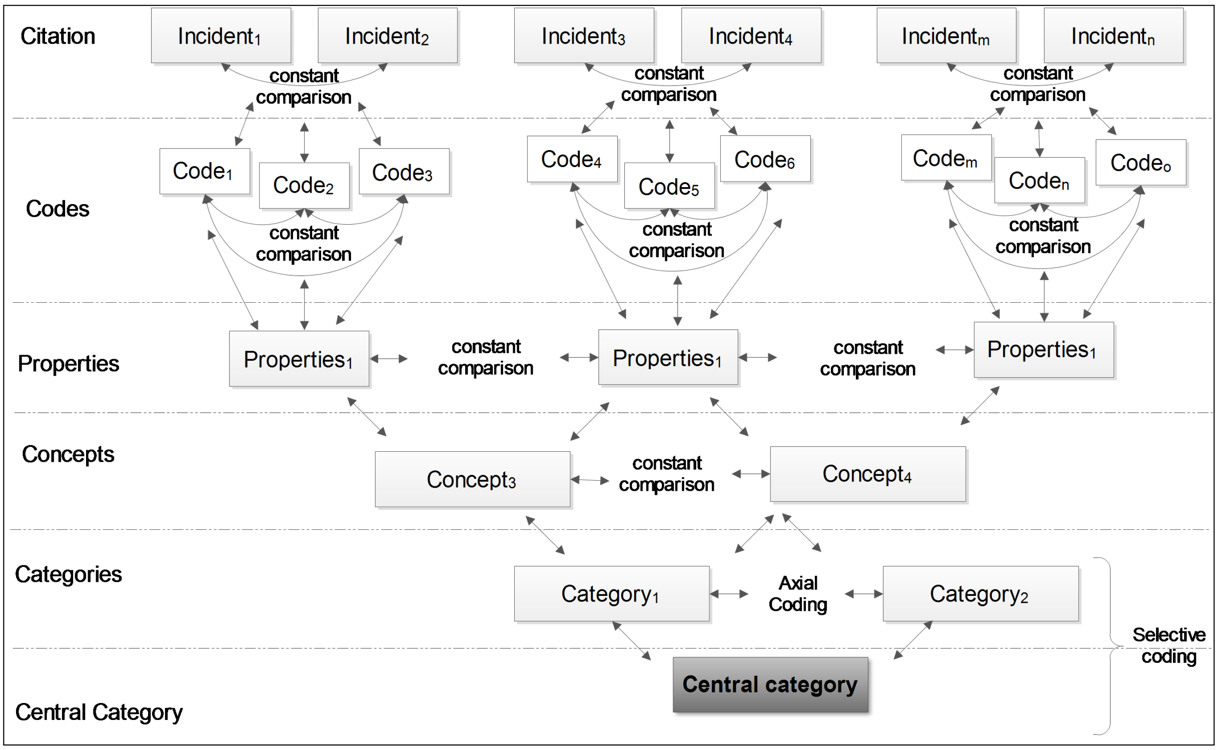
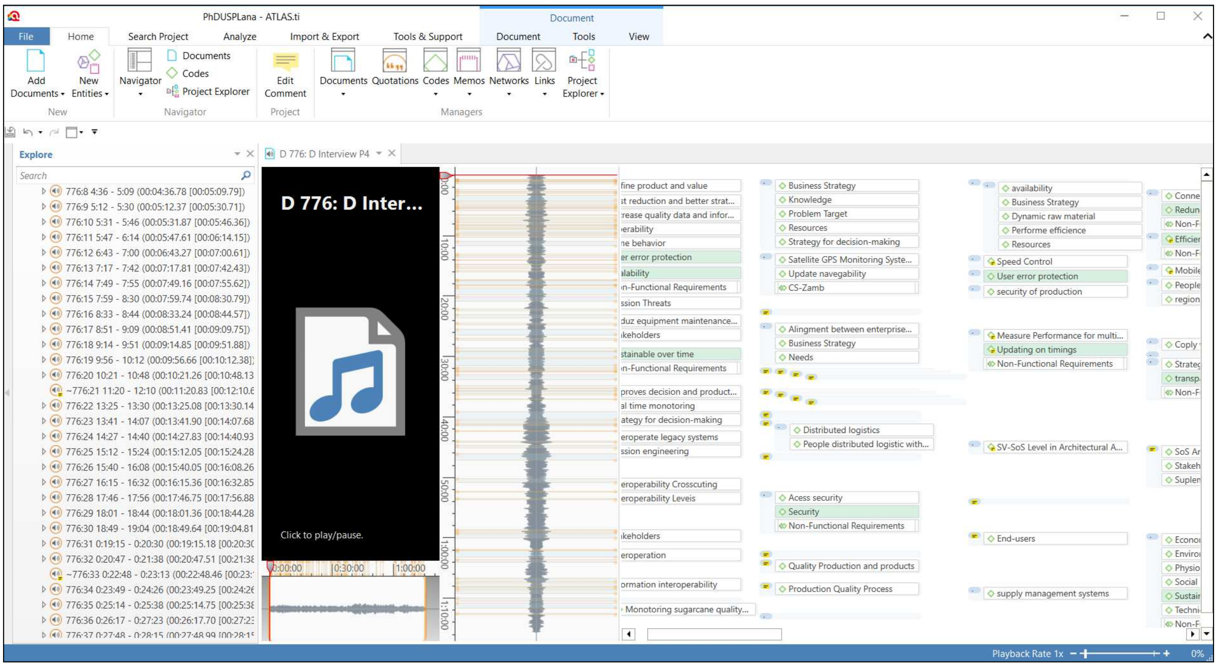


Figure 12 – Visual representation of the coding process

Source: Adapted from Freitas (2009).

and high-level categories emerge from data (CORBIN; STRAUSS, 2008). Codification starts after the first datum has been collected. We initiated the *open code* and *color code* to refine data through the low-level analysis and provide meanings to words or sentences (see Figure 13, which also displays an ATLAS Ti screen for the codification of documents, surveys, and records). For instance, Figure 13a shows the codification of D776, an interview record, and Figure 13b displays the application of color code in the codification process. An except of two interviews (i.e., IT_8 and IT_22) showed the way the codes for category `communication channel` and sub-category `network` were identified in the codification process.

315 codes were identified for the first nine interviews and 51 respondents, and other 293 codes were identified for the other 17 interviews, thus totaling 608 codes. Similar codes were grouped for the generation of concepts (i.e. network), and new and different codes with the same meaning were identified. The codes were renamed, and totaled 386. New analysis and validations were performed with interviewees towards refining the codes, which resulted in 271 codes (cf. Appendix M). As codification advanced, the level of abstraction increased until reaching a category level (e.g., communication channel, Figure 13b) that provided a phenomenon or sub-category association for the composition of actions and consequences that form the substantive theory (CHARMAZ, 2014; STRAUSS; CORBIN, 1998; CORBIN; STRAUSS, 2008).



(a) Excerpt of the interview analysis in Atlas.ti

Subcategory	Network
Codes:	Formal Data
	Meaning
	Application
	Transmitter/Receiver
	Interface description
	Transmission procotols
	Transmission rates
	Timelines or data latency
	Command (executable instruction)
	Standards

Interview
IT_8 : The systems can adapt to different interfaces and reconfigurations of several devices for the same vehicle. Interoperation is required, since it is used in several environments or domains.

Interview
IT_22: Interoperability uses various technologies that enable several systems to communicate with our organizations. However, it must support the decision-making process at all levels. Data are processed at runtime and made available at runtime or online, depending on the situation. Their performance must be assessed, since they cannot be maintained offline for a long time.

(b) Excerpt from codification by color coding

Figure 13 – Excerpt from codification by Atlas.ti and Microsoft Excel 2016.

Source: Elaborated by the author.

The use of *constant comparisons* enables the definition of a set of conditions for the designation of categories (GLASER; STRAUSS, 1967). Such comparisons were made in both levels, i.e., literature-driven and empirical analysis, and between empirical results and the literature. Figure 14 shows a low-level of literature comparison (studies were compared and those that connected requirements and architecture were identified). Such an analysis was important, since technical requirements directly influence architecture and its first phase of development, i.e., architectural analysis (SIDEREN; CHAPULART, 2015). Therefore, we claim for constant interactions between RE and software architecture in the SoS development, especially because interoperability requirements are an architectural concern usually not clearly described in Requirements Specification Document (RSD) or in Systems Specification Document (SyRE) due to the lack of both technical and/or business stakeholders and identification of the business requirements in a requirements process (BASS *et al.*, 2012; KNODEL; NAAB, 2016). The

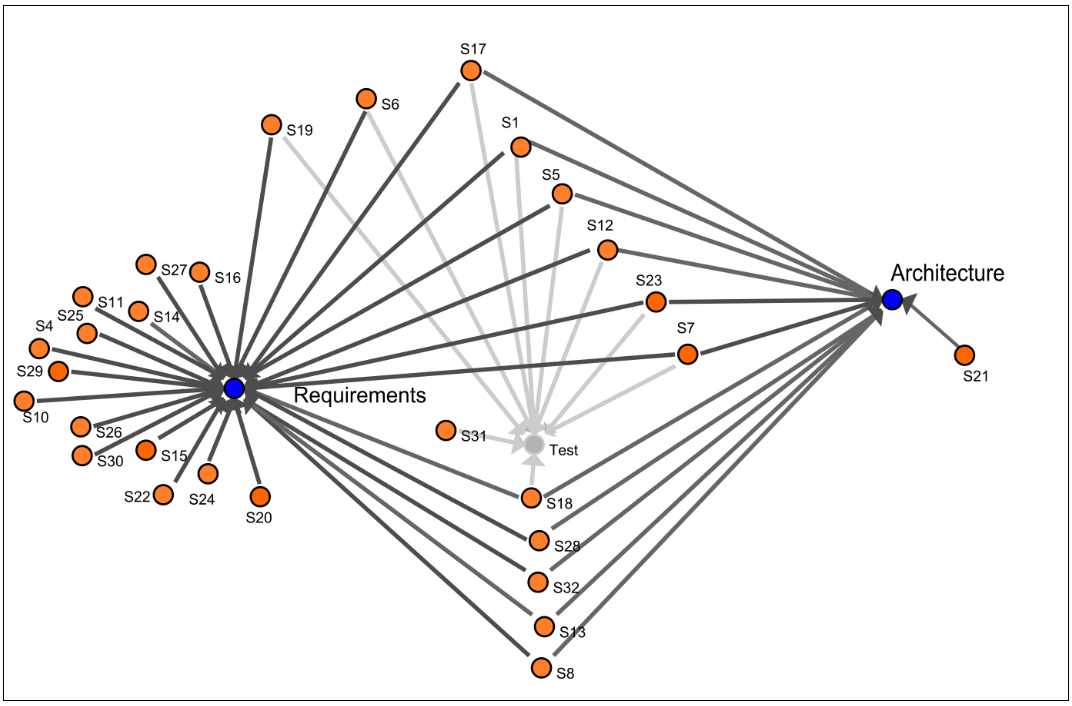


Figure 14 – Excerpt from relationship between requirements engineering and architecture

Source: Elaborated by the author.

comparisons identified similar incidents in divergent categories, thus enabling corrections during the codification process, and not only in the feedback process performed by the participants. Codification is considered a researcher's creative activity that requires his/her knowledge and experience for delineating a satisfactory and ideal representation of a phenomenon. The activity was supported by *theoretical sensitivity*, and questions about data were asked towards new possibilities and contradictions in categories already identified. They were based on scientific and practical knowledge and personal experience. The expert's knowledge was used for the validation of the categories and feedback from the participants. Figure 15 shows an example of validation- `usability` was classified as the concept of coalition interoperation and in the feedback on the generality of this requirement. Therefore, it was classified in the Non-functional Requirements (NFR) category.

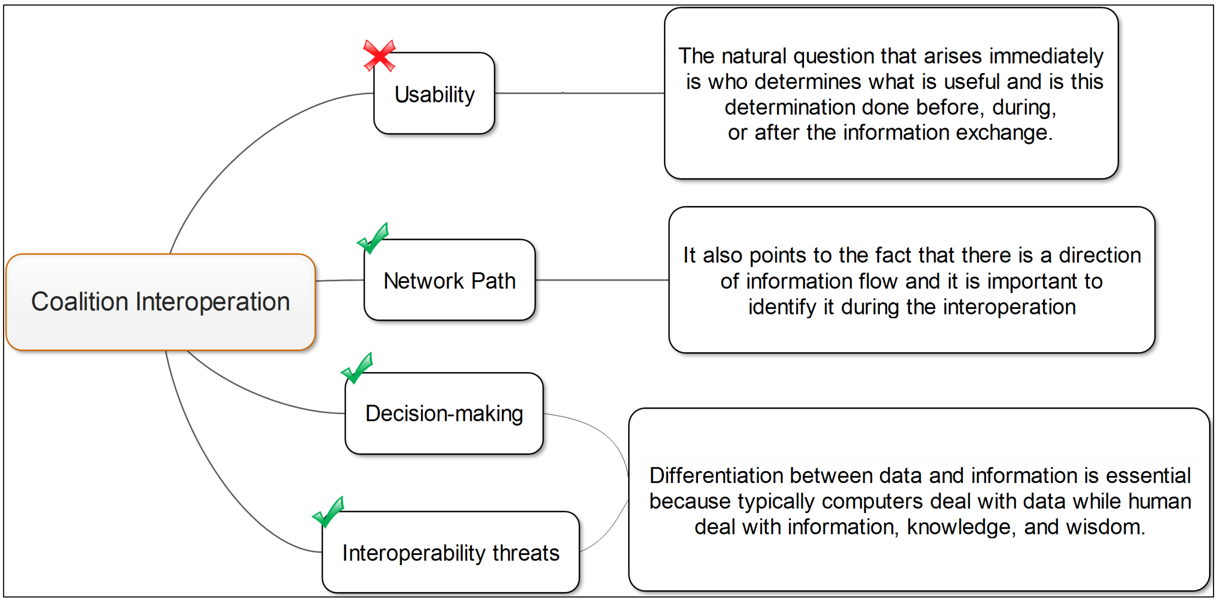


Figure 15 – Excerpt from a coalition interoperation concept

Source: Elaborated by the author.

The next step, called *axial coding*, relates categories to subcategories. Categories are considered high-level and more abstract than other concepts and generated by constant comparisons of similarities and differences. Codification was performed by a paradigm model that enables a systematic thought about data and the way codes are related to each other (HALAWEH, 2011). The data were analyzed according to color codes and open codes, and reorganized according to the predominant relationship.

Lastly, the *selective coding* was adopted for integrating and refining the theory, and organizing the other categories around the *central or core category*, considered the most representative, since it comprehends the topic of research and the most recurrent data.

3.7 Phase 4: Literature Comparison

This is the last phase of the construction of the substantive theory. The literature was revisited towards comparisons with the premises found in an empirical analysis that originated the emergent theory. As aforementioned, a crucial process in the design of the theory is the constant comparison among concepts, premises, or hypotheses and the literature for the verification of similarities and contradictions. The first validation of the emergent theory increases reliability, and the second brings opposite views to be analyzed. They enrich and deepen the analysis, offering new opportunities for improvements in both emerging theory and the literature (GLASER; STRAUSS, 1967; URQUHART, 1997).

The literature analysis through a systematic process is addressed in Section 3.5.1. All comparisons were based on the sets of primary studies, and, when necessary, manual searches were performed in databases or gray literature.

3.8 Threats to Validity

Several criteria have been proposed for the assessment of validity and reliability of an emergent theory (CORBIN; STRAUSS, 2008). Among such criteria, the one recommended by Charmaz (2005) comprehends both scientific and creative aspects necessary for the quality of research. Each criterion is addressed again in Chapter 6 together with its corresponding contribution.

Table 6 – Quality assessment of grounded theory

Source: Adapted from Charmaz (2005).

Criteria for the evaluation of the Grounded Theory

Criteria	**Description**
Credibility	Has the research acquired intimate familiarity with the setting or topic? Has a systematic comparison between observations and categories been made? Are there enough data for the validation of the claims? Is there strength of correlation between the data gathered and arguments? Has the research provided enough evidence to enable readers to make an independent assessment
Originality	Does category offer new insights? Has the analysis provided new findings and rendered data? What is the social and theoretical significance of this research? How can the substantive theory challenge, extend, or refine current ideas, concepts, and practices?
Resonance	Does the category portray the entireness of the studied experience? Has the substantive theory made sense to participants? Has the analysis offered them deeper insights about the construct around their lives and worlds?
Usefulness	Does the analysis offer interpretation to be used by people in their daily life? Can the analysis motivate further research in significant areas? Do the categories suggested have any generic process? Has it been designed? How can the research investigation contribute to knowledge?

According to Charmaz (2005), a positive combination of originality and credibility increases the criteria of resonance and usefulness, hence, the contribution.

3.9 Final Remarks

This chapter described the methodological procedures used for the design of the substantive theory and addressed the grounded theory method and the literature-driven analysis that supported its construction. The next chapter is devoted to the findings and the substantive theory itself.

CHAPTER 4

Grounding the Substantive Theory

4.1 Initial Remarks

THIS chapter describes the ten constitutive elements of the substantive theory raised in the empirical analysis, whose findings were first addressed for each section following from a literature-driven analysis. Section 4.2 introduces the categories that integrate the substantive theory and each category, its subcategories, and codes (see, Appendix M) and Section 4.2.1 focuses on the organization of SoS, divergence between the concepts of interoperation and interoperability requirements, and definition of concepts related to technical interoperability requirements relevant to this research. Section 4.2.2 is dedicated to the relation between operational technology and information technology, discussing some challenges and benefits to a necessary convergence. An SoS category, called blended SoS, was established from both behaviors, i.e., those controlled by a central control, and those to be negotiated with CSs owners. Table 8 was adapted and extended for each Maier's category, and the categories were related to the blended one.

Section 4.2.3 introduces the analysis of the incidents to the category of data and communication channels and the way they have influenced the analyses of technical interoperability requirements. Sections 4.2.4 and 4.2.5 are devoted to the quality attributes and their importance to various levels and issues related to sustainable technical interoperability, discussing the dilemma standardization versus openness and the social, economic, technical, environmental, and individual perspectives, respectively. Section 4.2.6 focuses on business alignment and misalignment from a design time and runtime viewpoint. The requirements were discussed as a connection between SoS mission and SoS architecture and the Triplet Model supported the specification and design of SoS. The relation between the Triplet Model and architectures, as well as the way each architecture supports each other, is also addressed. Sections 4.2.7 and 4.2.8 describe the problems and consequences based on the interviewee's viewpoint, and those arisen from errors of communication during the process of analysis or SoS design. Section 4.2.9 addresses the challenges faced by analysis of technical interoperability requirements and classified into three groups, called barriers, threats, and SoS properties. Section 4.2.10 concerns singularity mission strategy , which promotes mission engineering planning through enterprise mission for the refinement of the SoS mission and analysis of technical interoperability requirements. Finally, Section 4.3 provides the final considerations.

4.2 Constitutive Elements of the Substantive Theory

Grounded theory is not a linear process, since it enables multiple interactions between interviewer and interviewee, hence, the understanding of a phenomenon studied. In the present research, it is related to analysis of technical interoperability requirements driven by mission engineering. The following eleven categories have emerged from an empirical analysis supported by a literature analysis: **(i)** Duality between interoperation and interoperability requirements; **(ii)** Convergence between Operational Technology (OT) and Information Technology (IT); **(iii)** Divergence of the data and communication channel; **(iv)** Technical interoperability quality; **(v)** Sustainable technical interoperability; **(vi)** Strategical decisions; **(vii)** Problems and consequences; **(viii)** Aware communication; **(ix)** Technical interoperability requirement challenges; **(x)** Singularity Mission strategy ; and **(xi)** Technical interoperability awareness. Ten elements introduced in the following sections will be used as a basis for the development of the theory at its different levels together with central category `Technical interoperability awareness`, addressed in Chapter 5.

4.2.1 Duality between Interoperation and Interoperability Requirements

Interoperation and interoperability requirements are complementary; requirements are elements that support the process of interoperation together with other elements of interoperability, such as architectural decisions. Due to different understandings of the terms, SoS structure and the concepts related to interoperation and interoperability requirements were analyzed considering that they were used interchangeably during the interviews, many interviewees were confused (QURESHI *et al.*, 2010; ROQUE; CHAPURLAT, 2009). Figure 16 illustrates a generic structure of an SoS discussed with the interviewees. At $level0$, the constituents are concerned with their own mission (i.e., individual mission), capabilities, and services they can provide. According to SoS category, they are concerned with the capabilities provided by the SoS (e.g., directed); otherwise, they would achieve their mission and not the SoS mission. $Level1$ to $leveln$ consider the organization of the constituents that work together to achieve the SoS mission. Therefore, the architecture must be considered a backbone of the SoS, and SoS with central control has physical elements, such as middleware or mediators. Descendants' mission, capability, and requirements are identified by asking the *HOW* questions about the mission, capabilities, and requirements

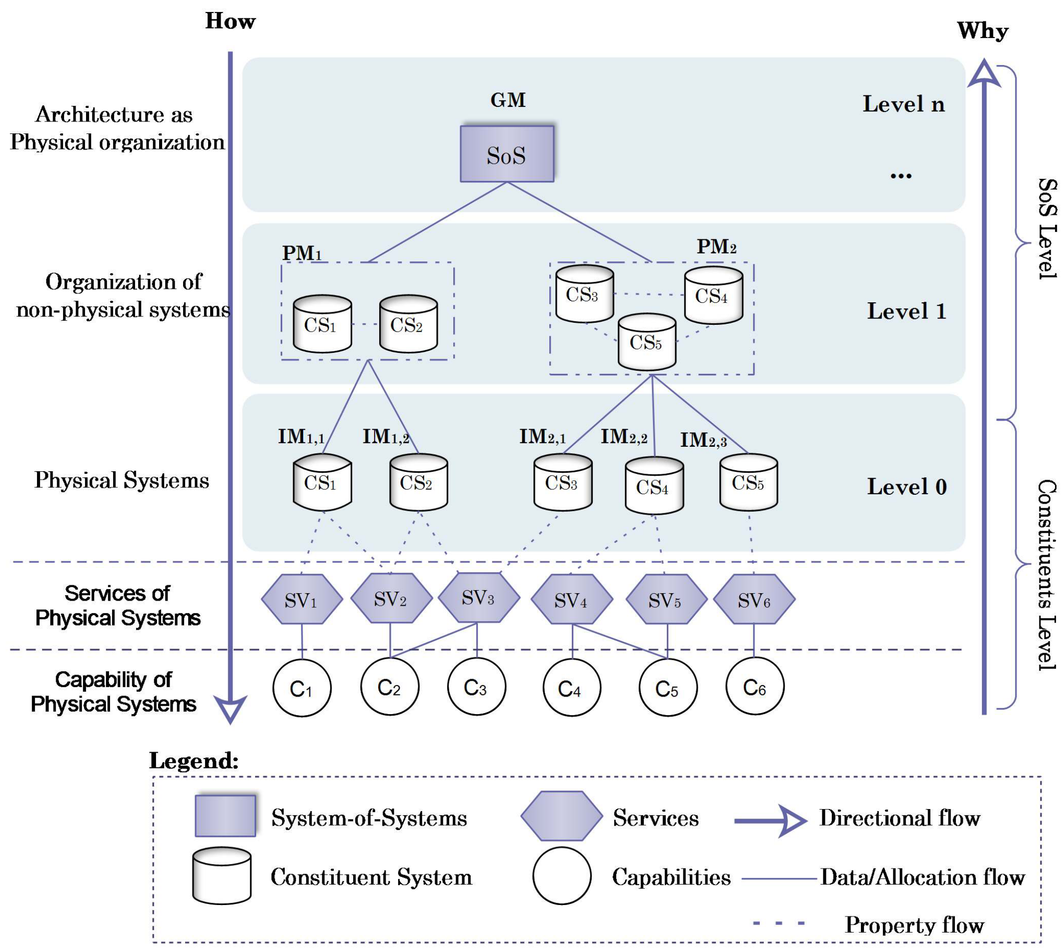


Figure 16 – Generic theory for organization from an SoS

Source: Elaborated by the author.

already identified, and parents' mission, capabilities, and requirements are identified through *WHY* questions, thus showing the importance of descendants in the achievement of the SoS Mission.

In the theory proposed, the duality is related to the unclear meaning of interoperation and interoperability requirements in the relevant literature. To define or identify such concepts, several interoperability definitions, types, and models found in literature were analyzed as well as the `technical interoperability quality` categories addressed in Section 4.2.4 (FORD *et al.*, 2007a; MALLEK *et al.*, 2012; PEARSON; KOLODNY, 2013; MORELAND., 2015; BILLAUD *et al.*, 2015; DACLIN *et al.*, 2016; ISO/IEC/IEEE-29148, 2011; ISO/IEC/IEEE-21840, 2019; ISO/IEC/IEEE-21839, 2019). Interoperability requirements and technical interoperability requirements are considered some of the elements needed to promote interoperation. The SoS interoperability requirements have been specified in the requirements engineering level, which considers "WHAT", and in the architecturally significant requirements level, which answers "WHAT and WHY" together with an architectural decision from the business level. Therefore, an **interoperability requirement** can be recognized as *a statement that specifies a function,* *ability or characteristics, related to the ability of a partner to ensure its partnership in terms of compatibility, interoperation, autonomy, and reversibility (MALLEK et al., 2012)*. On the other hand, a **technical interoperability requirement** is:

Definition 1. *A statement that specifies an infrastructure resource, characteristics, ability or constraints of communication with data exchange related to the ability of a collaborator to ensure their collaboration regarding composability, compatibility, integrability, interoperation, autonomy, variability, transparency, resilience, sustainability, and reversibility.*

Such requirements are elicited, analyzed, specified, and validated at design time and evolve or emerge at runtime, thus impacting the interoperation of the constituents that are a result only at runtime. Their design process is expressed in resources (e.g., mediators, network, architecture, and services) indispensable for interoperation satisfying an agreement contract with guarantees and demands to the interface level (e.g., behavior, functionalities, and data), operational agreements (availability, response time, resilience, and performance), commercial agreements (e.g., changes management and business guarantees), standards or their standardization with need for updated over time, or other resources formally imposed by the document (ISO/IEC/IEEE-29148, 2011; ISO/IEC/IEEE-24765, 2017; MALLEK *et al.*, 2012). **Interoperation** represents the ways constituents and mediators can be arranged towards forming the SoS that are inherently dynamic with concrete constituents and mediators that can participate in an SoS at runtime being often partially known or even unknown at design time. Such architectural elements must be identified at runtime towards proper arrangements that contribute to the accomplishment of the global mission, thus resulting in alternative SoS architectures. Interoperation representing an SoS can expose interfaces of its own, so that it can be composed to form larger SoS. The following statement supports these arguments.

> *Interoperability requirements are a false goal. In fact, the goal is not to interoperate the SoS, but identify, analyze, verify and validate the requirements of interoperability for the achievement of the SoS mission. Interoperability requirements are only a part of the process of interoperation and not the entire one. The interoperation of more than two constituents requires much more than What to do? and How to do it? Do I need to know why to do it? When to do? What is the rationale? and so forth (INTERVIEWEE ID44).*

> *Interoperability requires knowledge of the domains, since they establish the context of WHAT information is exchanged, knowledge on the technical details by architects, setting up of HOW information is exchanged, and, regarding its engineers, WHO brings both words together for a conceptually aligned technically mature exchange (INTERVIEWEE ID47).*

Figure 17 shows a general view of the collaboration of interoperability requirements and the SoS interoperation concept. CS_A collaborated with CS_B and CS_C and met particular

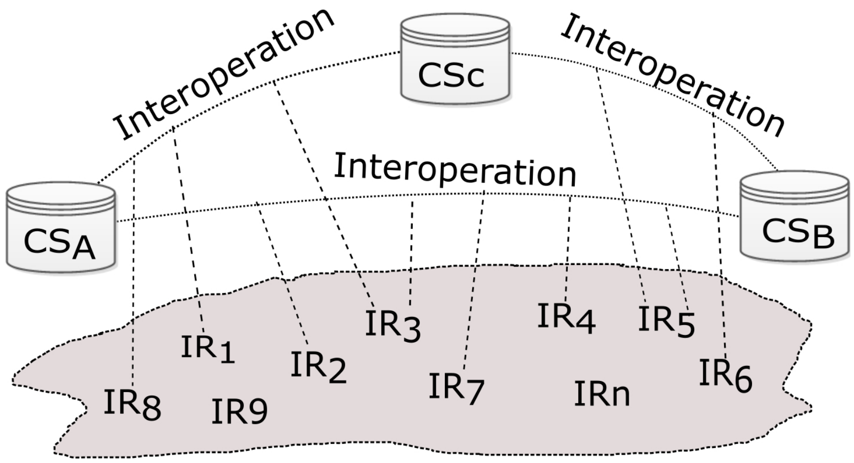


Figure 17 – Theoretical collaboration between interoperability requirements and SoS interoperation

Source: Elaborated by the author.

interoperability requirements (i.e., IR1, IR2, IR3, IR4, IR5, IR6, IR7, and IR8[1]) with all design decisions performed in an architectural synthesis. Such requirements were elicited from a sequence of behavior extracted from business processes orchestrated and/or choreographed according to the capabilities each one provides. An emerging trend in the SoS field is the outside-in viewpoint, which induces not only the attendance of stakeholders' value end-to-end, but also the mission. It evaluates technical interoperability requirements from the perspective of mission engineering, and not only considering mission analysis. All business processes are related to mission engineering, and include enterprise analysis (discussed in Section 4.2.6), which recognizes operational needs of an enterprise or a set of enterprises to may be solved by collaboration or implementation of complex software systems. Such systems face interoperability problems, discussed in Section 4.2.9 and 4.2.7, which must be treated before they influence the results of the global mission.

During the theorization process, the analysis of **technical interoperability requirements driven by mission engineering** can be comprehended as the *gathering, specification, verification, and validation of the technical capability and/or requirements through the discovery and comprehension of the challenges of the SoS infrastructure and the needs and expectations of stakeholders for the new systems or updated through the definition or understanding of its mission.* According to the interviewees, the analysis has benefits and difficulties, as shown in Table 7. The benefits help obtain the clear statements of interoperability requirements supporting the measurement of efforts for the implementation of SoS, while the resources can be prepared. Some technical constraints, known in advance, can be analyzed by the SoS architect in the synthesis phase. Mission and requirements are refined in requirements engineering and futher refined in the architectural analysis for the allocation of each capability and requirement for the

[1] Examples of IR: (i) The system shall use open protocols for establishing communication between the central control and their constituents and (ii) The persistence layer should be deployed for allowing the use of different database technologies.

Table 7 – Benefits and difficulties in analyses of interoperability requirements

Source: Elaborated by the author.

Benefits	**Difficulties**
Clear statement of interoperability requirements	Analysis involves high costs, and resources
Estimate of efforts SoS implantation	Testing higher priority
Preparation of resources	It is hard to perform
Prior knowledge of the technical constraints	Resources and time
Insight about business needs and expectations	Consequences are unknown
Identification of constituents	
Prioritization of capabilities and relation to CSs	
Identification of some infrastructures	

architectural elements. The difficulties indicated the reason why the analysis of interoperability requirements has not been performed - it involves high costs and resources, testing has a preference because it has more priority than interoperability analysis, and some interviewees consider it difficult to perform. The latter is related to challenges faced in the analysis process of the requirements by interviewees, such as (i) lack of a methodology (e.g., processes and methods), (ii) lack of best practices for practitioners, (iii) lack of a unified taxonomy, (iv) lack of guidelines for the interoperability analysis, (v) lack of a specific language that specifies interoperability requirements, (vi) lack of an automated technological support, and (vii) lack of support for inter-artifact and intra-artifact traceability. Although the interviewees agree on the importance of analysis of technical interoperability requirements, the consequences of not applying it are unknown; however, the results, for example, of not conducting the testing are known (in case of doubts, known results should be chosen). Furthermore, the current languages have been insufficient for the analyses of interoperability requirements to fulfill the interactions and comparisons between models and views of the different levels of development, thus hampering traceability. Traceability to the SoS level is more complex than CSs, since inter-artifact and intra-artifact must be traced in two directions and in multiple levels and layers, according to the complexity of the systems. A tool must support it and discover if the technical interoperability requirements have been met by the architecture. Such questions can be confirmed by some interviewees‘ comments. ID37 and ID44 highlighted the need for “Guidelines for backward compatibility after evolution” and “a fit for metric purposes”, respectively. ID26 reported the following concerns over the accelerating increase in both size and complexity of systems:

> *The conceptual competence of the participants. The dynamics, size, and heterogeneity of SoS threatens to exceed the cognitive capabilities of the participants. Methods may help somewhat. Practical taxonomies (mind the plural, e.g., non-unified because of the heterogeneous problem field) may help, and domain-specific notations may help (INTERVIEWEE ID26).*

4.2.2 Convergence between Operational Technology (OT) and Information Technology (IT)

The OT and IT convergence category has emerged from the relationship of two concepts, namely "degree of infrastructure control and management of equipment" and "degree of interoperability and monitoring", which is a direct concern of industry 4.0. Such concepts were individually created towards solving distinct problems with particular architecture and communication protocols. **IT** refers to technologies for information processing with the use of hardware, software, communication infrastructure, and related services for application connection and data exchange [2], and **OT** regards the use of hardware and software that detect or cause a change through the direct monitoring and/or control of physical devices, processes, and events in the enterprise[3]. Some examples of OT systems are embedded systems, cyber-physical systems, smart transportation, intelligent patient monitoring, agro4.0, healthcare management solutions, smart vehicles, fleet transportation, delivery optimization solutions, supply chain management solution, rail systems, and Internet of Things.

Such convergence can be understood as collaborative work, or even as a unification between OT and IT for the supply of services, use of protocols, standards, communication channels, control, etc. The design of a converged SoS architecture would eliminate the inefficiencies and technical barriers to interoperability, such as multiple standards and protocols. Although convergence is necessary, it poses new challenges to the analysis of technical interoperability requirements throughout the lifecycle of the SoS. Interviewees know that quality attributes, such as scalability, modularity, privacy, and composability, lead to some important technical interoperability issues, where maintaining the correctness and performing the V&V of the SoS are great challenging, when are not considered in systems development with timing constraints and continuous operation requirements (SERPANOS, 2018).

Such challenges affect the interoperability of legacy systems between OT and IT with new systems, and interoperability of networks, resource sharing, and security process must be enhanced. The security process is not particularly concerned with the OT, and poses threats of attacks that can lead to loss of life or assets. Risk analysis, dependability, and reliability are concerns that each day have more importance to technology, since the infrastructure is created for establish premises to be met and used by devices, sensors, actuators, transmitters, software, middleware, and among others.

The "degree of interoperability and monitoring" constantly require data and decisions at runtime, since the few data available online have become outdated for many decision-making processes, which must use current data. The interviewees reported a novel behavior (emergent

[2] Adapted from <https://csrc.nist.gov/glossary/term/Operational-technology>
[3] <https://csrc.nist.gov/glossary/term/Operational-technology>

behavior) not observed at runtime, when their systems were working together. They believe several things will be discovered as technical needs during the monitoring of systems in the operational environment. For instance, *EnterpriseN* was monitoring the various hardware quality requirements to measure the performance of their equipment operating in collaboration. When *EquipmentA* interoperated with *EquipmentB*, the temperature increased and the performance of the whole system decreased and returned to the normal, with the output of the system. Therefore, the evolutionary maintenance was performed in *EquipmentA* to cover this issue. Enterprises have systems that collaborate, which they have control of or use at least to easily manage trading modifications and others in which such changes are unforeseen. Those constituents can be classified into two control groups inside the properties of an SoS. The first is comprised of systems with specific purposes, but no behavioral independence, whereas the other involves operationally and managerially independent systems whose behavior is not subordinated to the central managed purpose.

4.2.2.1 Establishment of a Blended Category for Systems-of-Systems

The four Maier's categories are often well differentiated from each other; however, in practice, the SoS borders cannot be well defined; therefore, distinct constituents (e.g., single constituents or complex systems as constituents) can be classified into several types of management and control in the same SoS (DAHMANN, 2015; NCUBE; LIM, 2018). Thereby, the strategies of management and control among the mission engineering team, SoS team, and each CS team to satisfy the needs of the SoS are distinct for a CS that collaborates between the directed and acknowledged levels. Thus, specific requirements and architectural structures are necessary for an intermediate SoS to consider OT and IT and their evolution over time. By definition, directed SoS are composed of CSs whose behavior works towards achieving the SoS goal. Such systems must be reliable, since they should be built and known by enterprises; however, according to the interviewees, most current CSs are not built by them, but by a third party, which attends their requests of adaptation or evolution when necessary[4]. On the other hand, enterprises have required behaviors neither exhibited by those systems nor provided by these CSs owners; but such capabilities are available in other CSs not controlled by the SoS central control. The interviewees have used a contractual relationship with CSs owners to make those capabilities available in their SoS; however, they are characteristics present in the acknowledged type, and changes are dependent upon negotiations between SoS central control and the CSs owners. The development, management, and interoperation of the architecture of the intermediate SoS makes it highly complex and challenges, since the interoperation is between OT and IT technologies.

Figure 18 shows a data-based solution, called Blended category, which is a deliberative SoS with a central control and a mission and SoS engineering team that builds the SoS towards to fulfill specific purposes. Therefore, central control is intended to maximize the success

[4] The systems are acquired by licenses and the so-called upgrades, or evolution, are carried out separately.

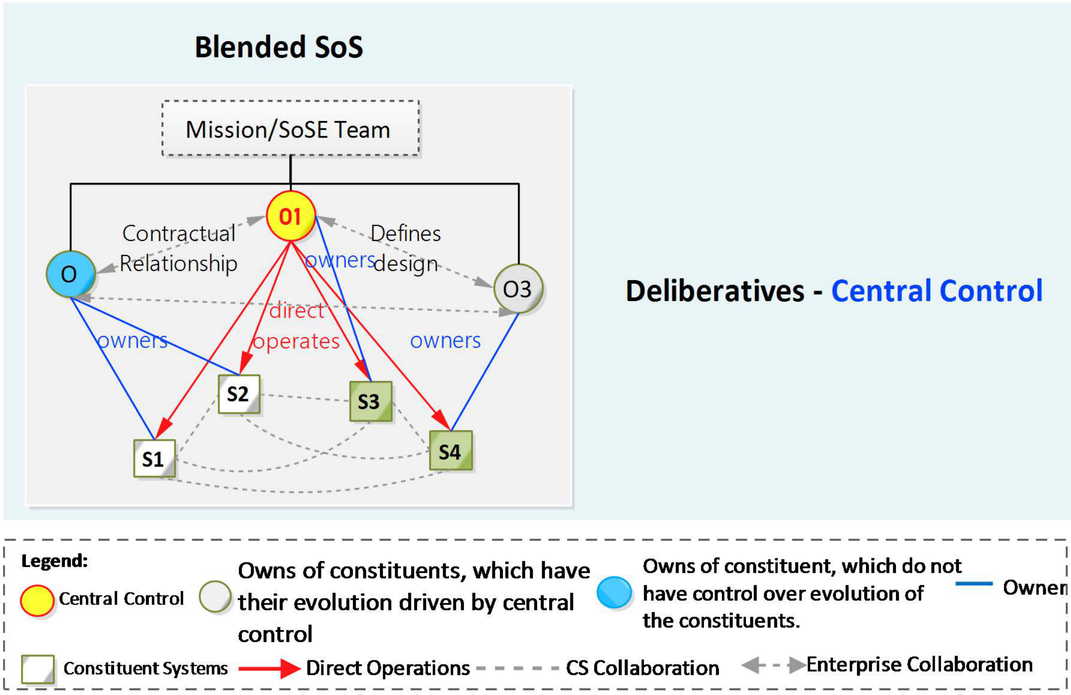


Figure 18 – Blended Systems-of-Systems: a theoretical SoS category

Source: Elaborated by the author.

towards achieving emergent behaviors, ensuring conditions for a continuous operation of the system. A contractual relationship with guarantees and demands must be established between the owner (i.e., O2) and the central entity (i.e., O1) to meet business objectives and supply of capability necessary for the accomplishment of the overall mission. S1 and S2 are constituents of operational and managerial independence, since their behavior is not subordinated to 01; however, they contribute to the achievement of the overall mission.

The greatest benefit of the blended category is increased flexibility to attend the expectations and needs of businesses that comply with mission needs, requirements, and capabilities emerged at runtime, environmental changes, and legal regulations for both OT and IT. However, the mission and SoS team will face significant challenges of adaptation for supporting converged architecture open-ended sustainability, evolvability, scalability, availability, reliability, and dependability. Moreover, the structure and analyses of technical infrastructure and technical interoperability requirements of OT and IT are even more challenging due to the trade-offs from the increase in complexity. The following specific properties were identified to differentiate blended SoS from directed and acknowledged SoS:

1. *SoS Mission*: the blended type concerns the end-to-end and the mission threat detected in the SoS development lifecycle, which can hinder the achievement of the SoS mission over time. Mission engineering concepts were applied to its planning.
2. *Stakeholders*: stakeholders in blended SoS are considered according to their decision

power and influence capabilities under SoS. Such property is related to the number of stakeholders of a blended SoS during its lifecycle. Therefore, they can be changed, and those that contribute to the development are maintained even if the roles are changed. Roles and responsibilities of stakeholders should be changed in the development lifecycle to improve contributions.

3. *Constituent systems*: these architectural elements represent systems that integrate an SoS. Constituents that compose this SoS can also be other SoS (e.g., satellite constellations, smart vehicles) or even OT such as an Internet of Things (e.g., wireless sensor networks (WSN) with sink nodes). Each constituent can be described in terms of their interfaces, i.e., the interaction points exposed to the environment. According to the interviewees, the interactions of those that comprise other SoS are much more critical and costly to interoperate.

4. *SoS Central*: the central control of a blended SoS does not have complete authority over the behavior of all constituents. Blended SoS was designed by a combination of constituents whose behavior is subordinated to the central entity and others that evolve independently.

5. *Interaction between platforms*: Blended SoS can facilitate the interaction between different platforms, since they enable more than one SoS as constituent.

Our aim is to work with the blended category, despite the overlaps of boundaries reported in the literature (DAHMANN, 2015; NCUBE; LIM, 2018). To the best of our knowledge, this is the first study that characterizes, describes, and designs a blended SoS with interoperability requirements derived from mission engineering. Other properties of blended SoS compared to Maier's types can be found in Section 4.2.2.2; Table 8.

4.2.2.2 Characterization of the Blended Systems-of-Systems

SoS are usually designed in an interactive and incremental fashion through the interoperation of constituents driven by their mission. Kazman *et al.* (2013) introduced three contexts of SoS development, called greenfield, brownfield, and closed source, and Madni and Sievers (2014a) described the following four types of systems that are designed to interoperate an SoS, known as:

- New systems, which are intentionally projected to be inserted into an existing SoS;
- New systems that collaborate in an SoS composed of individual systems and were not established to work together; and
- Constituent that cooperates *ad hoc* with each other without interoperating.

However, a more dynamic context is necessary for the design of the blended category and because of the `blended context`, which enables the combination of other types of interoperation usually found in domains with multiple regulations and operationally and managerially independent constituents. For instance, the structures of Flood Monitoring, agro4.0, and eHealth are comprised of newly designed constituents (i.e., Greenfield), altered constituents (i.e., Brownfield), and constituents as satellites, not accessed by the SoS for performing or requiring modifications (i.e., Closed Source). Several needs or opportunities will emerge at runtime due to the properties of the emergent behavior, adaptability, and evolutionary development, and must be partitioned into smaller parts to be fulfilled by constituents (JAMSHIDI, 2009).

In a top-down view, needs can also lead to the addition of new constituents, their development, or improvement of an existing one to meet that needed capability. The blended category has a high level of distributed systems and its architecture must display properties similar to those of self-healing architecture, which comprises reliability and robustness to hardware and software failures, resilience, return to the original form when a failure occurs in the components of CSs and spreads to several parts of the systems and affects the SoS capabilities. Figure 19 shows graphically how blended type is compared with the initial types proposed by Maier (1996) and Dahmann and Baldwin (2008). Whereas Table 8 displays the general characteristics of

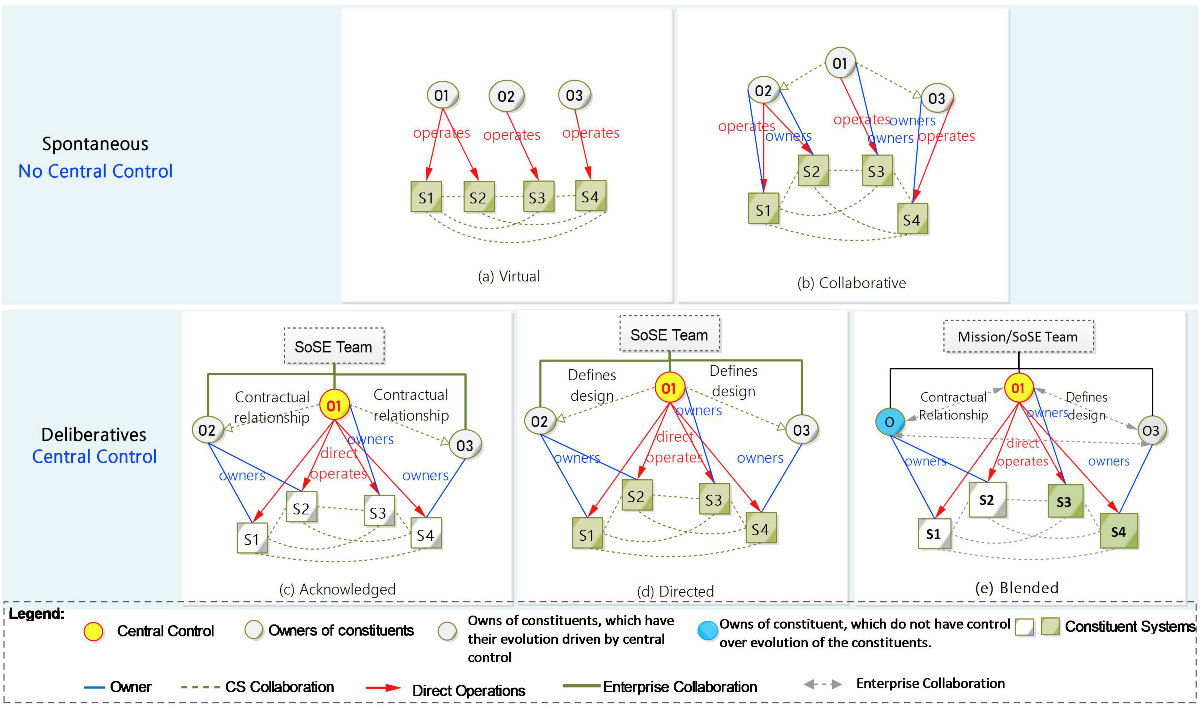


Figure 19 – Systems-of-Systems types: a comparison with theoretical category

Source: Elaborated by the author.

the blended category and the relation to other categories (i.e., virtual, collaborative, directed, and acknowledged). The table designed by Ki-Aries *et al.* (2018) is composed of eight aspects (i.e., description, stakeholders involvement, governance, operational focus, acquisition, test & evaluation, boundaries & interfaces, and performance & behavior), described for four Maier's categories. We have adapted them and added six others, i.e., purpose, mission, goal, objectives, requirements, and maintenance & change mechanism to all categories, as well as addressed the whole aspects to the blended category. A blended SoS has been described as an interdesigned management with a central control that does not have complete authority over the constituents. Therefore, an increase in complexity, heterogeneity, interconnectedness, and asynchronous cycles between the independent systems that comprise an SoS, as resources of hardware (e.g., mobile devices, servers, network communication, etc), software (e.g., services, mediators, application of services or micro-services, etc), and generally, the relationship among the owners of the organizations involved make the detection of interoperation threats over the lifetime of SoS more challenging (AL-NASHIF; MYTHILI, 2007). Therefore, properties of autonomic computing, continuous availability, dependability, and survivability are essential for the monitoring of the behavior of this category at runtime and to maintain the operation flow (PSAIER; DUSTDAR, 2011). For instance, during the sugarcane production process, the entire length of the field must be met. Processes from previous activities, such as nutrients, moisture, pests, and diseases, must be constantly monitored, Since they are paramount to the evolution of the crop and harvest decisions (e.g., maturity point), that is, preventing the crop from reaching flowering reduces both sugar loss and loss of value of the product delivered to the industry.

Adaptability in blended category complies with other categories. Its root is related to the identification of behaviors and the implementation of a characteristic for a specific occurrence by executing some operations at runtime, which were described by mission engineering or SoS team (i.e., foreseen or proactive), or unplanned behaviors (i.e., unforeseen or reactive) may still happen (WERMELINGER, 1999; Bradbury, J. S., 2004). This occurs because, in the development universe of SoS, we have many types of emergence, such as ontological and epistemological. The former is an unforeseen (i.e., strong and spook) emergence which produces a novel behavior, whereas the latter can be considered foreseen (i.e., simple and weak) at design time and arises from the lack of knowledge on the system (TOLK, 2019). For example, thinking of interoperability for a harvest emergence behavior (e.g., two tractors) for the end-user leads to epistemological emergence, and we at least know the capabilities, requirements, boundary, and metrics that we are interested in. On the other hand, for an SoS participating in another SoS as constituent, the emergence can be changeable as it will occur at both the constituent and SoS levels. Such needs that emerge at runtime must be identified, classified and partitioned in smaller parts to be negotiated with the owners of the constituent (JAMSHIDI, 2009).

In a blended SoS, its mission requires complete knowledge of the business objective and operational capability that will be met by such system, as well as knowledge of the owner organizations of the constituents. Stakeholders are identified to both the SoS and constituents level and normally have interest and inconsistent priorities. Most of stakeholders are likely to be known to constituents managed by a central control. Otherwise, their identification can be challenging. In this category, only the main stakeholders are sought for; they have decision power and influence capabilities under SoS (CHUNG; CRAWFORD, 2016; ROWLEY, 1997). For instance, consider the SoS A with a group of stakeholders $\{1,2,3,...n\}$; $(1 \leq stakeholders \leq n)$.

Table 8 – Characterising Blended category

White color - refers to Ki-Aries *et al.* (2018) definitions. **Yellow color** - refers to adaptations from Ki-Aries *et al.* (2018). **Blue color** - refers to new definitions Source: Adapted from Ki-Aries *et al.* (2018).

Characterization of Types of Systems-of-Systems						
Types	Aspects	Deliberatives			Spontaneous	
		Directed SoS	Acknowledged SoS	Blended SoS	Collaborative SoS	Virtual SoS
SoS Categories	Description	A Directed SoS can be described as possessing "interrelated collaboration, with central control management, operation and control over the SoS as a whole"	An Acknowledged SoS has "designated management, but limited control over the independent collaboration of the SoS".	A Blended SoS has an interdesigned management with a central control, but no complete authority over constituents.	A Collaborative SoS has "no central of management , therefore, operational and control must be formed and agreed as mutual independent collaboration".	A Virtual SoS develops "individual independent collaboration with no central management, operation or control of the SoS as a whole".
Management and Oversight	Stakeholders Engagement	•Stakeholders are at system and SoS levels; •Interrelated independent system owners; •Some competing interests and priorities; •May have limited interest in the SoS; •Most stakeholders are likely to be recognized.	•Stakeholders are at system and SoS levels; • independent systems owners; • Competing interests and priorities; •Possible interest in the SoS; • Some stakeholders may not be recognized.	•Stakeholders are at system and SoS levels; • Interrelated and independent systems owners; • Some competing interests and priorities; •Possible interest in the SoS; • Some stakeholders may not be recognized.	•Stakeholders are at system level and collaborate mutually at the SoS levels; • independent systems owners; • Competing interests and priorities; Possible interest in the SoS; • Some stakeholders may not be recognized.	•Stakeholders are at system and SoS levels; • Independent systems owners may not develop direct interactive collaboration; • May show no vested interest in the SoS or Constituents; •Many stakeholders may not be recognized -
	Management	•Some levels of complexity with central management and funding for both SoS and interrelated collaboration of constituents; •SoS has authority over all the constituents.	• Adds level of complexity due to designated management and funding for both SoS and individual systems. •With independent collaboration, the SoS does not have authority over all the system;	•Increased levels of complexity due to interdesigned management and funding for the interrelated collaboration of SoS and individual systems. •SoS is comprised of constituents it controls to evolving and others it it does not control	• Further levels of complexity due to the mutual independent collaboration of SoS management with funding only at or from individual system level • SoS has no authority over the systems	•Increased levels of complexity and uncertainty due to no central management and funding for SoS limited to individual system level; • System do not have authority over the SoS as a whole.
Operational Environment	Operational Focus	•Directed collaboration toowards the meeting of a set of operational objectives;	• Designated collaboration towards the meeting of a set of operational objectives;	•interdesigned collaboration for the meeting of a set of operational objectives	• Mutually agreed collaboration towards the meeting of a set of operational objectives;	• Individually aligned towards the meeting of a set of operational objectives
	Purpose	• Constituents' purpose is usually further aligned with the SoS' purpose	• Constituents' purpose may not be known and aligned with the SoS' purpose	• Some constituents' purpose are usually more align with SoS those of SoS and constituents' purpose that may not be known and neither align with the SoS' purpose	• Constituents' purpose are aligned with the SoS' purpose	• Constituents' purpose are not known and the SoS level lacks agreement
	Mission	•Constituents' missions may or may not align with the SoS' mission	•Constituents' missions are not aligned with the SoS' mission	•Constituents' missions may or may not align with the SoS' mission	•Constituents' missions are generally not aligned with the SoS' mission, but they shared it.	•Constituents' missions are generally not aligned with the SoS' mission
	Goal	•SoS' goals and constituents' goals are not clearly defined	•The goals SoS level and constituent level are not goal clearly recognized	•High-level goals are vaguely stated for both SoS and constituents levels.	•No clear goal statement	No clear goal statement
	Objectives	•Both, SoS level and constituent level have clearly defined objectives	• There are recognize objectives	•A combination of recognize and clear objectives is designed to SoS level.	•No clear objectives	No clear objectives

Table 7 – Characterising Blended category (Continuation)

White color - refers to Ki-Aries *et al.* (2018) definitions. **Blue color** - refers to new definitions
Source: Adapted from Ki-Aries *et al.* (2018).

Characterising Mission Engieering-Driven Systems-of-Systems						
Types	Aspects	Deliberatives			Spontaneous	
		Directed SoS	Acknowledged SoS	Blended SoS	Collaborative SoS	Virtual SoS
SoS Design	Acquisition	• Complexity from multiple systems lifecycles, new developments, technologies, acquisition programs, developmental, and legacy systems; • Stated capability objectives up-front, which may provide basis for requirements; • Benefits from the central control for the establishment, integration and interoperation of constituents needs	• Complexity from multiple systems lifecycles, new developments, technologies, acquisition programs, developmental, and legacy systems; • Stated capability objectives up-front, which may provide basis for requirements; • Designated management and independent system needs are established	• Complexity from multiple systems lifecycles, new developments, technologies, acquisition programs, developmental, and legacy systems; •Mission objectives determined upfront and updated continually, which may provide basis for requirements end-to-end; •Interdesigned management with benefits from central control to establish, integrate and interoperate independent systems need are established	• Complexity from multiple systems lifecycles, new developments, technologies, acquisition programs, developmental, and legacy systems; • Stated capability objectives up-front, which may provide basis for requirements; • Mutually agreed independent system needs are established	• Complexity from multiple systems lifecycles, new developments, technologies, acquisition programs, developmental, and legacy systems; • Stated capability objectives based on limited needs may be noted up-front, which may provide basis for requirements; • Individual independent system needs may not establish needs of others systems
	Requirements	• A single, large, and dominant organisation must play a requirements engineering leadership role, and define both SoS-Level requirements and constituent Systems Level requirements. • The central SoS authority must support the controls, mandates and direct the lifecycles of the constituent systems. • The requirements evolution of both, SoS and constituents levels are controlled and coordinated by the central authority. In this type of SoS classical requirements engineering approaches, methods, techniques, and tools may apply.	•The constituents are owned and operated by different organizations, and requirements engineering to SoS' level is performed by central authority and address requirements across the SoS. •Requirements engineering for each constituent is performed independently and addresses requirements from their owner's perspective with independent life cycles. •Requirements engineering is performed independently by the SoS owner and constituents with minimal collaboration.	•Requirements engineering is performed by mission/Sos team which define both SoS level requirements and some constituents systems level requirements •The central control have authority only over some constituents. So, there are constituents that have independent lifecycle of requirements engineering; •The evolution of an Blended SoS are controlled and management by the central control for SoS level and by some constituents level. To constituents independents managerial and operational the new implementation to SoS level need be negotiated, but minimal structure are provided to achievement of the global mission.	•There is no single dominant organisation which performs the leadership role for SoS' level requirements engineering. •Requirements engineering is performed by each constituent independently. •There are not one group owns the SoS, there is no obvious group that owns its requirements.	•The SoS purposes are dynamic and requirements change frequently. There is no central requirement engineering authority. •There is informal and irregular, if at all, requirements engineering at constituents level. •The SoS will be composed and the participating constituents cannot be known in advance. Requirements engineering at design time cannot be apply.
	Test & Evaluation	• Constituents of a same SoS often have asynchronous lifecycles • Complexity of all the moving parts and potential for unintended consequences	• More challenging due to the difficulty of synchronising multiples' life cycles; • Complexity of all the moving parts and potencial for unintended consequences.	•The challenging is great and complex since constituents have several lifecycle and difficulty to synchronising them • Complexity of all the moving parts and potential for unintended consequences.	• Complete testing is more challenging due to the difficulty of synchronising across multiples system's lifecycles; • Complexity of all the moving parts and potential for unintended consequences.	• Testing cannot be completed in full and is challenge due to the difficulty of synchronising across multiples system´life cycles; • Limited access and complexity of all the moving parts and potential for unintended consequences.
	Maintenance & Change mechanism	• Constituents maintain an ability to operate independently, but each one is subordinated to the central authority	• Changes in the systems are based on the collaboration between SoS and constituents;	• Changes are based on two principles: (i) subordinated to central control or (ii) collaboration between SoS and constituents	• Enforcement and maintenance of standards	• Constituents are relatively invisible to one another.
Engineering and Design	Boundaries & Interface	• Focus is on identifying the constituents within direct management and control that contribute to the SoS mission, functionality and data flow;	• Focus is on identifying the constituents within designated management and control that contribute to the SoS mission, functionality and data flow;	• Focus is on identifying the constituents within interdesigned management and control that contribute to the SoS mission, functionality and data flow;	• Focus is on identifying the constituents and mutually agreed management and control that contribute to the SoS mission, functionality and data flow;	• Focus is on identifying the constituents and expected indirect collaboration and control that contribute to the SoS mission, functionality and data flow;
	Performance & Behavior	• Directly managed and monitored at SoS level to satisfy SoS use needs • Balancing needs of the systems benefits from direct co-ordination	• Dedignated management and monitoring at SoS level and system levels to satisfy SoS use needs; • Balancing needs of the systems benefits from designated co-ordination	• Interdesigned management and monitoring at SoS level and constituent level to satisfy SoS use needs; • Balancing needs of the systems benefits from interdesigned co-ordination	• Mutually agreed management and monitoring at systems level to satisfy SoS use needs; • Balancing needs of all systems is reliant on mutual co-ordination	• Direct and indirect management and monitoring at system levels to satisfy SoS use needs; • Balancing needs of the systems and indirect systems may not be achieved.

The stakeholders network analysis recognizes that customer 10 and end-user 1000 have great influence over other people that receive the value provided by this SoS or over the decision-making, due to the influence that both have on the stakeholders network. If the needs of 10 or 1000 stakeholders are not met, in their needs, the whole SoS and the organizations involved can be negatively impacted, which includes their lack of engagement in the lifecycle of the systems.

Blended implementation can use approaches based on mission engineering, SoS engineering, or only on SoS engineering. Its constituents involve a large number of other SoS or complex systems. A mission-driven approach offers a holistic view of the cooperation of organizations and the constituents collaborating. Its design can examine the outinside-in development model (SAEED *et al.*, 2015) and the concept of experience-based value (SCHMITT, 2011), which were the first to consider the value, from general to specific, provided by the SoS to each level keeping the focus on the user's need. Moreover, the requirements engineering process is developed in both levels by the mission/Sos team which defines both SoS level requirements and some constituent systems level requirements. . The evolution of a blended SoS is controlled and managed by the central control of the SoS level and by some constituent level. A new implementation to the SoS level must be negotiated for managerially and operationally independent constituents; however, a minimal structure is provided for the achievement of the global mission. After the identification of the capabilities and the acquisition policy normally found in ConMisOps (i.e., concept of SoS mission operations), the constituents can be selected and gathered to collaborate (cf. Section 4.2.10).

The business process provides capabilities and the constituents that can collaborate to the SoS mission and support mission objective are identified. Due to the high complexity of an SoS, different views of requirements should be applied from the early stages towards facilitating interaction, communication, and representation of several parts of systems. Therefore, the stakeholder heterogeneity should contemplate the requirements architecture constituted of those views for a better comprehension of the needs, capabilities, and requirements. Such views have been applied to requirements for traditional systems in both academia and industry to support the different aspects of systems and better understand them towards reducing stakeholders and end-user dissatisfaction (NUSEIBEH *et al.*, 1994; LORMANS *et al.*, 2008; DAUN *et al.*, 2012). They are essential for the blended SoS level, facilitating conflict resolution, forward and backward traceability and the initial establishment of the architecture.

Furthermore, the architecture of those systems must be open so that software and hardware become independent and interoperate through mediators for entire levels, which enables the choice of the best constituents without concerns with the enterprise that must cooperate and the constituents that must collaborate with SoS blended (STEVENS, 2008; KRIESBERG, 2012). Changes and dynamic evolution are the most frequent due to the diversity and autonomy of constituents at various levels. In many cases, constituent redundancy is mandatory, despite its impact on the performance. Both redundancy and overlap in mediators can improve the query performance by selecting data retrieved for end-user and maintaining through scrutiny and combination of information from most efficient sources to fulfill the SoS mission (VASSALOS; PAPAKONSTANTINOU, 1998).

4.2.3 Divergence of the Data and Communication Channel

Our aim was to analyze incidents that contributed positively or negatively to the technical infrastructure of data and communication. The reason to perform interoperability analysis is that technical interoperability requirements have been important in institutions to the strategical, tactical, and operational levels. The refinement of the categories showed a significant divergence between the current infrastructure per domain, besides satisfaction of the customer and user related to the several incidents that they brought up.

Many of such incidents have posed challenges for different domains (e.g., Network availability for the agroindustry, which limits the access of institutions to on-field Internet, thus precluding the establishment of an SoS). Other incidents, such as `data latency`, `shared application`, `data format`, `transmission rates`, (cf.Appendix M) also contribute to the related problems. However, the advent of industry 4.0 is a reality and the applicability of technology is required in whole areas to improve productivity and competitiveness in the market. The following statement illustrates such issues.

> *[...] We have areas of no converge; we must work using an offline system, when available in a network area, the data are transferred to the servers. In the process of defining what was going to be done, we must consider, besides the robust ones, an important performance for a volume of data exchanged that we will need to have. But, it is not only that, it is linked to a reengineering that needed to be done, analysis of transmitters to be designed, so that we are able to work as it is today. [...] In an infrastructure, engineering must look not only at what we already know, but also at planning what is to come and how to work in the long run (INTERVIEWEE IT_22).*

Regarding the infrastructure needs within the process of technical interoperability requirement analysis driven by mission engineering, an analysis of the situation must be conducted prior to the identification of each capacity together with stakeholders. Although the knowledge of the domain is necessary, it is not enough for the identification of the real needs, which differ from organization to organization. The structure of some domains for countries like Brazil, such as avionics and telecommunication, is better organized than that of agriculture and health, which are still in the structuring process to better apply the technologies in an interoperated manner.

4.2.4 Technical Interoperability Quality

The analysis revealed the quality attributes, of which are not achieved hierarchically, considered relevant to the technical environment. Performance is more relevant to the context of an SoS, although dependent on different factors (e.g., reliability and performance of constituents, security, fault tolerance of network, among others). Therefore, a higher-level quality attribute may depend on several non-hierarchical attributes of the lower levels, differently from what is represented in ISO/IEC-25010 (2011). The incidents identified were organized into three subcategories of codes, namely technical infrastructure, software, and hardware. Quality attributes common to all infrastructures were classified as `technical infrastructure`, those related to software were organized into `software`, and others were grouped into `hardware` (cf. Appendix M).

The attributes were characterized according to the interviewed population, i.e., industrial workers or academics developing industrial projects. The nomenclature adopted for each requirement of the two communities was evaluated and is provided in Section 5.4.2.3. According to our experience, an interoperability requirement is considered a quality attribute by the academia; however, the industry often classifies it as a technical requirement. This analysis provides both justifications, illustrated by the following statements:

> *I consider it a quality attribute, because, sometimes, a high level of interoperability is not a mandatory requirement at first moment, but it can certainly drive the ability to evolve an ecosystem of systems, since it hampers integration with new technologies and systems that still have not been built. Therefore, interoperability is a business driver and an important quality attribute to be analyzed during the evaluation of a system (INTERVIEWEE ID9).*

> *[Quality Attribute] The communication between SoS is based on the interoperability of each level (functional, data model, communication protocol) (INTERVIEWEE ID35).*

> *Interoperability is the ability of a system to perform with other systems, which makes it an attribute, rather than a technical requirement. A technical requirement implies the system must be interoperable with respect to a specified set of systems (INTERVIEWEE ID50).*

> *At some point in the project, it is a quality attribute, for instance, during the requirements analysis phase. Interoperability is an orthogonal issue, and that is why it must be considered a quality attribute. But sometimes people need to look at interoperability as a technical requirement due to its impact on the whole architecture; it must be specified according to all architectural layers, so that all stakeholders have the proper point of view on the issue (INTERVIEWEE ID27).*

> *Technical requirements enable the exchange of information, and the quality attribute refers to the correct use of this information (INTERVIEWEE ID47).*

In general, we agree with the interviewees' viewpoint that both quality attributes and technical requirements are relevant for the SoS development. According to them, requirements are essential to the entire lifecycle and some of them are even analyzed at runtime (e.g., temperature, energy consumption, and performance efficiency of some sensors). Some are considered priorities for a business objective and show availability (e.g., network, systems, and data), security, traceability, performance, maintainability, simplicity, compatibility, response time, reuse of resource, recoverability, robustness, high-performance, scalability, sustainability, etc. The generalization of the quality attributes considered the domains involved, although some specific technical requirements may not have been raised by our population, and may be relevant for some other domains or even for an organization. An example is the agroindustry domain, which involves more than one organization with diversified standards for technical requirement `platform scalability`, of high relevance for operational market strategies (this requirement may not even be mentioned in another domain).

> *Today we have information security as main item, it is very evident and this comes from strategies, it has the ability to integrate the evolution system and also the platform evolution, it is something that we are starting the analysis process for understanding. Another important requirement not considered in the past is partnership, which refers to knowing partner, the strategy, and collaborating. Beyond the technical scalability and business scalability. It is what has knocked on our door today [...] (INTERVIEWEE IT_11).*

4.2.5 Sustainable Technical Interoperability

The `Sustainable Technical Interoperability` category varies according to the level of requirements identification, and is comprised of five viewpoints and one dilemma considered most common by our population.

(i) **dilemma standardization versus openness:** refers to difficulty in interoperating individual systems developed using different technologies. Lack of standardization enables the emergence of multiple solutions to the same market segment, which are unable to interoperate with each other. New technologies are often designed and sometimes the solutions developed are of interest; however, the two points evaluated by interviewees, interaction with the existing systems and the risk of implanting a new technology in critical domains when they must have everything working all the time, as shown in the example below.

> *We had two solutions, one provided by a startup and the other by a comforting company and we opted, despite being more expensive for the large company. The startup's solution was technically better, flexible for*

> *changes and updates. However, it was not chosen, because we must of a high level of security, it was not possible to risk with new technology. From the moment that any failure occurs, invasions allow access to all distributed companies and access to the company's strategic processes. So, the decision was to increase the cost, reduce the flexibility and concern with risk that was the most critical (INTERVIEWEE IT_10)*

Standardization is necessary and helps to ensure some of the important technical requirements such as safety, interoperability, and compatibility, but when it is combined with openness, it increases flexibility and competitiveness. Figure 20 shows how the combination of standardization and opening goes further than the possibility of free modification. It allows the creation of a new business model, a new way of developing technology, with open interfaces, which facilitates data exchange and development of the highest rates of innovation and portability of systems to improve the technical requirements applied to the coalition processes.

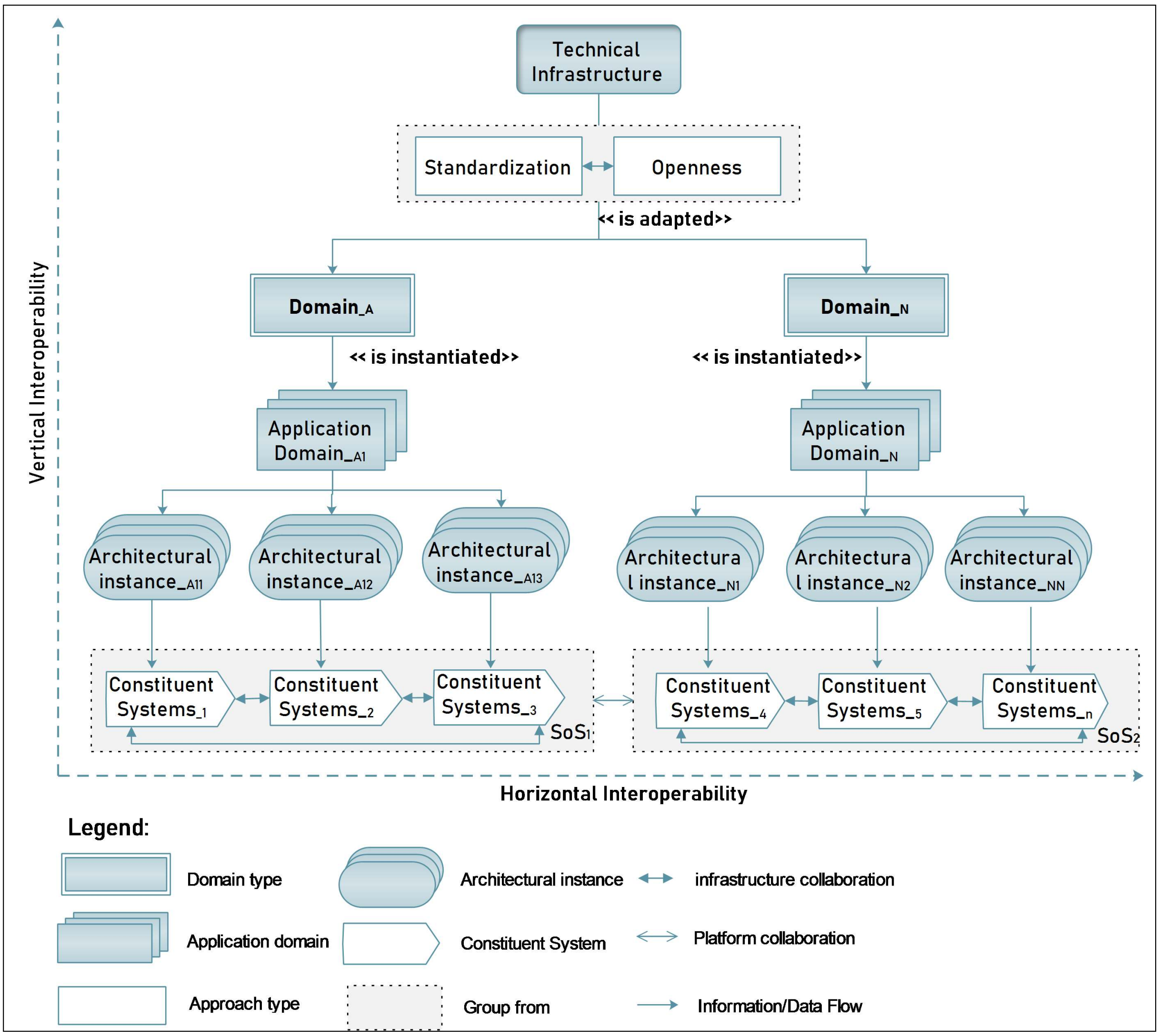


Figure 20 – Overall view of the generic model for technical Infrastructure

Source: Elaborated by the author.

(ii) **social perspective**: concerns the relationship between stakeholder and end-user, management of their needs and collaboration between the partners. According to interviewees, the technical interoperability requirements, and the whole technology, should support the enterprises to better decision-making at multiple organizational levels using the same set of data. For them, distributed organizations that have central control for each unit and a unit that centralizes the others have particular challenges to maintain the stakeholders engaged during the whole process of development of any system or activity. However, the engagement of the stakeholders gives more transparency to the decision-making process ensuring collaboration and feedback to the improvement of the system and further control of the other processes and achievement of business objectives, as the following statement demonstrates.

> *In this relationship, again it is technology serving as a means to simply increase reliability, which is one of the most important items today when you assume that you improve knowledge of the operating environment, providing traceability of the raw material of the product, for these final customers this is a great gain because the quality is measured at the time of delivery. However, when that customer places a purchase order and he can track the batch production specifications in real-time, it will certainly be a differential for customer loyalty if we attend to concerns that are important to him, such as sustainability, care for people, etc. [...] (INTERVIEWEE IT_16).*

During the analysis, it was possible to perceive that a diversity of stakeholders found in the highly complex environment, which have different cultures, concerns, policies, needs, and conflicts, still impose a great challenge for all domains. For them, an activity more relevant than identifying the interested stakeholders in the system is to recognize those who have influence and decision-making power. What are their roles and responsibilities? How does each one influence the development process and the fulfillment of the mission requirements? Which stakeholders are interested in each system, or still in the data, information, knowledge or services provided by the interoperation of these systems? How do they define the strategy of stakeholders change throughout the development lifecycle? These pieces of information are essential to interviewees to establish business strategies, to determine performance measurement to the business goal and to plan how to provide further experience instead of only value to the end-user.

In addition, there is an intrinsic concern related to keeping collaboration among many partner enterprises (e.g., system owners). These collaborations increase value to everyone involved in the collaboration and improve some processes such as the negotiation between these organizations. In the collaboration process, they often seek to improve the capacity provided by the system by using new technologies, modified ones or some evolved to software, hardware or infrastructure. However, there is a great concern to maintain the health and safety of technical and non-technical resources.

(iii) **economic perspective**: addresses an overview of the experience and challenges have faced to maintain their assets in an environment of constant changes as that of industry 4.0. In the context of the data gathered for MEDINA, the first challenge is related to planning the interoperation design (i.e., a coalition of individual systems). The domain of the technology 4.0 is emergent and what is functional for one domain will not necessarily be so for another. Hence, they have a network infrastructure problem that not always is available or needs to be implemented to different necessities. Interoperation cost is higher and hard to keep within the budget planning, but that investment is necessary. New strategy planning should be performed or adjusted to meet the new market requirements, directing transaction cost, equipment maintenance cost, managing the reuse and sharing of resources to reduce cost and improve the risk analysis so that they are able to improve the value provided and that of their own enterprise. Increasing profits, by providing services and products with more transparency in their produced processes will retain their customers and end-users. The following statement illustrates this case.

> *[...] It is necessary to be aware of the constant changes in the market and use the right strategies to be able to remain competitive, otherwise, other organization uses new technology, provides faster service or a product with some novelty and you are left behind (INTERVIEWEE ID18).*

On the other hand, the current market brings multiple stakeholders with varied needs as discussed in social perspective. In this context, understanding these needs and mapping how they will be met are also a challenge to them. This is why, only some mission needs will be met by a system functionality, and many others are fulfilled by different activities. It is essential, in this process, to comprehend the mission that the system needs to solve and the quality concerns to be evaluated and how these concerns will be measured during the operation of the system. In general, everyone agrees that the first requirements to be analyzed are functional appropriateness to technical infrastructures and to systems, following by functional completeness, functional correctness, due to the fact that the technical part (e.g., network, hardware, and software) needs to be working and available to accomplish the interoperation. They need to be able to deliver the correct capability and any other requirements can only be assessed after these are working properly. The following statements illustrate this issue.

> *The first step is to define where you want to go, that is, the mission you want to accomplish, depending on the decision-making can be modified and the infrastructure required to meet business goals will be different. Therefore, for us, the alignment of the industry's strategy to the agricultural sector is essential to fulfill our mission. Particularly, there is still a great difficulty to have the systems interoperated for the field, since there is a complete lack of communication infrastructure necessary for minimal connectivity. The field still has many barriers that brings different challenges for the implementation of technologies in industry 4.0, which today have become indispensable. Promoting interoperation in the field gives us more flexibility since all of our fields and units are distributed; improves the decision-making and their accuracy; improve*

> *the quality of our data and reduce rework by introducing redundant data in many systems. In addition to providing a holistic and strategic vision of all areas and levels of the companies. [...] (INTERVIEWEE IT_22).*

> *[...] Sustainability is a very difficult requirement to measure both in production and for the technology which we use, but the concept of sustainability of maintenance of resources over time is what we use as one of the main indicators today in our organization. If you talk about systemic or network availability, then you have to have these resources working today. For example, there is nothing that gives us more problems than an offline email provider, because it is the organization's main means of communication. For questions of standards, we are also working on the issue of information security, we have undergone external audits that follow specific rules and protocols that until recently did not have this type of necessary relevance (INTERVIEWEE IT_23).*

(iv) **technical perspective**: regards the usage of resources over time. An important requirement for this perspective is reliability, focused on availability, fault tolerance, resilience, robustness, correctness, and recoverability. These are results that will contribute to better the use of resources for decision-making and provision of services to the end-user. For example, when respondents have a network structure that has their maintenance planning, they can work for a long time to fulfill their mission without causing damage to any user inside or outside the company or group of users. Furthermore, the needs for extension or evolution of the technical infrastructure can be identified in advance, before the problem arises, especially if there are contractual agreements between central control and system owner. This statement illustrates the development of a network and the challenges faced by blind spots that, for a long time, may lead to important modifications in the network infrastructure or technology.

> *[...] Our network was measured and designed to meet the entire extension demanded, and all blind spots were identified. However, the entry of new antennas or repeaters were covered by these points. To avoid running out of information at any time due to a network problem, all devices work offline. Thus, the information is archived locally and as soon as it has access to an available connection it can be transmitted, but if it loses the connection, the data is safe. We cannot risk having a data range to decide. An interval of data failure due to a problem in our infrastructure. Failure of a sensor, for example, can lead to loss of assets and even loss of life [...] (INTERVIEWEE IT_9).*

In the technical perspective, sustainability also considers the interaction between systems and the quality of the information that is being produced in these interoperations as a result of design decisions, in order to analyze technical interoperability requirements. From this perspective, problems such as accessibility, durability, and reliability of data, usefulness, and security of information must be supported by simulation at design time and attend at runtime due to the inherent properties of SoS, which make it dynamic for the environment all the time.

Associated with data, the hardware is just as important. Today, systems are distributed to process large amounts of data on multifunctional devices with concurrent users. Sometimes the network and the software are available, but they have problems with the hardware infrastructure, which can occur in different circumstances: (i) the hardware was chosen with the wrong capacity; (ii) the hardware was chosen correctly, but the volume of users or data increased and it did not support it. At the end, the analysis of the platform for evolution and interoperation has been the most recent challenge, because they need to understand a new environment in which many are starting, others are adapting to it, and some are working in an interoperable way at the SoS level.

(v) **environmental and individual perspective**: In this section, we address the two last perspectives. The former perspective regards the conservation of the natural resources and entire domains needed to comply with some regulations of protection or sustainability of these resources. Particularly in the agricultural domain, this is a critical perspective with direct dependence on the climate and natural ecosystems. For the interviewees, some important attributes of quality are integrity of resources, mitigation of environmental risks, reuse and time behavior so that, for a long period of time, it is possible to keep activities happening for a long time. Another example is aviation, which has a direct influence on climate change, noise and health effects at airports, pollution and so on. In such cases, besides sustainability plans to reduce problems, they promote improvements in end-user awareness when choosing sustainable enterprises, since the activities described above are distributed in technical activities to the operation of these domains. The following statement shows how interoperation helps not only in technical processes, but also throughout the organizational structure.

> *We have a large number of regulations to comply with, two examples are related to our employees and their health, as well as environmental sustainability, with special attention to springs. When we do not have real-time control, we may have a problem with the logistics of our employees and the management of each stakeholder in the operational processes. (INTERVIEWEE IT_17).*

The latter perspective refers to the care for human life, promotion of well-being, leisure, mental and physical health, personal and professional motivation, respect and self-respect, support to activities, etc. Interviewees discuss efforts concerning cooperation and collaboration between systems and remember that one of the organization's greatest assets is still are people and many of them are to be considered constituents of an SoS. Because of this, they seek to encourage cooperation and collaboration through autonomy, adaptability, reducing waste and cost, mitigating the health and safety risks and increasing the responsive time (i.e., faster delivery of activities that involve a large dependence on humans).

Therefore, when a collaborator is happy, motivated, and healthy, the productivity level grows or at least increases compared to opposite situations, improving the financial health of companies. Interviewees have highlighted that only the quality of a product, service, or system is no longer as a competitive factor for the market, they consider quality as mandatory factor with pre-defined metrics that need to be fulfilled at the end of the production or design process as part of a contract. The results of an organization are directly related to its production chain and how it can continue to exist over time for the most diverse dimensions. For them, much more than understanding the processes of an organization, it is necessary to be aware of the current organizational needs and know which strategies can be applied to the solution process connecting humans and technologies.

4.2.6 Strategical decisions

The first interviews revealed that the interviewees sought to assess business alignment and misalignment, some addressed and discussed the necessity to work with the problem space, others with solution space and there were some that addressed both. Considering the problematic space of the analysis of technical interoperability requirements, the difficulty is the same as those of other areas and the time to start the analysis is more far-reaching. We organized the codes identified during the data analysis in two sets, namely (i) design time viewpoint, and (ii) runtime viewpoint. The viewpoint at design time regards the understanding of the statement of stakeholders' needs or interaction with the organization responsible for the system. Through comprehension of the goal of the business and its capabilities, tasks can be specified, value and product are also inserted in the portfolio. Strategy as a simulation for testing and particular architectural concerns have already been planned.

According to interviewees, problem space is the phase of structure that still needs to improve in the operations level at runtime, such as communication, connection, data exchange, semantic and syntactic data, the necessity of hardware, new capabilities or physical or human resources. They understood that between problem space and solution space during the execution process there is a new space that is neither just a problem, nor just a solution, but a mixture of both. In summary, the problem space is permeated by the following codes: `find problem target`, `business objectives analysis`, `statement of needs`, `comply with policy, regulations, and laws`, `set of capability`, `define product and value`, `measure performance for multilevel`, `business goal versus systems goal`, `simulation environments`, and `collaboration among organizations`.

Regarding at runtime viewpoint, the planning process is related to working in various scenarios and to managing to unforeseen change at design time. A point that stands out in the concept of strategic decisions is the capability of data exchange and the possibility to apply the knowledge taken from these data to multiple levels for decision-making. Indeed, all activities take place in an operational environment and many decisions will be changed according to the environmental change. For the interviewees, the quality of these decisions is directly linked to their capability to monitor operational tasks and to improve the business procedures. An interviewee said: *Knowing what our capacity needs are, we can easily identify* *which systems are needed or which hardware, equipment or machine are meeting or not meeting our quality concerns (Interviewee I_T18)*. They comprehend the capabilities of SoS and the need for robustness, open and flexible architecture to support the various modifications imposed, or external changes or the scalability of the business itself.

Because of this, architectural decisions are strategical within of the analysis of technical interoperability requirements and these decision must be identified in whole organizational levels. When they have regulations, laws, or policies for the SoS to comply with, it will be considered as a technical requirements. The same will hold true for constraints that are related to their domain. For instance, the application domain of the Agro4.0 always has the humidity and precipitation data available as a constraint and normally has to use Wi-Fi and off-line network because of the blind spots. Thus, support is necessary to decide, for example, which systems type to choose, which architectural pattern should be better (e.g., microkernel, clientserver, event-driven, n-tier, and microservices), how many mediators are enough to maintain the communication among constituents, data presentation exchange, how the services will be provided (e.g., adopt directory service, and strategy to interoperate new systems with legacy systems).

Indeed, theorizing from the analysis of technical interoperability requirements driven by mission engineering helps to understand the challenges from mission needs to technical interoperability requirements through situational analysis of the organization and technical stakeholders, which are present in this phase and not only in the architectural one, when these requirements are gathered and analyzed. For this reason, the architectural decision must begin before the architectural phase. There must also be a new space between problem space and solution space, which is the integration of both, called `interaction space`. Therefore, the strategic decision category can be considered as a part of all organizations, and all activities of SoS development.

4.2.6.1 Requirements as bridge between Mission and Architecture of Systems-of-Systems

Based on the assumption discussed in the concept of `Strategical decision`, we propose, for the SoS level, an interleaving between the activities of requirements with architecture and mission to describe an interactive and incremental fashion of the development of SoS, in terms of requirement analysis. The interleaving between the requirement activities and designing of software is not new and has been discussed in the literature of monolithic systems for some years, i.e., in traditional systems (NUSEIBEH, 2001; CLELAND-HUANG *et al.*, 2013; CHEN *et al.*, 2013; GALSTER *et al.*, 2014). This interaction was originally proposed by Nuseibeh (2001) and has been developed in different aspects, as addressed in Cleland-Huang *et al.* (2013), who discuss the necessity of evaluating the design alternatives in front of requirements and constraints identified or changed in requirements engineering during the entire lifecycle of development.

On the other hand, the same authors presented the *mountain range* concept, which has different candidate requirements to be evaluated in various candidate architectures. RE activities are executed in the whole software development process with high dynamism and interactivity with other activities, mainly architectural design or software testing. Such information agrees with the literature that discusses the need that these two areas (i.e., RE and architectural design) work in constant collaboration, because, in practice, they are not linked (NUSEIBEH, 2001; HALEY; NUSEIBEH, 2008; KEATING *et al.*, 2015; LARSSON *et al.*, 2016; NATARAJAN *et al.*, 2016). Due to their importance "Twin Peaks" models have also been discussed in SoS domain (HALEY; NUSEIBEH, 2008) and (LARSSON *et al.*, 2016). However, we claim that greater benefits can be achieved in SoS development if there is an interactive specification with SoS Mission. The development of an SoS is governed by the end-to-end mission, and needs or opportunities may arise leading to the emergence of requirements, capabilities or other missions, which can be met at design time or sometimes at runtime. Therefore, the mission must be presented in interaction with RE to maintain the focus on provided and on satisfying the necessities of internal stakeholders. These modifications directly affect the SoS architecture that needs to be adjusted or restructured to adapt to these changes, either at runtime or design time. Figure 21 shows the `Triplet Model`, which was refined based on industry practitioners it is the base of mission-driven development of SoS. It resulted from the systematic mapping of SoS development initiatives partially published in (LANA *et al.*, 2016). Below, the interviewee's statement 8, which points out the necessity to keep this interaction and some challenges faced.

> *Requirements have a very strong connection with architecture and design and there is a fine, very fine line that separates these two things. So, I have to be careful during requirements engineering not to pay attention and talk about design and talk about architecture, because that may mean making premature decisions that will limit the solution upfront. On the other hand, I cannot do the requirements work by completely ignoring architectural decisions mainly decisions that are a high risk on the part of the project. I need to consider this because it impacts the requirements work. For instance, you will develop a mobile system if the platform is Android or IOS is an architectural decision, only if I choose platform A or platform B this may have an impact on requirements because operating systems A provide things that B does not provide and what the operating system does I don't need to insert as a feature to be developed (INTERVIEWEE IT_8).*

Figure 21 is based on the philosophy of the *mountain range* (CLELAND-HUANG *et al.*, 2013), which presents candidate requirements and architectures. For the SoS domain, the first step after understanding the needs of stakeholders is to make the SoS mission available to specify its possible capabilities requirements, and constraints. Based on the information, it is possible to identify the first individual mission elements also known as constituent systems. For each mission objective, it is necessary to verify the capability and the performance measure that will meet the final process for the mission to be achieved with a satisfactory performance considering the best case. In addition, SoS mission has multiple individual mission elements to provide the same capability, and its choice will be related to the purpose of this system and its main ilities.

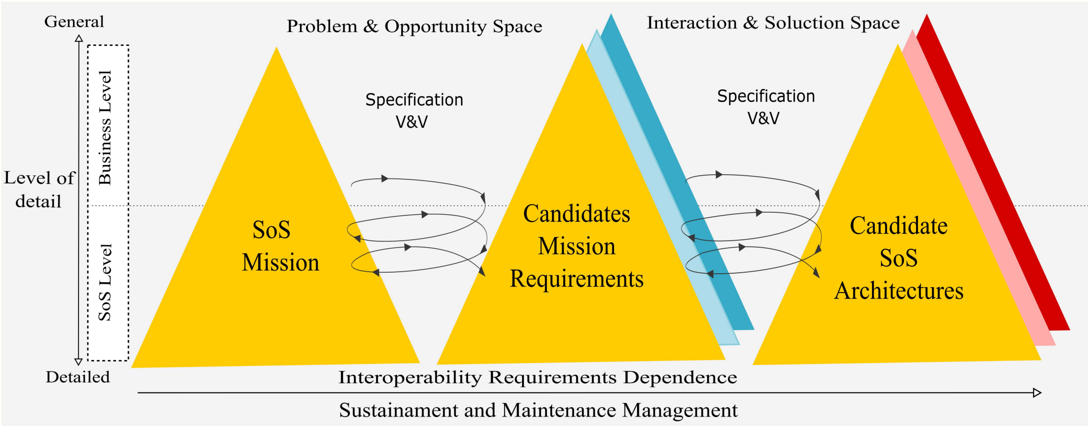


Figure 21 – Triplet Model: the interleaving between SoS mission, SoS requirements, and SoS architecture

Source: Elaborated by the author.

In the development of an SoS, we have to monitor at least two levels (business and SoS), as an organization of individual mission elements that collaborate between themselves to achieve a specific mission. To each global mission, both levels will be considered to perform this mission and many mission requirements are necessary for the planning of the system at design time and to its operation at runtime. Furthermore, for different external stakeholders, constraints will influence changes to the SoS, which may also affect the mission, requirements specification, architecture and the infrastructure itself. For instance, in the agricultural process of sugarcane cultivation, harvest could be performed by burning in the state of São Paulo until recently. Because of law No. 11.241/2002 all sugarcane field processes had to be adapted to manual or automated, and their entire systems that were still individual and non-interoperable had to be changed. The operation of an SoS is neither only directed, nor only acknowledged, but it has central control and to secure the accomplishment of the mission, it needs to maintain the contracts and agreements between the owners of the CSs and central control or between the organizations that are owner of SoS. When the contracts and concordance agreements are performed, the interoperability requirements have to be maintained to a minimal level of operation. In this case, the Maier's essential characteristics can all be observed in any SoS with central control. In Figure 21, this dependence is represented by an arrow increasing from left to right (i.e., SoS mission to candidate architecture), when the whole level of interoperability needs to be reached for a sustainable development of the SoS during the lifecycle.

Differently from the Twin Peaks models, which split into two spaces apparently well delimited but complementary, the Triplet Model has two shared spaces, where one directly influences the other being composed of: (i) problem and opportunity space; and (ii) interactions and solution space. Mission and RE are, by definition, related to *problem and opportunity space (i.e., WHAT)*, determining the dimensions of the problem, opportunities to design a mission that will be solved by the system (or in our case by the SoS). Similarly, architectural analysis is also concerned with problem space; it refines the capabilities and/or requirements provided by the constituents that were identified in the mission, RE or even in the architectural analysis activity itself and is important for the SoS architecture or for the mission fulfillment (i.e., Architectural Significant Requirements (ASR)) (CHEN *et al.*, 2013). Indeed, architectural concerns are described as capabilities and/or requirements of the SoS, and they can also insert mandatory design decisions (e.g., adoption of existing technology). Hence, initial architectural decisions may already be made during mission analysis and architectural analysis to help shape the candidate solutions for the architectural synthesis, which agrees with the to empirical analysis (BOER; VLIET, 2009; CERVANTS; KAZMAN, 2016). So, the constant interaction between mission, RE, and architectural analysis becomes indispensable. Our statement also requires the interaction between analysis and synthesis result in an *Interaction Space (i.e., why)*, with concerns not only to defining architectural problems, but also to negotiating, specifying, and validating capabilities and requirements. Architectural design (i.e., Architectural synthesis and evaluation) is concerned with *Solution Space (i.e., how)*. Architectural synthesis establishes the candidate solutions with architectural decisions, architectural descriptions, and so on to solve the problem or accomplish the mission. These solutions are evaluated based on the final document of ASR to provide a validated architecture (HOFMEISTER *et al.*, 2007). However, SoS is evolutionary and has emergent behavior and these characteristics can cause frequent change at RE level and at the architectural level, even leading to changes of SoS mission itself. Therefore, architectural requirements will be continually modified and interoperation can be affected by those requirements that have been inserted, removed, or modified. A similar procedure occurs in the individual mission elements level when they are withdrawn from inserted in SoS, or modified in their structure.

One general example of a technical requirement elicited and specified based on the Triplet model is shown in Table 10. We have a mission objective of `dealing with bad weather during harvest`, which was identified as a constraint to `use of the standard W3C` at the high level of SoS. The capability involved in this process is `performing communication and data exchange among constituents` and as the mission needs access to weather data, the metrics can be related to reliability of the weather data, the availability of the precipitation data, etc. Such choice is directed to stakeholder concerns related to decision making. The mission objective is specified in a qualitative way, which makes difficult the selection of precision measurement to the objective, while analysis is being performed at the high level of the SoS. In another perspective, these two capabilities support the identification of individual mission elements (i.e., constituents) that may be more refined in SoS requirements to be met by these elements.

4.2.6.2 Triplet Model and Architectures

All SoS are composed of structural and non-structural elements (e.g., individual systems, software, hardware, people, etc, which are recognized as mission elements), and such elements and their relationships are abstract organizations of the disposal of these elements in an operational

Table 10 – Template of specification of scenario used in the Triplet Model

Code: TI_{NFR1}	
Mission Objective:	Dealing with bad weather during harvest
Description:	The system must use W3C standards to establish communication between the central control and its constituents.
Quality Attributes:	Integration, Interoperability
Scenario	
Responsible Constituent:	Non-physical organization
Capability:	Communication and data exchange
Resource:	Network, hardware, and software
Stimulus:	Request between central control and their constituents
Source of Stimulus:	People and software
Environment:	Normal operation
Impacted Constituent:	All constituents
Response:	The system will use W3C standards to perform the communication and meet the request.
Rationale:	Standard used by organization
Cross-reference:	-

environment, known as mission architecture. Figure 22 displays the relationship between the Triplet model and mission architecture. To establish a mission architecture, the first decision is to identify the SoS mission and to perform the MisCon (i.e., mission concept (cf. Section 4.2.10) describing how the mission will work. In other words, how data is acquired or the mission is carried out to meet end-users needs (LARSON; WERTZ, 1991). It is required to look for mission elements and verify which alternatives for each of them would best meet the mission objectives, establishing a set of candidate architectures for evaluation in late-stage (ESTEFANIA, 2010). At runtime, changes (e.g., proactive, reactive, opportunistic, or adaptations) will occur leading to a new combination of mission elements that are also required to achieve the concern of the stakeholders and MisCon. Operational mission consists of integration and interoperation of these elements to support the ConMisOsp, besides of meeting policies, procedures, laws, regulations, and important data flows (WERTZ *et al.*, 2015).

Mission architecture is met by specifying of capabilities and requirements, which are well structured and organized in an architectural form, in a way that they are stable, usable, adaptable to change, and elegant; such form is called requirements architecture (HULGAN, 2012). Requirements architecture is a development based on the main candidate mission architecture, or only on the validated mission architecture. In the establishment of systems when the requirements architecture is sound, it supports the design and its description. However, if the requirements architecture is faulty or poor and it might be difficult to be known because some requirements were changed, the reuse of requirements and traceability can only be superficial or unused, the rationale can be unclear, and business objectives cannot be met. In a general way, a reliable requirements architecture is:

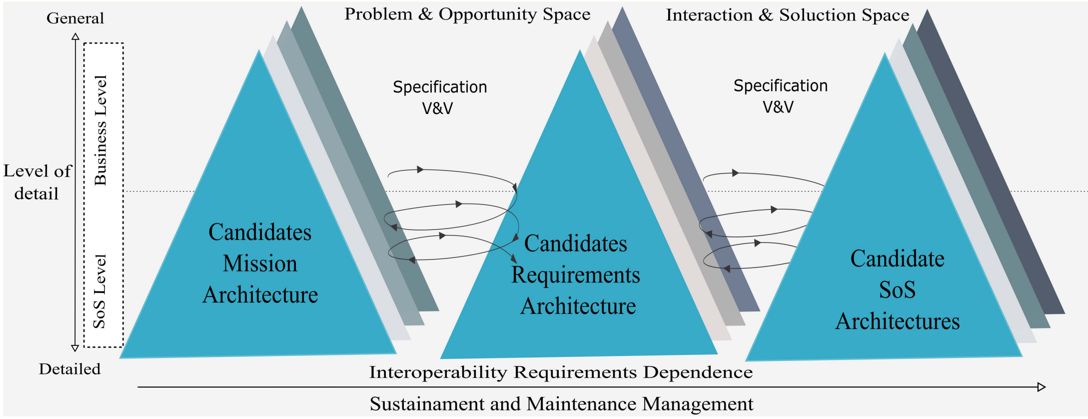


Figure 22 – Triplet model and the relation with architecture

Source: Elaborated by the author.

- **Maintainable:** degree of effectiveness and efficiency in which the capabilities and requirements organization can be modified by the intended maintainers and can be adapted to changes in mission, capabilities, and requirements.
- **Traceable:** ability to identify the origin and destination of each capability and requirement in the flow of a software development product, establishing the reason for its existence
- **Usable:** the requirements are architected in such a way that you could either produce output for each of them or navigate to the requirements using the tool and find the requirements objects that are relevant. The hierarchies and traces you create should be consistent, you should not create one hierarchy where the functional requirements are children of the models and another hierarchy for the same project where functional requirements are not children of the models but are traced to them. The absolute worst thing to do is to list all requirements objects in a flat list or to manage your requirements in Word or Excel, although they are still the mechanism most used by the industry.
- **Scalable:** ability to handle increased new capabilities and requirements by repeatedly applying a cost-effective strategy to extend the growth of a systems horizontally and vertically with minimal overhead
- **Elegant:** are there just enough hierarchies to facilitate use? Are you repeating hierarchies just to make traceability easier? Does your architecture contain duplicate models or requirements?
- **Generalizable:** degree to which an architecture should be reapetable in any project through of the reuse of various domains, which is the same if the approach to requirements architecture was different.

Hence, the elicitation to the architectural level and the activity of analysis of technical requirements can become more focused and less costly if requirements architecture is established and validated before initiating the development of architectural analysis. However, there are still interaction and requirements that can emerge in the synthesis phase during the architectural decision and influence requirements architecture as well as the analysis architecture. The dependence on interoperability has been growing from mission architecture to SoS architecture, because, at an architectural level, where the interoperation is executed, to any other level, only are identifying possibilities of a better way to perform this process that it is consolidated at runtime with the SoS architecture. On the other hand, the evaluation and maintenance processes, need to be monitored for mission, requirements, and SoS architecture, since any change at runtime can modify the three architectures or at least influence their organization or of the mission elements.

Figure 23 shows the relation between Mission Architecture (MA) and Systems Architecture (SA), besides of the connection to Enterprise Analysis (EA). To perform EA, it is necessary

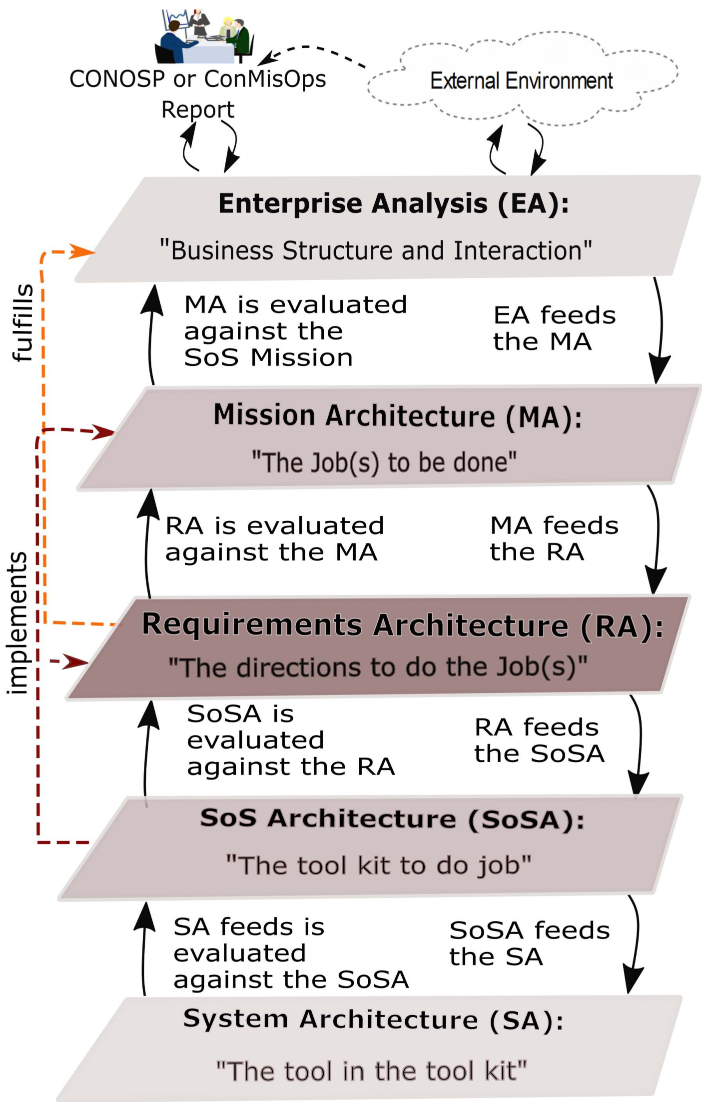


Figure 23 – Hierarchies of the architectural levels

Source: Adapted from Stokes (2012).

to receive the ConOps or ConMisOps (cf. Section 4.2.10) and there is constant interaction between EA and the external environment. EA is determined by the interaction and structure of the business, further specificities are related to services provided and their interaction and how these services are controlled. Such analysis and capabilities are the basis to feed the MA (i.e., they define the HOW, each time you go down an arrow). MA concerns "the job(s) or tasks to be done" with the ConMisOps already established, the process of requirements that defines the directions of activities supporting tasks such as, refinement of stakeholders, acquisition activity of constituents performed integrated with the level of EA as well as the identification and analysis of capability and requirements. For the fourth level, the SoSA is the tool kit to conduct the job, it is responsible for capturing the interaction of constituents through interoperability, identifying and organizing the message exchange flow across the interface, considering each category of SoS and their characteristics at design time and runtime. It also supports the gap analysis of capability, when conducting architectural synthesis and evaluation. At the end, SA can be considered the main tool inside the kit, because without it our SoS will not work. In a generic way, the SA works with each system to identify how the components interact, identifying the capabilities and gaps, and message exchange, preventing failure in its operation. It is noteworthy that, for a constituent to perform these activities, it needs to be a constituent known to the control center, which will not always happen with the virtual system and can happen with the collaborative systems depending on the context.

The left side denotes the WHY, encompassing the execution of the grouping of inferior elements, which are evaluated in comparison upward organization. For instance, SA is organized inside the structure of an SoSA, so each SA is also evaluated against the SoSA to identify the inconsistencies, gaps or improvements. In SoSA the evaluation verifies if requirements artifacts are accordingly addressed by SoS architecture and should also verify whether the mission artifacts have been correctly mapped to this architecture. In a similar form, RA is an assessment to demonstrate the compliance from RA to the mapping of MA, beyond carrying out the concerns and needs that exist in EA. While constituent testing and evaluation are still considered a challenge (LUNA *et al.*, 2013a), performing it from the early stages of the definition process (i.e., HOW) can reduce time and cost in both definition and performance evaluation (i.e., WHY) since procedures and artifacts are chained and often co-dependent.

4.2.7 Problems and consequences

Some interviewees report the main problems and their consequences when they are working with technical interoperability requirements to implantation of any infrastructure or to improve an existing one. Lack of clarity in the use of protocols causes more difficulty in changing the technology over time; hence, the cost of interoperation also increases. Another factor that contributes to higher cost is the closed standard and its diversity of multiple systems, because users are not able to adapt to a specific domain or system. As reported below:

> *I cannot do interoperability if all systems are not prepared to exchange data with each other. There are technologies that do not need to interoperate within the organization, but there is a specific group that is working for a specific mission that needs to communicate 24 hours a day, 7 days a week. When we have to change technology, the technology team often stays weeks to look for ways to mitigate the problems of adapting a standard that is not very clear or using a protocol that is not yet updated. (INTERVIEWEE IT_10).*

In the development of an SoS there is a large diversity of constituents that can be changed at runtime according to the capabilities to be attended or because of the evolution of constituents. During the interviews we observed that the more specific the system, the more problematic it is for interviewees to deal with the question of the interoperation in an SoS. Threats are challenging to identify, because it is necessary to verify various levels, such as mission and interoperability and, after identifying them, there is the need to mitigate them. Lack of cooperation skills between people and often between the organizations, as well as problems with network coverage, are beyond conceptual, and are key to leading to difficult interoperating everywhere and to wrong implications and decisions.

4.2.8 Aware communication

Aware communication is important for theorizing, since a communication problem can lead to errors in understanding and analyzing requirements. Stakeholders will change to the development process; thus, the development activities are to be interleaved to meet the requirements that can emerge or be modified throughout the lifecycle. For the interviewees, the phase of greater interaction between engineers and stakeholders is related to the understanding of the mission needs and to defining the capabilities. However, the communication between them can be complex and error-prone, because of cultural conflicts, concepts understanding, stakeholders will not always being clear about their needs and the engineer may not knowing the business context (FARINHA; SILVA, 2013).

Volatility, variability, adaptability, and difficulty in understanding are other factors that can provide challenges to the communication and analysis process of requirements mainly when the elicitation is performed without registering the rationale and the owner of requirements. By registering rationale support to the process of prioritization and negotiation, it is easier for stakeholders and interviewees, to argue when the information is consistent. In addition, the interviewees argue that simulation is important in many situations that are difficult to understand or that have a high cost. *We can understand what the engineer is talking about and how it is being done more easily when we see it working somehow, it is easier for us to give suggestions (INTERVIEWEE IT_17).*

Requirements specification must performed using an easily understood language by stakeholders since the early stage of the development and such question has been widely discussed in traditional systems. (CHRISTEL; KANG, 1992; RAJAGOPAL *et al.*, 2005; NISAR *et al.*, 2015). Moreover, validation of requirements was also mentioned because it is considered an activity responsible by checking realism, consistency, and completeness. Validation given support to reduce errors in specification, then when errors are discovered, specification must be modified to correct these problems to comply with mission requirements. For many practitioners, the success of the technical interoperability analysis process regards the multirole and multidiscipline interrelation that will support all processes of SoS development, as described in the following statements.

> *Everyone involved in the establishment of SoS should be concerned with interoperability (i.e., with the process of the interoperation of the individual systems) to one extent or another. However, when is related to "interoperability requirements analysis is perhaps best performed by systems analysts" (INTERVIEWEE ID03).*

> *Agree with the importance of domain analysts, since "interoperability requires knowledge of the domain, as they establish the context of WHAT information is exchanged, knowledge about the technical details known by architects, setting up HOW information is exchanged, and the interoperability engineers, WHO brings both words together for a conceptually aligned technically mature exchange" (INTERVIEWEE ID47).*

4.2.9 Technical Interoperability Requirements Challenges

To discuss the challenge of technical interoperability requirements, different issues were raised by respondents and classified into three groups: barriers, threats, and SoS properties (cf. Section 4.2.9.1). Barriers prevent interoperability from being achieved; in this case, they are not even established, and are related to five types, particularly to agro4.0. There is a barrier related to field infrastructure, and depending on the area and how this area is arranged, there are blind spots that make it impossible to reach network connection, in which case all activities must be done offline for that area until a new technology is applied:

1. **technical:** which includes lack of consistent data standards, multiple protocols, data quality, need for specific hardware or technology, how the data will be presented and exchanged, and so on.
2. **architectural:** refers to a problem with the specification of the interfaces of the multiple software, which have a different design to present the same function complicating the interoperation process and making the provision of services difficult. In addition, there are data and interface specifications of legacy systems to be integrated with current systems.
3. **economic:** concerns the cost of infrastructure development needed to implement an efficiency and robust SoS that meets the mission requirements and changing requirements

during SoS lifetime, and to some domains that need to exchange information with many organizations, there is a lack of incentive to keep data available or to share them.

4. **legal:** regards legal incentives within the country of origin of the SoS that sometimes do not exist to many domains and private organizations may have a problem using various solution of several countries because they use variable legislation.

5. **cultural:** relates to the distributed environment of SoS that composes the team and stakeholders with different languages and knowledge. The cultural barrier may increase the distance between stakeholders and mission and SoS team to requirements identification, negotiation, validation and so on. The conceptual issue may also be affected by communication and implementation.

Threats describe the circumstances or events that may impact interoperation, such as unauthorized access; in this case, it is established, but can be disrupted or negatively impacted. Threats are identified by apply qualities attributes of security in terms of integrity, confidentiality, accountability, authenticity, and assurance; beyond reliability, which addresses maturity, availability, fault tolerance, recoverability, robustness, correctness, and completeness. Interoperation of Systems makes them vulnerable to attacks, worms, viruses, malware, phishing and user errors that impair SoS integrity, confidentiality and availability, especially when it is being interoperated with the OT and IT. Reliability is a key aspect of an SoS to achieve of the mission outcomes, when technical interoperability presents some problems, such as failure of omission or assumption, failure of individual systems or failure of components that affect the delivery of SoS capacity, network failure, and communication, power failure, etc. Otherwise, the mission of an SoS can be compromised for an unlimited time.

According to the interviewees, other crosscutting aspects that impact the entire organization at multiple levels are transparency and data and information privacy, data backup and storage, the evolution of individual systems, distributed logistics, and morphoclimatic aspects, sustainability, and security. Many of them have been looking for ways to mitigate these threats, but not all of them are possible to be mitigated and end up becoming a known threat and even a future barrier if they are not eliminated or managed strategically. The statement illustrates this fact.

> *We know that there are threats, but you need to have systematic strategies (here the interviewee is talking about system strategy) to identify, they are accepted and mitigated. [...] such as lack of network, lack of power, external factors, data center recovery and problem with interfaces. In my opinion, systems cannot fail to be interoperated by threats, it is necessary to know how to identify and what can be mitigated or must be mitigated and what is not possible, will be a known threat and must be measured if it is possible to live with it or not (INTERVIEWEE IT_22).*

4.2.9.1 Systems-of-Systems Properties

We observed which SoS properties have affected the technical interoperability requirements. Ten properties were identified and organized using the (P) letter in ascending order: **(P1)** Collaboration between constituent systems, **(P2)** Evolutionary development of SoS, **(P3)** Distribution of the constituent systems, **(P4)** Operational and managerial independence of the constituent systems, **(P5)** Heterogeneity of the constituent systems, **(P6)** Emergent behavior of the SoS, **(P7)** Mission of the SoS, **(P8)** Mission of the constituent systems, **(P9)** Adaptability of the SoS, and **(P10)** Open-ended architecture of the SoS.

Among these properties, P1 has a large concordance agreement. In the process of interoperation, first, it is necessary to establish the integration that promotes the collaboration between constituents, being considered as a base to process the interoperation between two or more CSs. As a consequence of P4 and P5, the SoS also changes over time and adapting to several situations achieves an evolutionary development (P2). This process can be difficult because P3 and P5 generate more interaction between constituents and emergent behavior (P6). Technical interoperability requirements are impacted by P7 when it is necessary to identify the capabilities and the constituents and how they will be interoperated.

After interoperation, the SoS continues evolve, and the interoperability requirements and capabilities may change at runtime. Constituents can evolve or be changed (P8), inserted or excluded and then the interoperability requirements will be affected to adapt to a new configuration of constituents without failure of the SoS mission. Environmental changes may occur and, in this case, the own SoS mission can be updated or changed over time and new systems, technologies, standards, hardware, software, networks, and mediators can be necessary to improve the exchange of information and maintain efficient communication between them. Although P10 has the lowest concordance agreement, if the architecture is not open-ended, it is unprepared for adaptation (P9) and can bring great harm to the entire process of evolutionary development of SoS and adaptation to emerging behaviors unforeseen at runtime and those foreseen at design time, but they need to emerge at runtime to achieve the SoS Mission.

4.2.10 Singularity Mission strategy

Singularity concerns to the condition or quality of strategy of the mission, as a unique and distinct concern. Its change can promote a large effect on all chain of SoS development. Guided by the findings of the data analysis,identifying and understanding mission strategy driven by business context was verified as a necessity to perform situational analysis (KHAN *et al.*, 2013; DAHMANN, 2019) to evaluate the technical health and stakeholders needs to identify or adequate the SoS mission. By means of situational analysis, internal and external factors of a business are to be recognized, as well as their capabilities. Such an analysis may be conducted using several tools, as SWOT analysis[5], 5Cs model[6], and Porter's five forces. The report of an enterprise technical and strategical health makes it easier to develop the mission strategy plan and to design the SoS focused on mission and end-to-end requirements increasing the flexibility to achieve the mission outcomes.

Figure 24 shows the overview of Porter's five forces of (PORTER, 1998), which help understand the general organization of the SoS. By identifying the forces, we recognize the importance to monitor them inside the development process. The five forces recognized are important for interviewees to work in an environment that will have constant change where their continuous evolution of business process and the market competitiveness stimulate the growth and application of new technology and resources to improve the capabilities and services provided to the end-user. Each force is addressed as follows:

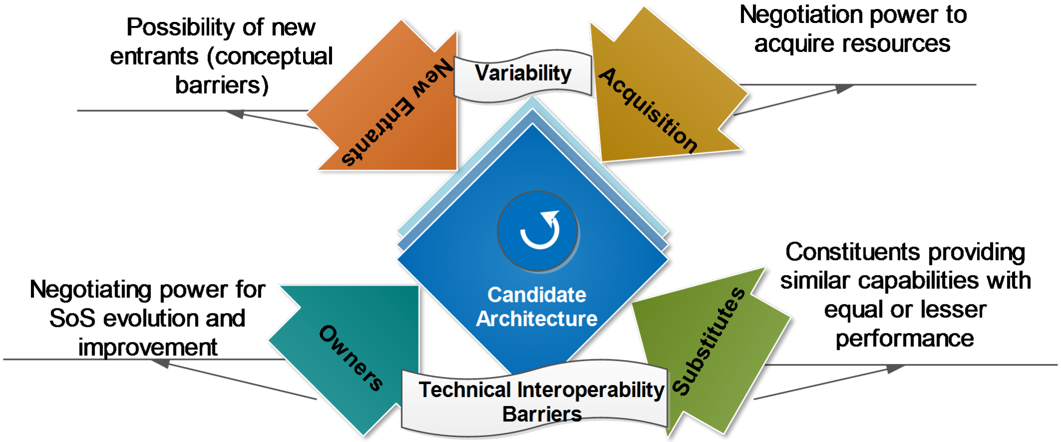


Figure 24 – Porter's Five Forces for technical interoperability awareness

Source: Elaborated by the author.

- **New entrants:** relates to the new companies that cooperate to improve or increase the provision of services or products for end-users, leading to an increase in the capacity of the entire SoS.
- **Owners:** refers to the bargaining power that the development team of the control center has to negotiate with the owners to implement new capacities, increasing or improving services or meeting a new demand from stakeholders.
- **Acquisition of constituents:** companies, due to their momentary and contractual power, can choose which constituent to acquire to meet a specific capacity affecting the cost of transactions and the quality of services provided which can lead to collaboration or competition.

5 It is support to identify Strengths, Weaknesses, Opportunities, Threats of organization and also of software.

6 It is support to identify and relationship the Context, Capability, Constituent, Collaborators, and Communication as discussed in Section 5.2.

- **Substitutes:** this strength can be characterized by organizations that produce or provide similar services to meet the same capacity. In this case, the quality of the service and the quality attributes with the trade-off analysis will influence the choice of which constituent will meet a specific capacity based on stakeholder concerns.

- **Variability:** relates to: (i) organizations with SoS that provide similar services; (ii) diversity of technologies and resources that directly impact interoperability, generating greater interoperability barriers between systems; (ii) the fulfillment of capacities predicted at design time, but which may emerge at runtime through a variation point according to the systems adaptation needs. The technical Interoperability Barriers were discussed in Section 4.2.9.

The interviewees considered the alignment between enterprise mission, system mission, and business process relevant to achieve outcomes. Thus, SoS mission strategy was associated with enterprise strategy, being derived from business goal and mission statement of the enterprise, since SoS mission is considered an operational capability, i.e., a task that needs to be executed through the collaboration of several systems that were not built to work with each other. Figure 25 displays the enterprise strategy plan and how each need that comprises the description of the ConOps[7] is extracted. Mission needs and requirements are extracted to the strategic level while stakeholders' needs and requirements are extracted to the tactical one. Based on these needs, the SoS needs, SoS capabilities, new constraints, and requirements at the operational level are identified.

Enterprise strategy has a purpose, which deals with the mission statement of which goals and Key Performance Indicator (KPI)[8] are selected to evaluate the management. For each goal, various objectives can be described to a primary or secondary degree whenever necessary. The objectives are achieved through specific strategies that can vary from three to five strategic actions for each enterprise internal or external goal that has tasks (i.e., tactics) to be performed. Based on enterprise mission refinement, we can elaborate on the Concept of SoS Mission Operations (ConMisOsp) plan, which is an integral part of ConOsp and describes: (i) how the mission will work at runtime; (ii) estimate the effectiveness of implantation of the Mission Concept (MisCon) plan; (iii) identification and process of acquisition of constituents; (iv) governance and management analysis; (v) technical and security analysis; (vi) Mission sustainability plan; and (vi) provides context to MisCon; as presented in Figure 26.

[7] For this thesis, ConOps is at the organization level, it addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems, as black boxes, to forward the organization's goals and objectives. The ConOps documents describe the organization's assumptions or intention in relation to an overall operation or series of operations of the business with using the systems to be developed, existing systems, and possible future systems. This document is frequently embodied in long-range strategic plans and annual operational plans. The ConOps document serves as basis for the future business and systems, for the project to understand its background, and for the

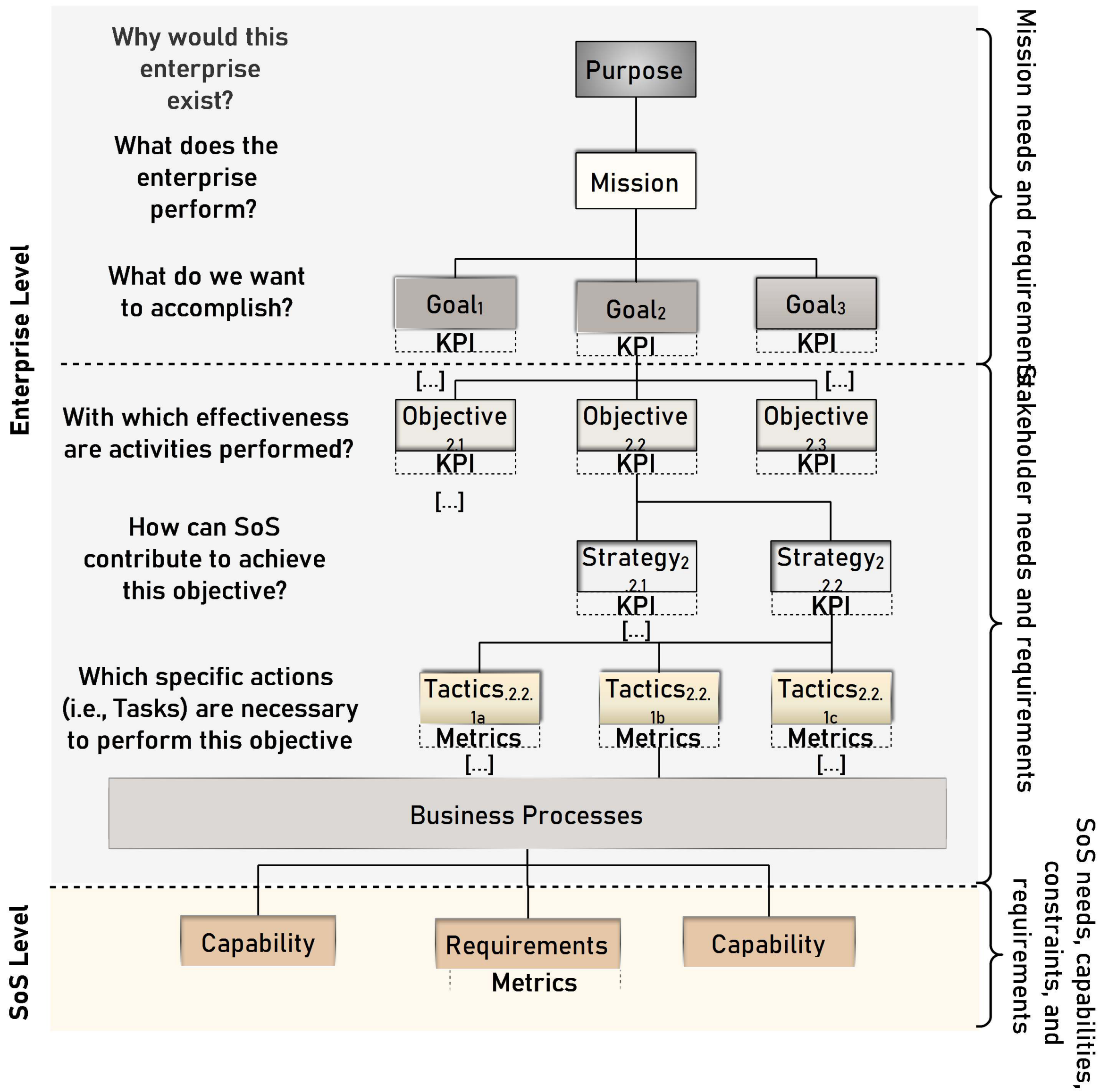


Figure 25 – Enterprise mission plan

Source: Elaborated by the author.

ConMisOsp emphasizes that each SoS has a general purpose that is translated to the mission and can have multiple mission goals. The mission goals do not have any indicator, because goals are normally high level, i.e., abstract and the indicator will be applied to the mission objective, which must be specific and measured and can have multiple levels, whenever required. In addition, the indicators of the mission objectives are often established at the MisCon level, because it is an end-user oriented document that describes SoS characteristics to be delivered from the user and operators viewpoint. Particularly, the MisCon document is produced to communicate an overall view of SoS mission as part of a system development or acquisition program, more directed to what the SoS will do and why (i.e., rationale). It enables the design

users implement the stakeholder's requirements elicitation (ISO/IEC/IEEE-29148, 2011).

8 The performance analysis is out to the scope of this thesis

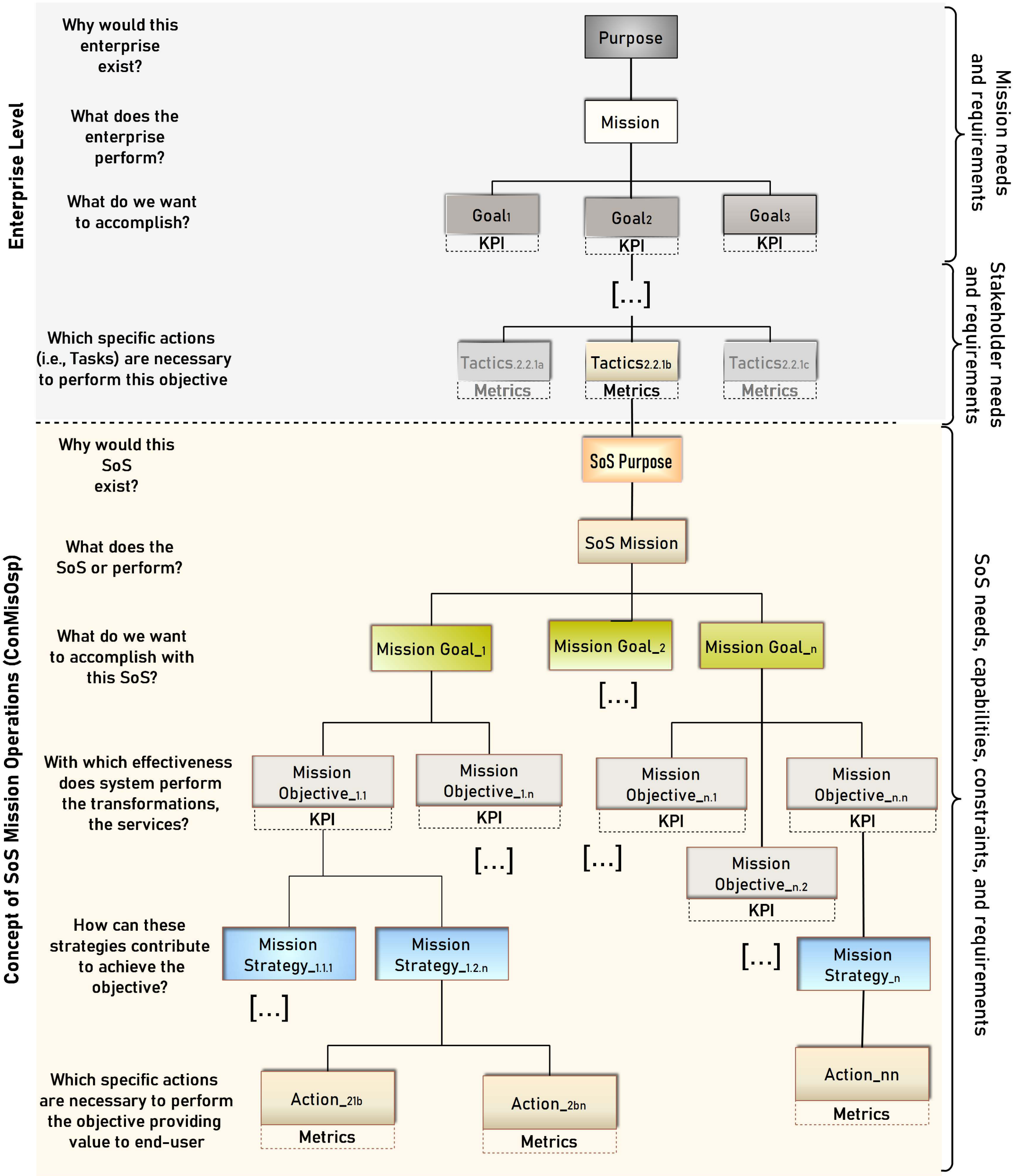


Figure 26 – A theoretical SoS mission concept plan

Source: Elaborated by the author.

and gives an overall picture of the interventions using one or a set of individual systems, to analyze the threats and the gaps. MisCon should be defined inside the ConMisOps document.

Despite each mission objective being quantified only in MisCon, its identification to ConMisOsp is also required because the strategies are identified based on mission objectives. Strategies that support the analysis of the needs of technical and non-technical resources and the acquisition planning or implementation for the action (i.e., activities or tasks) can be performed. ConMisOsp is important because it influences all the processes of internal decomposition of SoS, i.e., SoS mission, capabilities, requirements, constituents. Beyond that, it guides the identification of initial technical interoperability requirements and infrastructure requirements that are key to the design of the SoS. Each concept covered in ConMisOsp is defined below:

Definition 2. *Purpose: expresses the relevance of the SoS in its context of use. It presents the final aim of the SoS in its environment (FAISANDIER, 2015)*

Definition 3. *Mission: Expresses the set of services the SoS has to deliver, this means the main or global transformation the SoS performs, together with purpose, which clearly reveals the action to be taken and the rationale (FAISANDIER, 2015; LEGGETT, 2017)*

Definition 4. *Strategic Goal: the broad statements describing where the enterprise wants to be in the future in a qualitative way.*

Definition 5. *Mission Objectives: they are statement(s) responsible for quantifying mission goal(s), constraint(s) and services of the SoS as measurable data or properties related to space, time or effectiveness (FAISANDIER, 2015; NASA, 2018)*

Definition 6. *Strategy: defines actions to achieve the objectives, besides mobilizing technical and non-technical resources to execute these actions.*

Definition 7. *Operational concept and scenario (s): the mission is often an abstraction of a set of tasks (i.e., actions) performed by the SoS in its context of use and under defined operational conditions. Such specific groups of actions are associated with each other, as well as the matter, energy, and/or information they exchange with objects or the system of the context (FAISANDIER, 2015)*

From all the feedback received, the ConMisOsp and MisCon introduced detailed information that can reflect on the improvement of the identification of high-level requirements that can be more refined. However, this structure applied to ConMisOps is often hard to monitor and manage the results and has little flexibility to change the strategies and actions without influencing the mission objectives and including some threat to the achievement of the SoS mission. According to the interviewees, they usually have diverse variables interacting to compose a specific KPI and, in most cases, it is difficult to reach 100% of that KPI or control them to the horizontal and vertical levels, especially if they depend on human resources. Such needs directed the research to the evaluation of the Management by Objective (MBO) proposed by Drucker (2010), which define clear objectives for each collaborator and for each department according to business strategy, states that there must be collaboration of each department to achieve the enterprise mission. This was adapted in 1970, when Andy Grove, CEO of the Intel designed, based on the MBO methodology, the result-driven objective setting framework, called OKR (DOERR, 2018). Currently, the OKR has been used by renowned companies such as Google, Intel, Uber, Linkedin, Twitter, Spotify, Airbnb, Walmart, ING Bank, and TED Talk.

Figure 26 shows the OKR methodology, which combines qualitative and quantitative approaches that are broken down to identify the descendants and a bottom-up analysis must be carried out to obtain constant evolutionary feedback, promoting alignment up and down the organization (WODTKE, 2016). However, the core of OKR is the focus on results and not on the goal, which is characterized by multiple objectives (DOERR, 2018). Therefore, the key results are an important point of this methodology, because the objectives are measured through them, enabling the adjustment or changes in the actions to achieve such objectives with greater efficiency and effectiveness. Table 11 illustrates the mapping of OKR concepts for mission concepts shown in Figure 26, since many of them performed the same role. For instance, in OKR the "goal" and "objectives" concepts are used interchangeably, however, the goal is qualitative and objectives are quantitative, and for SoS mission, they are distinct. The "Key results" have the same role as "mission objectives" and "To do" has the same role as "action" or action plan in SoS mission concepts. Because of this, the original names were kept.

Table 11 – Mapping OKR concepts for SoS concepts

Source: Elaborated by the author.

Concepts	OKR	SoS Mission
Mission	Mission	SoS Mission
Objective	Objective	SoS Goal
Key Results	Key Results	Mission Objectives
To do	To do	Actions

Figure 27 shows ConMisOps restructured, based on OKR, with intent of providing a more flexible process of planning for the SoS mission. SoS-OKR can be adopted regardless of whether the organization itself uses OKR, the goal is to plan the lifespan of the SoS according to the changing environment and the needs of the organizations involved in the development of the SoS. For ConMisOps based on OKR, the operational activity was initiated, i.e., a task, which will generate the purpose and mission to the SoS. To be effective, the purpose and mission statement must be simple and clear (i.e., unambiguous and accurate), easy to understand and communicate (i.e., they are statements that reach different peoples, speak by them, motivate the use of the SoS, etc), inspire change (i.e., what transformation, in fact, it will promote), and meet the needs of stakeholders (i.e., the essence of the need and results being depicted in the mission). Then, it is necessary to evaluate the mission and purpose considering the aforementioned attributes and to perform the modification necessary to establish the mission goal. Each mission goal is broken down into mission objectives, which can have varied amounts in organization; in this case, most often three. Mission objectives are refined in various actions, which are to be performed as functions by constituents or by the SoS itself.

Having a good SoS mission is not enough, because the concrete mission objectives and the evaluation performed by the established actions will make the constituents achieve the SoS mission. In addition, the rows of actions are dotted and placed upwards to show that, by using the

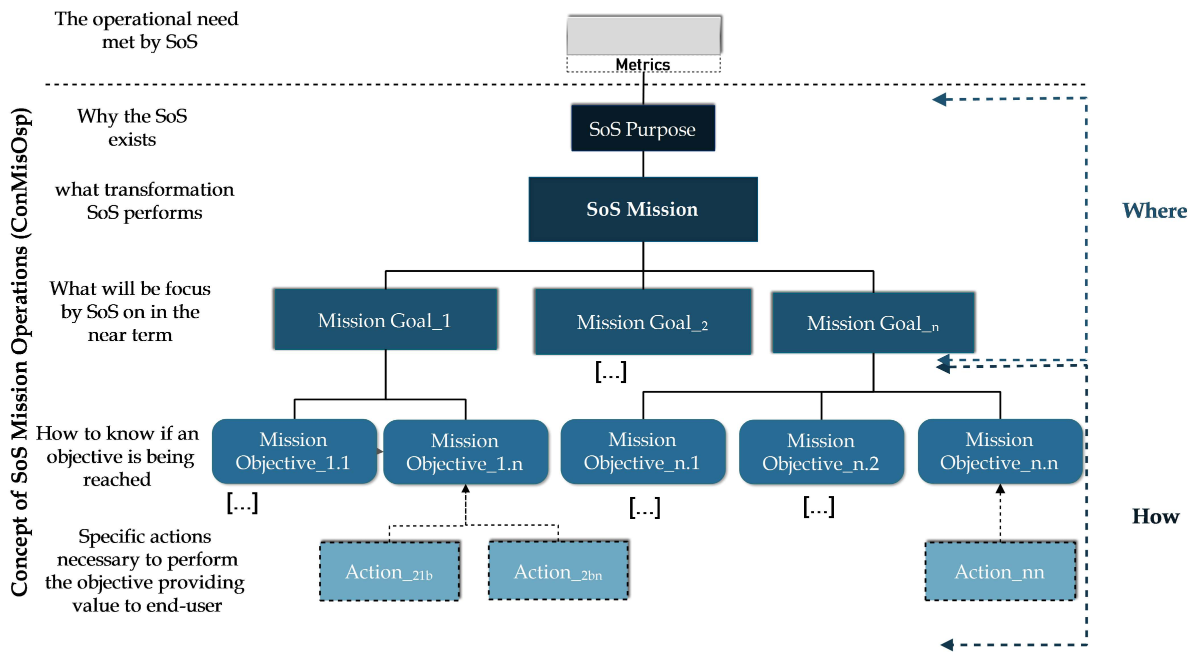


Figure 27 – ConMisOps based on OKR

Source: Elaborated by the author.

bottom-up approach, the actions can be modified as well as the objectives. One way to evaluate mission objective is by using the color grade, often used in OKR (DOERR, 2018). (i) from 0.0 to 0.3, in which the color is red and the mission objective is not believed to be reached by the constituents, unless that a new capability, strategy, requirements, architectural decision, or infrastructure is inserted, changed, or modified. (ii) from 0.4 to 0.6, in which the color is yellow and there is a great risk that mission objective will not be reached by constituents and that there are mission threats. In this case, the risk analysis is always identified and mitigated when it is possible. (iii) from 0.7 to 1.0, in which the color is green and the mission objectives will be successfully achieved by the constituents and, consequently, so will the global mission. However, due to the complexity of SoS the scale for this evaluation needs to be more researched, because several factors can influence the obtaining of a bottom-up analysis, beyond priorities analysis based on the color grades scale.

According to ConMisOsp the refinement performed for functional and non-functional requirements identifies the capabilities to SoS level and they can be allocated to each CSs. Mission-driven interoperability requirements are high-level capability that should be more decomposed during the design of the SoS according to an architectural decision. Therefore, technical interoperability requirements are a highly dependent on many architectural decisions, which are performed in the synthesis level Due to this, the interaction between analysis and synthesis, occurs according to the Triplet model. A statement from interviewee ID03 follows.

> *In my opinion, requirements, such as these (i.e., interoperability), are often put on forth as too specific and quantitative, probably because*

someone is trying to force a preconceived solution. I believe it's to try to express rather general needs and make them more specific as more is learned (INTERVIEWEE ID03).

4.3 Final Remarks

This chapter aimed to describe the findings of the categories identified to build the substantive theory to support the analysis of technical interoperability requirements driven by mission engineering. Each category was presented based on empirical analysis and, when necessary, a comparison between empirical analysis and relevant literature through literature-driven analysis was performed to demonstrated new elements and refinements of the empirical data. The needs of those interviewed were addressed maintaining the accuracy of the data collected through the grounded theory technique, and solutions for the challenges faced by them were developed and discussed considering the mission engineering and technical interoperability requirements field, which can be applied to deliberative SoS, specially to blended SoS. The ten categories are applied in Chapter 5 to develop the theoretical framework, which is composed of a core category that is described and related to the other categories.

CHAPTER 5

ATLANTA Theoretical Framework

5.1 Initial Remarks

THIS section discusses the construction of substantive theory represented by a theoretical framework, named ATLANTA based on its core category `Technical interoperability awareness` (cf. Table 39 in Appendix M), and on the constitutive elements from an empirical and literature-driven analysis (cf. Chapter 4). The research question defined in Chapter 1 is revisited and answered towards validating the research outcomes. Section 5.2 addresses the identification and analysis of the central category and the integrated best practices for technical interoperability awareness and 5Cs model. Section 5.3 focuses on the development of the theoretical framework, and Section 5.4 discusses the abstract model emerged from ATLANTA, whose three themes were determined and related to each other. Guidelines with transformation property of SoS mission and a theoretical proposition and refinement to the Triplet model were presented. Section 5.4.1 describes Theme 1, which addresses the nature of the technical interoperability requirements, Section 5.4.2 addresses the Theme 2 related to the strategic alignment of operational issues and constraints along with patterns for the specification of technical interoperability requirements driven by mission together with a technical interoperability frame and the technical interoperability quality model, called $QM2_{TI}$. Section 5.4.3 focuses on Theme 3 for the control and management of mission at runtime to achieve outcomes, and Section 5.5 summarizes the outcomes of the three themes that answered the research question. Section 5.5.1 reports the conceptual implications of the development of the substantive theory to the state-of-the-art and state-of-the-practice, and Section 5.6 provides the finals considerations.

5.2 Central Category: Characterization of the Technical Interoperability Awareness

An important issue in the construction of the theory is the identification of the central category to which the others are related. It is the backbone of the substantive theory, through which the researcher explains the key concept of the research, i.e., support to the analysis of technical interoperability requirements driven by mission engineering by means of empirical analysis (STRAUSS; CORBIN, 1990; CHARMAZ, 2005). During the data collection and empirical analysis, the interviewees showed serious concerns over environmental and technological changes and the way they are tackled in a national infrastructure scenario with the advent of industry 4.0, above all related to the national infrastructure scenario[1].

The analysis of Technical interoperability requirements driven by mission engineering have meaning and importance that exceed proven support for connecting two or more individual systems that often carry with them the strategic dynamics of the business itself in an intrinsic way. Such requirements provide a holistic view of infrastructure and technical constraints for benefiting business and meeting stakeholders' needs, while SoS is structured to strategically produce data and information on qualities. Data are transformed into long-term knowledge and consolidated as wisdom towards supporting the most correct decision-making within enterprises. Therefore, its importance for both SoS and organizations is no longer only technical, but strategic, as well. The core category for the substantive theory proposed in this thesis is `technical interoperability awareness`, which addresses the importance of interoperability elements and identifies and recognizes the analysis technical interoperability requirements as a knowledge required towards implementable actions that deliver solutions to the next generation of SoS.

Technical interoperability awareness *recognizes technical needs for interoperation by understanding time and space limits and business value for monitoring changes in the perspectives on future evolutionary procedures in the SoS development process.* The awareness and experience of the analysis of technical interoperability requirements is varied among the interviewees, and were affected by the lack of clarity of the concepts discussed in Section 4.2.1 and the several challenges. Towards addressing the process of creating holistic awareness in mission engineering context is required defocusing from a purely technological base to a means that sustains the collaboration among constituents and contributes to the achievement of the global mission. Stakeholders' awareness must be increased for benefiting its engagements and offering advantages to be used in other development phases. Regarding awareness, some actions and their responsibilities can be described and must be constantly observed to the success of the SoS mission, being at design time or at runtime, such as:

- The **central control** is aware of new scenarios to be necessary implemented and the

[1] <https://www.anatel.gov.br/dados/component/content/article/125-chamadas/353-plano-estrutural-de-redes-de-telecomunicacoes-pert>

capability of its structure.

- The **central control** is aware of interoperability threats and they can be mitigated towards not affecting the global mission.
- The **SoS team** is aware of threats from hardware, software, contracts, protocols and standards, and the way to reduce them towards not interrupting the global mission, or at least, the mission objectives.
- **SoS and mission teams** are aware of barriers in places and/or environments of implantation of SoS.
- The **mission team** is aware of possible mission threats that can emerge into the SoS during their lifecycle from their evolution and adaptation.
- The **central control** is aware of the main outcomes and global performance of the infrastructure available.
- The **central control team** is aware of the reliability, dependability, scalability, transparency, sustainability, and evolvability of whole technical infrastructure over time for supporting the scalability and evolvability from business and technology.
- **SoS and mission teams** are aware of architectural evolution and the way the technical interoperability requirements will behave towards meeting evolution.
- **SoS and mission teams** are aware they must verify at runtime the new technical interoperability requirements from business, environmental or technological changes, and
- **SoS and mission teams** are aware they must monitor the new requirements of the architectural adaptation or non-technical requirements.

Towards increased technical interoperability awareness in the SoS development process, we have built an integrated model of best practices upon eight related disciplines or concepts, e.g., enterprise analysis, mission engineering, funding resources, SoS architecture, business process analysis, and SoS RE Architecture (cf. Figure 28). Each property was inserted as an attribute of the central category and the bidirectional arrows of their relationship point to annotations that define the flow of information between them. The core of the model is blue and has five attributes, namely Id, responsible, description, SoSmission, and justification that should be used for the description of the above-mentioned best practices. ID is an obligation when related to requirement; otherwise, it is not used. Furthermore, the interviewees pointed out some codes considered relevant to the technical interoperability awareness category, for example, the indispensability for the 4.0 environment, which is constantly changing and offers several technologies for the achievement of the same capacity, or improvements in the technical situation through an adequate way for the identification of quality attributes and development of

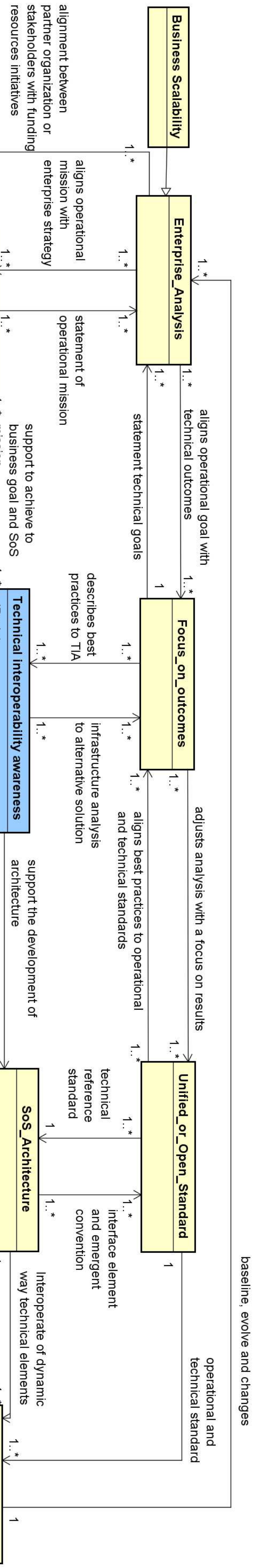


Figure 28 – Integrated best practices model for technical interoperability awareness

Source: Elaborated by the author.

analyses of possible trade-offs. In general, industrial sectors and academic environments lack infrastructures, regulations, and investments to work in an integrated way with industrial projects. Specifically, the analysis enterprise concepts involve codes for the understanding of business goals, decision making, knowledge of using data, and decision at several levels, essential for the success of SoS development and creation of such awareness.

All awareness generated by the best practices and relationships with other categories will provide support to the development of the SoS architecture; in counterpart; operational capabilities and technical constraints are evaluated and monitored at runtime by the central control team. Mission engineering enables technical interoperability requirements to collaborate with the business goals and SoS mission, as well as strategical analyses, since it aligns operational mission through enterprise analyses, thus identifying other operational missions. Such missions are focused on outcomes; therefore, technical goals must be established and aligned with operational goals. After this adjustment, new practices should be designed according to experience for aiding in monitoring and infrastructure analysis towards alternative solutions to both central category and best technical standards. Particularly, standards were discussed in the sustainable technical interoperability category in dilemma standardization versus openness (cf. Section 4.2.5); however, we understood that the needs of developing technical reference standards to design SoS architecture and work the interface elements and emergent convention to increase the facility of interoperation among standards. Such specificities help in the specification of the SoS RE architecture and description of the To-BE SoS. Business process analysis reveals needs, constraints, capabilities and low-level requirements, and investments justify cost/benefit analyses for alternative solutions. Investments are an initiative between partner organization and/or stakeholders with funding resources for the establishment of technical infrastructure, maintenance, and evolution over time.

5.2.1 5Cs Model for Technical Interoperability Awareness

5C's model has been commonly used in the market as an efficient situation analysis technique that supports business decision-making, since it involves a deeper analysis of both internal and external factors (WIND; MAHAJAN, 2001). Based on marketing guidelines, we have built a 5Cs model enlarging the technical interoperability awareness driven by mission engineering (see Figure 29). The model comprehends concepts of *context, collaborators, communication, capabilities*, and *constituents,* which are in constant interaction with technical interoperability needs, since they can be met or mediated towards achieving the SoS mission in a sustainable way.

Context analysis (cf. Table 12) identifies the needs of a domain that will support collaborators, which include organizations interested in the development of SoS and owners of the constituents and technical resources to be provided for the development of the system. Such collaborators deal with technical interoperability awareness towards mediating the whole SoS

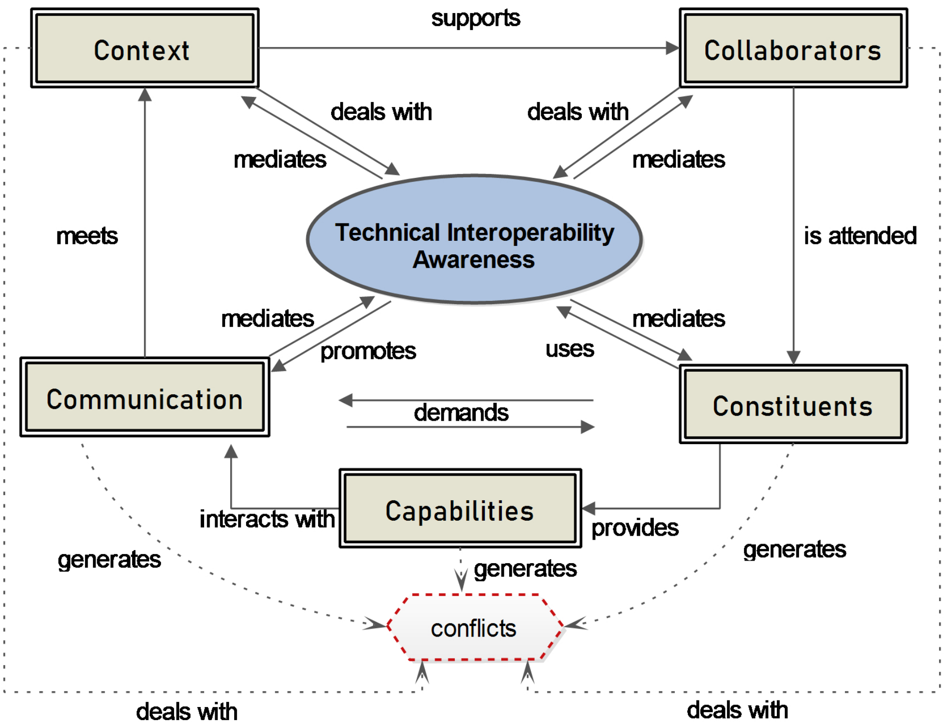


Figure 29 – 5Cs model for technical interoperability awareness

Source: Elaborated by the author.

lifecycle and promoting constant communications among constituents for complying with the context needs. Regarding constituent communication, the interviewees addressed that the unified communication has been proposed by Information and Communication Technology (ICT), which facilitates the integration between telecommunication and computer network by encompassing several devices and services. In their opinion, ICT can promote a possibility of having technical interoperability in a smoother way, although the process of requirements analysis driven by mission engineering remains a challenge.

Studies have addressed ICT standard how relevant standard is to improve interoperability challenges[2,3,4] (GASSER; PALFREY, 2007; MISURACA *et al.*, 2011). However, challenges, such as standardization of protocol and openness and specificity of domains are still discussed. Stakeholders demand communication mechanisms, many times associated with the capability they provide and the way they help the achievement of both global mission and SoS purpose. 4Cs (i.e., context, communication, capability, and constituents) have been considered, during the SoS design, generators of conflicts, thus requiring the application of negotiation, mediation, and prioritization approaches. Together with the central category, collaborators deal with the emergence of such conflicts in the situational analysis and during the development cycle.

2 <https://www.cencenelec.eu/standards/Sectorsold/ICT/Pages/default.aspx?Pages/default.aspx>
3 <https://ec.europa.eu/digital-single-market/en/news/more-interoperability-needed-boost-european-ict-\

Table 12 – Concepts of 5Cs Models for technical interoperability awareness

Source: Elaborated by the author.

Concepts	**Definition**
Context	An SoS's environment that determines the setting and circumstances of operational, technical, and regulatory development, and other influences upon SoS. It includes business goal, characteristics of the organization, stakeholders' concerns, and global SoS mission.
Communication	Related to technical communication that enables the interaction between resources and the interface of the constituents towards facilitating data exchange.
Collaborators	Involve the assessment of business needs and objectives, and indicate stakeholders' concerns to be improved. As to SoS owners, they deal with conflicts and unforeseen changes in the market.
Constituents	Individual mission elements or individual systems that make up the SoS.
Capabilities	Ability for achieving a desired effect under specified standards and conditions, through combinations of ways and means to perform a set of tasks.
Conflicts	Refers to any disagreement, different interest or need to be resolved through negotiation, prioritization, mediation and so on.

5.3 Establishing the ATLANTA Theoretical Framework

ATLANTA Theoretical Framework is a generic framework that incorporates ten categories addressed in Chapter 4 to provide an understanding for supporting the technical interoperability requirements analyses driven by mission engineering. Such an analysis process is surrounded by challenges and the adoption is still conditioned by the unfamiliar consequences that its non-conduction can bring, more than by the real benefits provided in the SoS development. The analysis processes conducted by the interviewees are performed in isolation by the owners of each system and not by the organization that manages all the systems through mission analysis. Such procedure can offer technical gaps and problems to the interoperation, directly impacting the mission concerns.

Substantive categories were related to each other through five sets of knowledge, which were considered inside of two groups, namely (i) non-crosscutting knowledge categories and (ii) crosscutting knowledge categories. Such sets of knowledge are acquiescing to the selected in Nakagawa *et al.* (2012). The former covers the **domain**, **application**, and **infrastructure**, whereas the latter is concerned with **communication**, and **design decision** as shown in Figure 30. Knowledge types were incorporated in ATLANTA for supporting the interviewees in the understanding of how technical interoperability requirements analyses driven by mission engineering can be considered in each group. Although all categories are connected to technical interoperability requirements and to support the analysis process, only maintaining the focus on infrastructure or communication needs does not provide satisfactory results to the interoperation

industrys-competitiveness>

[4] <https://www.itu.int/en/ITU-T/techwatch/Pages/learning-standards.aspx>

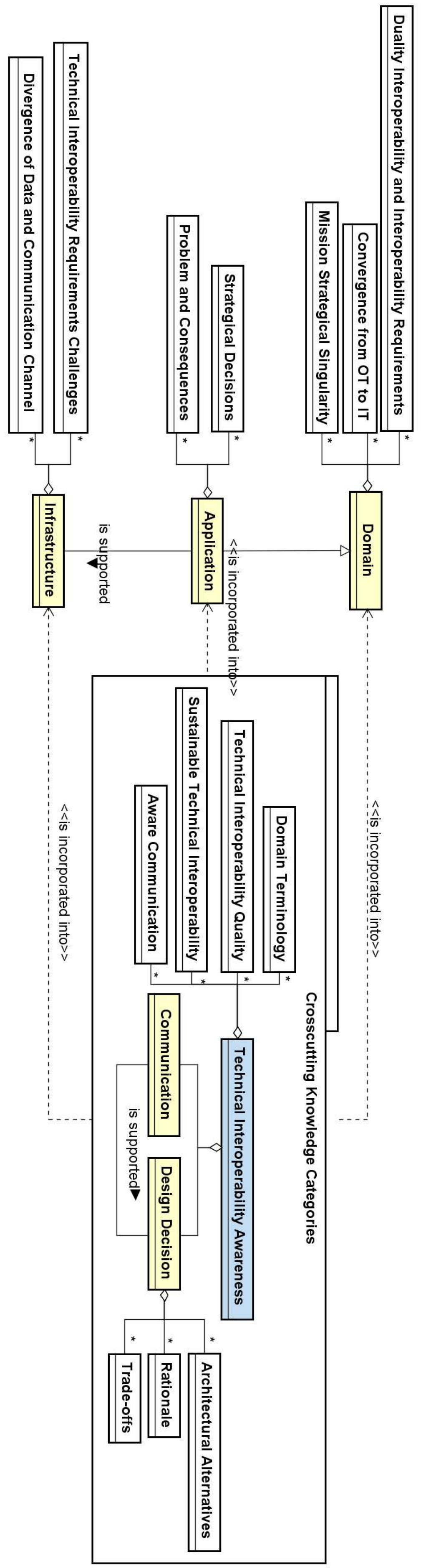


Figure 30 – ATLANTA Theoretical Framework

Source: Elaborated by the author.

process. The analysis must be continued, evolutionary, with frequent monitoring at runtime. Indeed, the activities or categories of technical interoperability analysis will be executed at design time when the SoS is thinking, and most frequently during its lifecycle to absolve new functionalities that arise and are not adapted by the architecture to express new behavior according to new events or scenarios (i.e., new systems, capability, or emergency of behavior unforeseen, which require to be treated at design time). The interviewees always raised the necessity of frequent alignment between business, market changes, and SoS, and have supported to improve the business objectives even with a critical environment of infrastructure and communication.

Domain elements are common characteristics, rules, concepts, and terminologies in a self-contained space of human action, which have well-defined functions and services. In our theory, the domain covers the concepts related to the technical interoperability making clear how interoperability requirements and interoperation have been considered by the industry, and how the concepts to the base of this theory were established (cf. Section 4.2.1). Such concepts were important to clarify the meaning to the interviewees of each terminology in the SoS domain. After explaining the concepts, we designed the systems model in which most of the interviewees are working with. The convergence of OT and IT was selected and a special SoS type was settled, named blended SoS, considered the closest to the reality of the interviewees. Furthermore, an approach to SoS mission based on mission engineering was required to strategically map mission and together with the interviewees an enterprise strategy plan called ConMisOp was established. A particular necessity of the industry for more simplicity and flexibility in the reason of often changes conduct to more refinements of the ConMisOps reaching a more agile process based on OKR. Such process facilitates the identification of SoS mission and also the actions, which can be changed with more flexibility without changes the mission objectives, improving the strategies for achieving the intermediate mission and consequently the global mission.

The second knowledge element is the application tier that concerns to interaction with end-user through the transport tier. In this moment new constraints are raised to technical interoperability requirements as discussed in Section 4.2.6, either at design time or at runtime. Although application tier is planned at design time, its applicability together with end-user is intensive at runtime, when the SoS provide its services. The interviewees noticed that such process of designing decision often needs to be taken at an earlier level of architectural synthesis, as it is performed in practice. Therefore, the interleaving between requirements, SoS mission, and SoS architecture was designed to the development of these system considering the specification of the SoS mission and technical interoperability requirements through scenarios. In practice, these activities are also performed in an integrated fashion and formally such integration and inserting specifics concepts of interoperability and SoS properties. Throughout the SoS development process, it is possible to emerge varied problems (cf. Section 4.2.7) that are solved by a design decision.

To infrastructure, besides the elements that enable the automation of the domain, it is composed of network elements that make possible the communication between several different automated domains. In the theoretical model, the technical infrastructure is addressed in all the categories, but two are further specific, which are technical interoperability requirement challenges (cf. Section 4.2.9) and divergence of the data and communication channel (cf. Section 4.2.3). The interviewees described the difficulty to identify the barriers, but they found easier to mitigate the threats. Five were the technical barriers found in empirical analysis (i.e., technical, architectural, economic, legal, and cultural) that now can be treated in the development of SoS. In special, one barrier was recognized to the agricultural domain that is linkage to physical infrastructure (i.e., the field itself) that have points that do not get internet signal, called blind spots. SoS properties were analyzed and discussed to verify which one influences technical interoperability requirements, which allowed that better attention can be applied to important properties as open-ended architecture perceived as less important than others by the interviewees in the analysis process. For the three knowledge elements presented, the situational analysis is recurrent as a mechanism for recognizing the current environment, its strengths, weaknesses, opportunities, and threats. Only from this perception, an accurate and more targeted proposal and analysis for a technical part can be performed, in an SoS proposal.

The last two elements are part of a set of knowledge that is crosswise spread across to the other domains during the SoS development, relating the domain, application, and infrastructure. To design the theory, the established categories are connected and related to each other considering the core category `technical interoperability awareness`. The same occurs with two elements of the set of knowledge `communication` and `design decision` that have a direct relation to other core categories, because the interviewees observed it as one critical relationship transverse when it is necessary to perform technical interoperability analysis to provide specifics requirements for supporting SoS interoperation. Current SoS domain has to meet upward and downward compatibility, because there are several substitutes inserted in the market to fulfill the same capability, enabling the coalition of many candidates. So, new concepts have to meet such requirements; otherwise, the analysis process could be difficult not to answer relevant industrial necessities.

Specifically, the core category `technical interoperability awareness` brings to the theoretical model integrated best practices and a variety of meanings in the analysis of technical interoperability requirements driven by mission engineering. Technical interoperability awareness is performed by aggregation of `domain terminology`, `technical interoperability quality`, `sustainable technical interoperability`, and `aware communication` that were recognized as main crosscutting categories to the awareness of the technical process. However, such process has involved cost and time that have to be analyzed and measured to be incorporated in ConMisOps planning. Domain analysis should be performed not only one time, but every time when there are modifications in the domain to which the SoS is being designed. Thus, such terminology (e.g., regulations, domain stakeholders, law, requirements, and constraints) would always be up to date to ConMisOps and MisCon. Technical quality, as mentioned can already be found in domain throughout all SoS until the end of its useful life. Due to this, understanding how such requirements impacts each knowledge element to technical analysis has high relevance, mainly because trade-offs are a great problem to architectural synthesis and evaluation. Technical interoperability quality model is more addressed in Section 5.4.2.3. Next, sustainability is a particular requirement addressed in this theory to support the longevity of the analysis process supported by ATLANTA Theoretical Framework through the enterprises and the industrial projects without failing to perform other important activities of the development process, such as verification, validation, testing & evaluation of requirements. These activities also contribute to the quality of the requirements and consequently, to the quality of the requirements architecture, and to the SoS architecture. At the end, aware communication is an element that must be carried out over time during the SoS development (among teams, teams and stakeholders, teams and owners, so on). Cultural issues and abstraction and technical levels, as well as issues related to the domain and the stakeholder clarity about what they want to be developed are to be verified and analyzed on time. Communication as the other elements is supported by design decision. In particular, SoS design and analysis of technical interoperability requirements have a decision that will affect architectural alternatives by reasons which are registered beyond compensations of trade-offs that should be considered during the application of the ATLANTA.

5.4 An Abstract Model of the ATLANTA Theoretical Framework

Regarding Charmaz (2005), there are various possibilities to present the findings of an emergent substantive theory, one of them is through the use of the abstraction of the theoretical framework by diagram using themes. The relationship of the major categories indicates the emergence of three theoretical themes and their relationships to each other. The three themes identified were: (i) **Nature of the technical interoperability requirements** is key for contribute to the process of (ii) **Strategic alignment of operational issues**, which impact the (iii) **Control and management of mission at runtime to achieve outcomes**. Each theme is related to the research problem and in turn with the research question raised from the problem, and presented in gray box. Such an abstraction indicates that substantive theory about analysis of technical interoperability requirements driven by mission engineering is able to contribute positively to the analysis based on key issues raised and discussed in the empirical analysis and grounded in the literature-driven analysis.

Research Question: *How can we specify a theory to support the analysis of technical interoperability requirements with the use of SoS mission engineering?*

The abstract model of Figure 30 in Section 5.3 portrays the relationship between industry practitioners and academics developing industrial project perspectives of SoS and similar systems.

Negative perceptions of technical interoperability requirements analyses had developed barriers to the analysis process that took interviewees not to perform analysis of interoperability. These created strategical possibilities to perform other similar activities that are also considered safe in their experience and cheaper. Many of them have positive perceptions about the analysis and encourage its use in the projects, even applying approaches not oriented to interoperability requirements analysis. In this context, there are still interviewees that do not know the concept of mission engineering, but apply its essence within the development process. Cao *et al.* (2011) address the challenge of "no counterpart in real-world" faced by requirement analysts when they work with large-scale software-intensive systems, as SoS. Such a challenge brings negative perceptions to when an SoS is developed, because they do not have any reference point as an example. For instance, in traditional systems when we attempt to develop a library information system to support its function, such as the purchase of books, loans, returns, etc. there is already a model in the real-world to fulfill the above activities. Then, based on this model it is feasible to elicit and analyze the technical requirements. However, many SoS are novel and there are no counterparts in the real world, and even if they existed, they would rarely be similar to the low level. Thus, the analysis has become more difficult to perform because multiples levels must be considered at design time and the analysis process intensifies at runtime when the whole operation comes into execution.

To perform the abstraction process, after establishing the major categories, the first step was to analyze the relationship of each minor category as illustrated in Table 13. Briefly, this table shows the three core themes and each sub-research question matching. Each knowledge level was organized by theme and the embedded categories related to one another across the themes. The yellow categories are related to crosscutting knowledge level and is presented to all themes, except the architectural alternatives, which is presented only to Theme 2 and Theme 3. Excerpts of analytic memos developed during empirical analysis were used to aid the illustration of the themes and the inter-relation between them. For each theme, one relational proposition was proposed, since it helps the substantive theory support the concepts by answering the question on Kipling's Questions, attributing to the studied phenomenon and that will be part of the construction of the theoretical blueprint (CORBIN; STRAUSS, 2008). Therefore, based on the analysis, three propositions were identified which help to ground the emergent theory, as presented hereafter. In addition, before showing, the theoretical model abstracted from Figure 30, each theme will be addressed in sections below grounded by literature.

Proposition 1. *The knowledge of the nature of technical interoperability requirements is a consequence of an aware communication between teams and stakeholders*

Proposition 2. *The nature of technical interoperability requirements supports the operational outcomes.*

Proposition 3. *The greater the strategic alignment in the operational context, the better the control and management of SoS mission at runtime.*

Theoretical code (Theme)	Research Question	Knowledge Level	Related Caregories	Excerpt from an original analytic memo	Inter-Theme Retaltionship	Propositions
Nature of the technical interoperability requirements **Theme 1**	Does a change in the mission engineering affect technical interoperability requirement analysis and interoperation of central control SoS?	Domain, Application, and Crosscutting	Mission Strategy Singularity Convergency of OT and IT Strategical Decision Technical Interoperability Quality Domain Terminology Aware Communication Technical Interoperability Awareness Sustainable Technical Interoperability Rationale Trade-offs	Mission specification help to understanding and attenting the stakeholders needs to low level through strategical and technological decision to identify the technical interoperability requirements that SoS with central control requires to improve the process of sustainability.	Relationship between Theme 1 and Enterprises	P1: The knowledge on the nature of technical interoperability requirements is a consequence of an aware communication between teams and stakeholders
Strategic alignment of operational issues **Theme 2**	Does a change in technical interoperability requirements among constituents impact the global mission?	Application, Infrastructure, and Crosscutting	Strategical Decision Problem and Consequences Technical Interoperability Requirements Challenge Divergence of Data and Communication Channel Technical Interoperability Quality Aware Communication Domain Terminology Technical Interoperability Awareness Sustainable Technical Interoperability Architectural Alternatives Rationale Trade-offs	The divergence between data and communication channel promotes problem to interoperation and to technical interoperability requirements analysis. Many times, they provide constrants to design of SoS to maintain the alignment strategical with operational goal.	Relationship between Theme 1 and Theme2	P2: the nature of technical interoperability requirements supports the operational outcomes
Control and management of mission at runtime to achieve outcomes. **Theme 3**	Does the behavior not incorporated by the architectural adaptation due to internal or external modification to impact the analysis of technical interoperability requirements?	Domain, Application, Infrastructure, and Crosscutting	Strategical Decision Divergence of Data and Communication Channel Technical Interoperability Quality Aware Communication Domain Terminology Technical Interoperability Awareness Sustainable Technical Interoperability Architectural Alternatives Rationale Trade-offs	The development of an SoS with central control provides various challenges when thinking about technical interoperability requirements, because each one has a specific way to work with requirements. To mission level at run time the control and management is related to behaviors that constituents have when interoperate and how these bahavior are accommodated into of the architecture to support the mission outcomes. Technical infrastructure can be exchange in this time and needs can emerge and must be directed at design time to analysis.	Relationship between Theme 2 and Theme3	P3: The greater the strategic alignment in the operational context, the better the control and management of SoS mission at runtime

Table 13 – Major categories and inter-theme relationships

Source: Elaborated by the author.

5.4.1 Theme 1: Nature of the technical interoperability requirements

The minor categories related to this theme were related in Table 13 and focused on them. We developed a way to help the interviewees to understand and to use the emergent theory, which is applied for SoS with central control and focused to blended category for being the closest type of interviewees. By analyzing the interaction of *Triplet Model* with the interviewees, the feedback received show that a further detailed meta-model presenting the possible relation between such concepts could support them in the process of interleaving between SoS mission, RE, and SoS architecting. Therefore, a meta-model was designed showing the high-level relation between the levels, mission, SoS and, CSs. The following statement illustrates the evaluation process.

> *I know twin peaks, the idea is interesting, but it is difficult to use because I have no more information about how the relationship between these elements is. Having a model, or a meta-model, could help improve that understanding and make it easier for adopting within the development process. (INTERVIEWEE IT_10)*

Figure 31 shows the metamodel of mission engineering, the green color classes are related to the concepts of the mission level; the yellow color classes to SoS level, while the blue color class regards to constituents' level. This meta-model enables specific and design global missions from enterprise analysis to the constituent element. Enterprise analysis can be supported by SWOT, 5Cs model, and Porter's five forces, whereas planning of mission must be performed by using the ConMisOsp, if executed to enterprise level, and its main purpose is describing the entire business operation and how the SoS contribute with it. Whether the purpose was to report about the activities performed in the operational environment and how technical interoperability requirements can help improve the interoperation process from the outside-in viewpoint, the planning must be performed for MisCon that will be directed to solution space.

According to the metamodel, **Systems-of-SystemElement** are compound by a set of **SystemsConstituentsElement** that collaborate among them to satisfy an SoS Mission. Each **Systems-of-SystemElement** is supported by a **Target Problem** that justifies the **SoS purpose** and its global mission. Each CS has its own capabilities met by **CapabilityElement** which reflect the services they provide to SoS. While the SoS is a non-physical CSs organization with a certain architecture with additional capabilities, which emerge from the composition of the individual capabilities of each CSs. An SoS mission is extracted from high level needs after performing **EnterpriseAnalyis_Element** and establishing the translation from **MissionElement** to **MissionObjectiveElement** together with a set of stakeholders of SoS. This process can be conducted by using the guidelines of ConMisOps based on OKR (cf. Figure 27). Considering this SoS mission, mission objectives are established and then partial and individual missions can be translated to meet those objectives by being further refined to smaller parts. Each SoS capability (cf. **CapabilityElement**) when decomposed to smaller parts is met by CSs that possess an individual mission (cf. Figure 16). Such SoS capabilities can be mapped of **BusinessProcessModel**

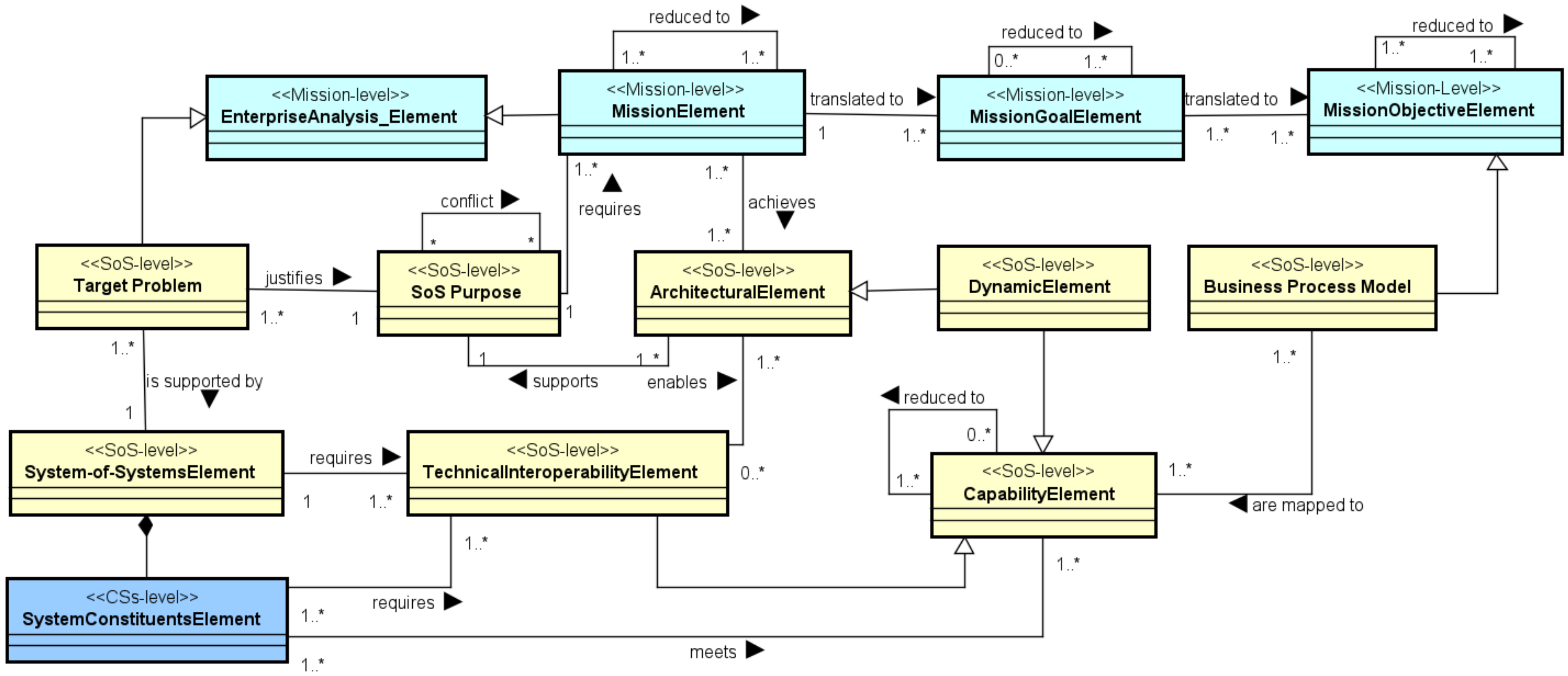


Figure 31 – Mission engineering metamodel

Source: Elaborated by the author.

and will be always changed whenever updates are necessary to operational process or to the business goal. This metamodel is only adopted as a way of mapping capabilities and requirements that are not adequate since it registers only one viewpoint of the need that is being modeled. Then, in industrial practical, it is suggested the application of the triangulation techniques to specify capabilities of gathering multiples viewpoints and register more than one mode of analyzing the capacity, being technical interoperability capability that will be translated to requirements or any other. SoS have dynamic architecture (cf. **ArchitecturalElement**) and at design time variability points (cf. **DynamicElement**) should be implemented to emergent behavior already foreseen. CS can be modified by different reasons, among them by solicitation of stakeholders and this modification can lead to changes in their capability. As a consequence, changes at runtime can occur and emergent behavior unforeseen can appear generating needs that previously did not exist. To meet such needs **technicalInteroperabilityElements** are essential because new capability can emerge and not being served by current constituents. In this case, one new constituent may be necessary for interoperation and new technical interoperability requirements by gathering and analyzing. One second possibility is the emergence of only SoS requirements and it will be necessary to negotiate with the owners of the constituents for its implementation or to sign a new contract or updated the current one to meet this new request. In addition, this need can become a new mission, thus new capabilities and constituents can be necessary to provide the services and mediators to be modified to these CSs. To go a step further, we decided to refine the Triplet model from the strategic to the technological level, with restrictions, principles, and rules that can help interviewees and the SoS , requirements, architecture, and mission engineering communities for better performing the mission transformation and derivation of technical interoperability requirements of this mission.

5.4.1.1 Transformations of Technical Interoperability Requirements driven by Mission Engineering

One of the main challenges faced for the SoS mission specification and transformation is to maintain the information consistency to the entire development cycle with updated traceability and also to understand how the changes will affect analysis and design, mainly the changes at runtime. For the interviewees, the traceability in this process is important to maintain the functional suitability, beyond aid to maintain traceability for data modification to support the security. Whenever we talk about the nature of the technical interoperability requirements, we are concerned with the stakeholder's needs and how it is possible to better understand these multiples needs registered the rationale of the evolution of these needs over time. $Triplet_T$ was projected in a systematic and iterative way and under well-structured principles using a methodology such as GORE, which facilitates the process of communication with multiples stakeholders, the refinement of high-level goals to lower-level goals, and operationalization for operationalizing goals as function, supporting in the identification of clearer information and conflict solution (YU, 1995; LAMSWEERDE, 2001). Thus, the interviewees can initiate the process of improvement of the aware communication to their various distributed stakeholders and rationale including the diversity of constituents that the SoS presents.

Figure 32 shows the $Triplet_T$, which is composed of three level at design time, i.e., organizational (a.k.a. Enterprise), SoS, and constituent and one level of mission management at runtime that will be discussed in detail in Section 5.4.3. The enterprise analysis is concerned with the problem and opportunity space to help in the situational analysis and it was addressed in initial Section 5.4.1. However, as the enterprises are further focused in outsidein model and

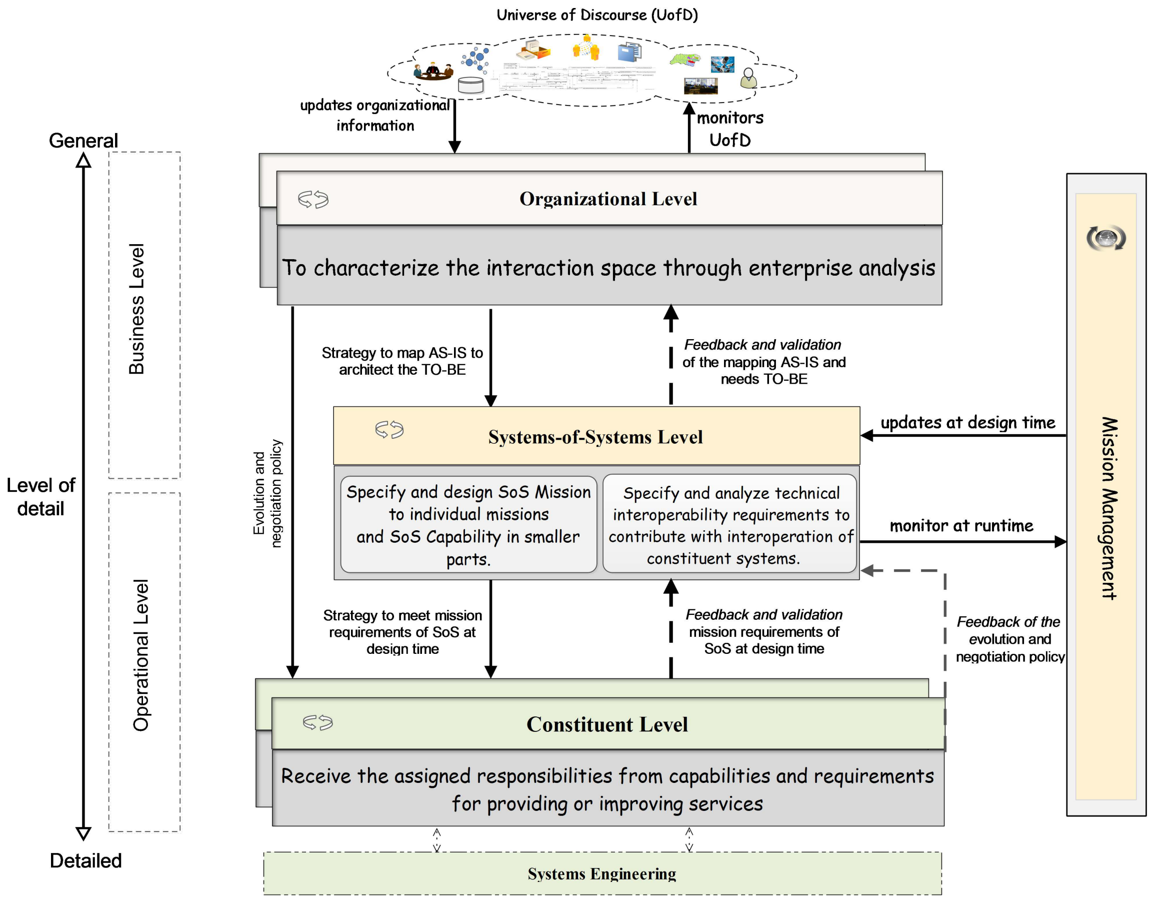


Figure 32 – Refinement of Triplet model to requirements transformation

Source: Elaborated by the author.

experience-based value, a constant environmental monitoring enables mission and SoS team to always maintain the modification updated, mainly those that directly trigger modifications to the SoS, to technical interoperability requirements or that can influence the accomplishment of their mission. Thus, it is possible to map the AS-IS systems to design TO-BE SoS in an iterative fashion. We can identify four transformation properties that are related to stakeholders and that later can be applied in the specification of SoS mission to smaller parts. Transformation properties are a set of rules, constraints, patterns, and definitions that were initially extracted of six works from the literature-driven analysis (JAYAPUTERA *et al.*, 2003; AYALA, 2015; VIERHAUSER *et al.*, 2016; SILVA *et al.*, 2015b; KUMAR; MERZOUKI R.AND BOUAMAMA, 2017; ZHU *et* al., 2017) and through these researchers we performed new analysis driven by data designed the adaptation and the development of the novel properties. In order to define these properties, we introduced lightweight formalism that will be used to all the levels.

Table 14 – Symbols List of the Transformation Properties

Symbols	Description	Symbols	Description
StK	Stakeholders	CSs	Constituents Systems
HLN	High-Level Needs	CS	Constituents System
M	Mission	CSC	Constituent System Capability
MO	Mission Objective	SoS	System-of-Systems
IM	Individual Mission	EB	Emergent Behavior
PM	Parent Mission	EDP	Evolutionary Development Property
DM	Descendant Mission	RC	Resources
C	Capability	NM	New Mission
CO	Capability Objective	NC	New Capability
PC	Parent Capability	NR	New Requirements
DC	Descendant Capability	ND	New Needs
R	Requirements	T	Tasks
PR	Parent Requirements	I	Intention
DR	Descendants Requirements	S	Service

Rule 1. *All stakeholders have Intention (I) that needs to be translated in HLN, ($1 \leq HLN \leq n$), which evolves into a more formal statement before reaching a capability (C) or requirement (R).*

Rule 2. *For all main StK there is a degree of influence, which needs to be identified and evaluated in relation to decision power under SoS.*

Rule 3. *For all main StK it is necessary to assign a role and responsibility, when a main Stk changes (i.e., remove) his role and responsibilities should be assigned to another one. If a new StK is added the role or roles could be reviewed right away. Otherwise, it could be reviewed from time to time to adjust.*

Rule 4. *All business process must have a logical sequence, be optimized and its value identified.*

These properties are related to the stakeholder's engagement that is needed to participate in the design of an SoS throughout the whole lifecycle so that the mission outcomes are more consistent.

On the other hand, the problem and opportunity space and interaction space are divided into SoS level. The former is regarding to specify and design SoS mission, whereas the latter to specify and analyze technical interoperability requirements that will further be explained in Section 5.4.2. To specify and design the SoS mission, aside from the principle, ConMisOps, and MisCon foregoing, one well-structured approach was applied, called middle out supports (USCHOLD; GRUNINGE, 1996). Middle out support the interviewees to identify needs at runtime, whereas the latter helps to model the interconnections between constituents to design the SoS. Using middle out monitoring and updates can reduce inaccuracies by leading to less re-work and less overall effort of the decomposition and management at runtime (USCHOLD; GRUNINGE, 1996). The first understanding as specifying the mission is to comprehend the generic structure of the SoS illustrated in Section 4.2.1, since they afford the refinement of a global mission to individual mission. This structure is initiated with a global mission, but to have an SoS mission, as discussed, it is needed to identify the *HLN* of the strategical planning through the analysis process performed by strategical level as addressed in Section 4.2.10. Only after, it is able to raise with the stakeholders the mission to be served by new SoS. The definition of *HLN* in SoS mission is described by property:

Rule 5. *The HLN are translated to only one M, which has at least two MO, where MO is* $2 \leq MO \leq n$.

Each main step of the generic structure was translated into one rule to make further flexible the refinement process. From `Rule 5` to `Rule 7` the refinement is top-down, whereas `Rule 8` performs a bottom-up analysis to collaborate between CSs. `Rule 9` and `Rule 10` are responsible to support the consistency and completeness of the model. After identifying the SoS mission, it is necessary to find the partial mission that distinguishes the intermediary mission of the organization's no-physical categories. Such a mission can be specified by following the next property:

Rule 6. *All PM have at least two DM (i.e., sub-missions)*

These missions originate the leaf mission that is specific of constituents (i.e., physical systems). When finding a leaf mission it is unable to specify it in new missions. Each leaf mission is described through the following property:

Rule 7. *An IM at the leaf level cannot be refined further into another DM.*

For this level, one CS receives one or more than one responsibility that needs to manage to collaborate with other CSs to the accomplishment of SoS mission. This assignment is described considering the properties below:

Rule 8. *A DM > 0 (* $leaf = 0$*) is formed by collaboration from some CSs at lower levels than its own level.*

Rule 9. *Every DM that is IM (i.e. leaf level) should be allocated to at least one CS.*

Rule 10. *A CS can be responsible for more than one DM that are IM.*

At the end, we identify a condition for the quality of the refinement, that is specified as the *MDecompose*. Meanwhile, the design and specification into a hierarchical model is fulfilled by decomposition. The *MDecompose* can be generalized and will be also applied by capability, requirements, and so on, which was defined as a general way as follows.

Constraint 1. ***GDecompose*** *= if $\vartheta_{i,j}$ can be decomposed into $\vartheta_{i,j}$, $\vartheta_{1,2},\ldots,\vartheta_{m,n}$ the decomposition between $\vartheta_{i,j}$ and $\vartheta_{1,1},\vartheta_{1,2},\ldots,\vartheta_{m,n}$ can be described as Decompose($\vartheta_{i,j}$ {$\vartheta_{1,1},\vartheta_{1,2},\ldots,\vartheta_{m,n}$}). $\vartheta_{i,j}$ is the parent of $\vartheta_{m,n}$, that is defined as Parent($\vartheta_{i,j}$, $\vartheta_{m,n}$). $\vartheta_{m,n}$ is the descendant of $\vartheta_{i,j}$, which can be described as Descendant($\vartheta_{m,n}$, $\vartheta_{i,j}$), where, ϑ can be a mission, capability or requirement; $i = \{1,2,3,\ldots m\}$; and $j = \{1,2,3,\ldots n\}$.*

During the process of specification and design, information is exchanged between levels. Thus, the application of iteration and appropriate feedback loops help to ensure communication that accounts for ongoing learning and decisions in the design process (WALDEN, 2007). This facilitates the incorporation of learning from further analysis and process application as the technical solution evolves (SEBOK, 2014). Beyond facilitating the identification of conflicts to various missions or interests of stakeholders, in this sense, constraints of $Mconflict$ and $MNegotiation$ are presented:

Constraint 2. *MConflict = If the achievement of M_1 conflicts with the achievement of M_2 the relation between M_1 and M_2 can be defined as $Conflict$ (M_2, M_1).*

Constraint 3. *MNegotiation = if $MConflict \neq 0$, then M_i must be negotiated with CS{Stk} and SoS{Stk}, the relation between them can be defined as MNegotiation(StK, M_i), $1 < i < n$.*

Thereby, interviewees have available a guideline of rules of how to specify SoS mission for supporting the activities performed by them during the initial stages of analysis of a blended SoS. The challenges to this point of process also exist, as the mission threats what are linked with quality attributes from security, reliability, and dependability. Moreover, controlling and maintaining a mission focus and traceability to the mission throughout its lifecycle reduce the risks related to compliance and operational suitability (BUTLER; WOODY, 2017). Figure 33 evidences the centrality to strategy decision and how the other categories have contributed to Theme 1 to be developed throughout of SoS development lifecycle. The first decision is related to the type of SoS represented by `convergence of OT to IT` and by enterprise analysis and identification of SoS mission that can be found in `singularity mission strategy`. `Aware communication` is essential to the whole process and it is associated with `strategy decision` as well as it is part of `technical interoperability awareness` that support best practices the technical interoperability to requirements analysis. Because of this, `rationale` is a particular property of `technical interoperability awareness` and `technical interoperability quality` and `sustainable technical interoperability` are fundamental parts for maintaining the best practices in perfect operation throughout the lifecycle. In view of the discussed arguments, it is possible to get to a theoretical proposition for theme 1:

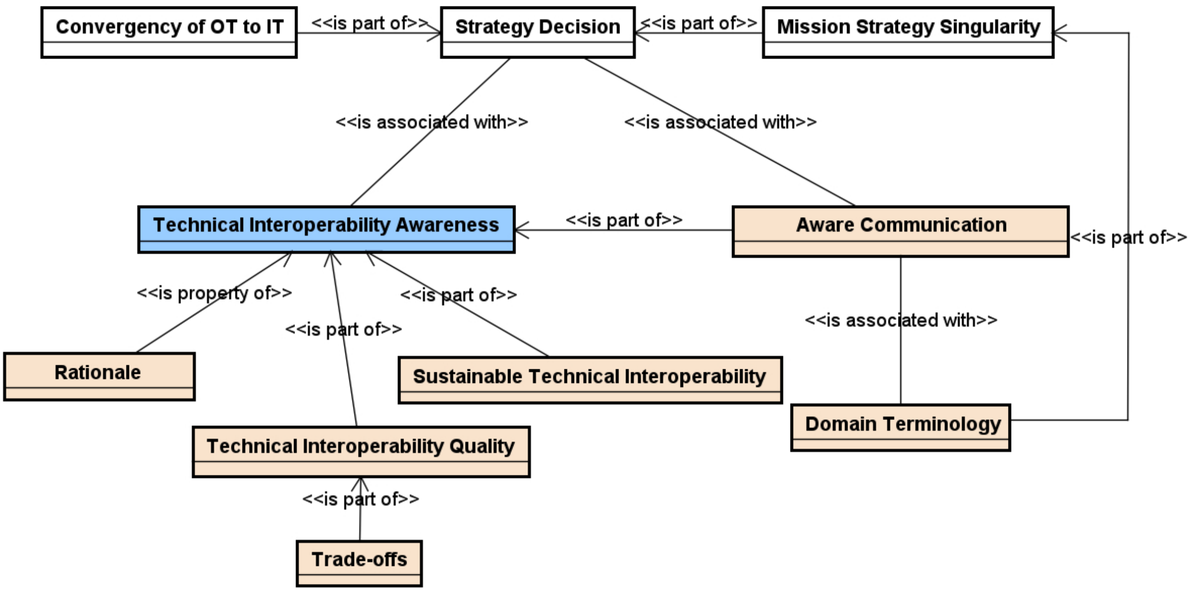


Figure 33 – Relationship between minor categories of the Theme 1

Source: Elaborated by the author.

Proposition 4. *Mission engineering substantially influences the analysis of technical interoperability requirements and both contribute to SoS interoperation at design time and runtime.*

5.4.2 Theme 2: Strategic alignment of operational issues

Such a theme refers to the strategic alignment of operational issues between business and technical interoperability requirements and that impacts the achievement of the overall mission. Figure 34 shows the relationship between categories identified in the selective coding process and indicates the alignment and challenge issues. Over again the alignment was one of the main issues raised by the interviewees. The `problem and consequences` faced by them are associated with `divergence of data and communication channel` that affect the interoperation between CSs. The latter is also associated with the category of challenges and the former is one of its reasons. Similarly, the lack of `aware communication` to induce challenges throughout SoS development lifecycle is the same in this process being associated with `domain terminology`. So, terminology composes the process of development of architecture alternatives, which supports the `sustainable technical interoperability` for maintaining a quite strategical decision and `technical interoperability awareness`. Our core category is part of the process of `technical interoperability quality` that is a great challenge for the analysis of technical interoperability requirements and have into their structure the inherent trade-offs of technical requirements that must be solved or at least known and registered in rationale.

The main concern of this theme is related to the technical infrastructure of communication (e.g., available network, data latency, shared application, data format, and transmission rates)

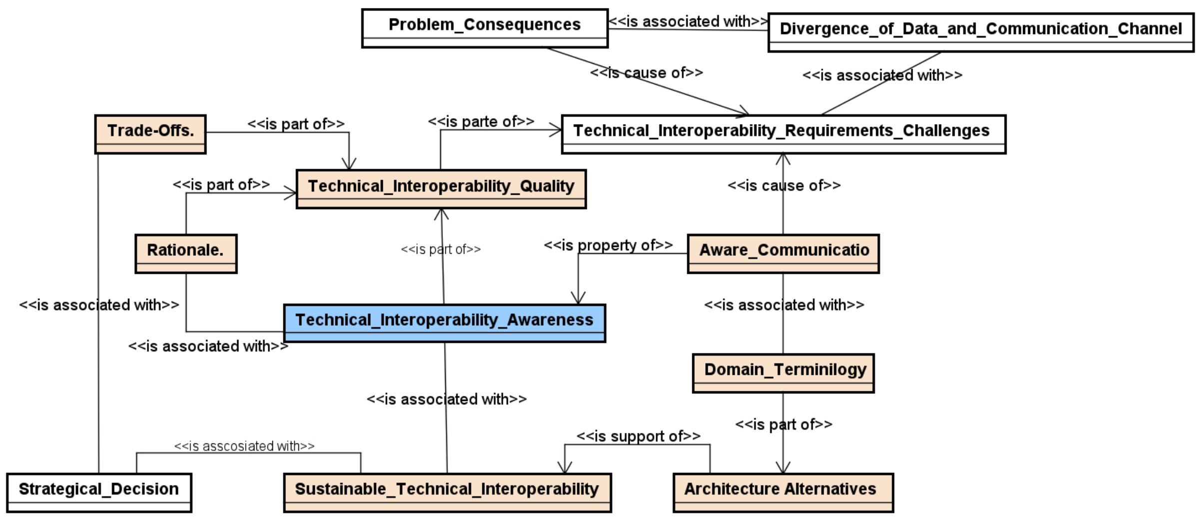


Figure 34 – Relationship between minor categories of the Theme 2

Source: Elaborated by the author.

addressed in 4.2.3 and the difficulties found to adopt protocol and standards that will support the interoperation of CSs. Barriers of technical interoperability requirements (cf. Section 4.2.9) still generate doubt among the interviewees and with the classification of the types of barriers there is one more way to improve their identification. To the interoperation, we understand that the interviewees demand not only refining the SoS mission as depicted in Section 5.4.1. Indeed, they desire the mechanisms that facilitate the specification and analysis of the technical interoperability requirements or at least the capability objectives to refine to smaller parts and obtain the technical requirements. In order to support them, the rules were designed to capability and requirements level using the specification based on the same philosophy used in ConMisOps, as displayed in Figure 35.

Such a figure illustrates the specification and design process of transformation of an SoS mission to requirements, being functional or quality attribute. The two first levels are common to ConMisOps plan and the others are specific to specification and design, their union generates the transformation to mission engineering level. Regarding top-down specification we can observe that specialization analysis to ensure the most important descendants of the problems space are being adequately addressed and the design is not driven by solution space (CROWDER *et al.*, 2016; JAMSHIDI, 2009). Hence, mission specification to the functional level supports the SoS team in the scope and cost definition and provides guidance for schedule development and control (PMI,). How aforementioned, parents' mission, capabilities, requirements are identified by asking *WHY* questions showing the importance of the descendants in the achievement of the SoS mission. The more constrained the technology resources space is, the more important it is to go through bottom-up analysis and synthesis. And since SoS architecture depends very heavily on the use of existing systems and infrastructure, it is critical that adequate attention is given to the bottom-up analysis (JAMSHIDI, 2009; SEBOK, 2014).

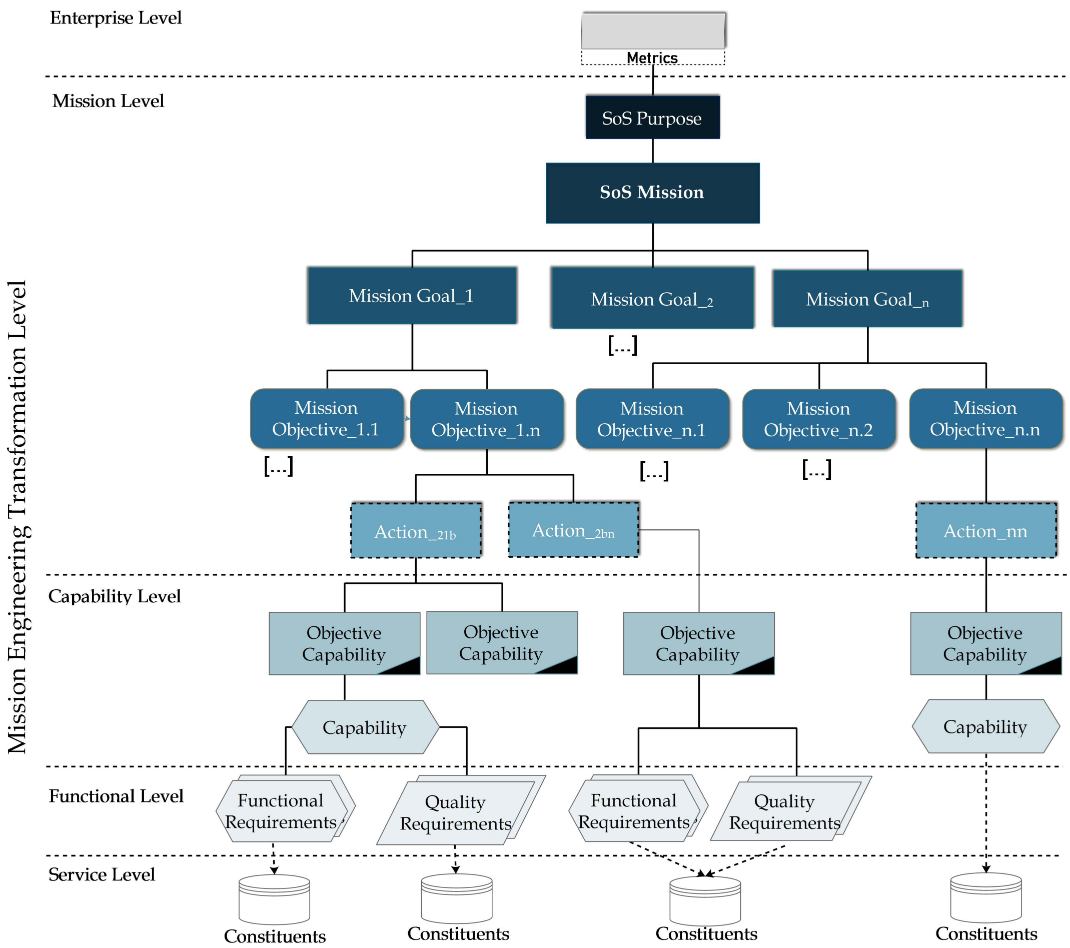


Figure 35 – Specifying capabilities and requirements from SoS

Source: Elaborated by the author.

Bottom-up is the analysis of the generalization, as well as by analyzing the aggregation. Using bottom-up analysis, backward traceability links can be established and they will aid to keep the accomplishment of the mission for verifying the consistency in the entire development, mainly when changes occur, identifying whether other products (i.e., specification) are updated. For instance, during architectural synthesis architectural decisions are performed, and these decisions cause modification to the architectural models. These models can have been derived from mission specification and they address requirements. The backward traceability can be useful to SoS architect to identify all required points to be updated in the requirement specification, capability specification, and in mission specification as well as in architectural documentation itself. However, traceability is a costly process and non-trivial to be carried out manually, mainly to complex systems as SoS. In the transformation, when the capability objective is broken down, two results are expected, a set of capabilities or a group of requirements, functional or quality attributes. The capability objective and a capability are defined as follow:

Definition 8. *Capability Objectives (CO): They are sometimes general statement(s) of one capability mapped in the desired operational task or mission that evolve over time (LANE,* *2017);*

Definition 9. *Capability (C): It is the ability to achieve the desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks (T) (BALDWIN, 2008);*

Capability and services have an important role in quality of SoS, since they can identify the various `substitutes` as discussed in Section 4.2.10 to each capability based on multiples possibilities and in concerns of stakeholders through architectural decision to choose the systems that better meet that concern. For example, a stakeholder has availability and performance as main concern; however, he/she prefers sacrificing performance to have the system running full time. In a coalition, we have three systems that provide the same capability, but only one ensures the availability, the other offers better performance but, there is a historic of constant falling. In this situation, the decision is made of an individual way by CS and in comply with the concern of stakeholders. Similar process occurs when identifying first the service desired to afterward identify the constituents that will provide it. The refinement from SoS mission to smaller parts (i.e., individual missions), selection capabilities and assigned to CS or still its acquisition is a process that can be fulfilled to various iterations of the SoS development life cycle. It helps partitioning SoS capabilities that can emerge of environmental changes or by derivation of the mission objective. These SoS capabilities also are decomposed into smaller parts to lower levels of requirements, which are allocated to the responsibility of the CS. Normally, this process is fulfilled when the CS does not have one necessary capability to accomplish the SoS mission or need to improve the capability provided by the CS to the achievement of the mission (i.e., SoS evolution).

Figure 35 was designed in five levels: enterprise, mission, capability, functional, and service. Although the enterprise and mission level have been addressed in theme 1, let's describe them together with others in a summarized way and address from capability ahead with the rules and so forth. The enterprise level is related to the organization environment that comprises the intention of the SoS stakeholders and organization involved. Considering that the ConMisOPs and the preliminary MisCon have been established, the SoS engineer, SoS requirements engineer, and SoS architect can keep them as a guide to gather stakeholders' needs, which are identified by stakeholders from business operations by means of structured processes (WALDEN *et al.*, 2015). On the other hand, if ConMisOps and preliminary MisCon were not established, SoS team needs to identify the SoS stakeholders and their responsibilities using the project plan. Besides that, environmental conditions or constraints must be identified, as well as a strategy for business or mission process, gaps in the goal and objectives should be reviewed. Afterward, the major stakeholders should be identified and managed. This is necessary to ensure that those stakeholders who have the most influence in the SoS development are available and willing to contribute to its development (BALDWIN, 2008).

SoS have emergent behavior and evolutionary development, and as a consequence stakeholders can arise, change, or only change the role through all of the development lifecycle of the SoS. Because of this, the major stakeholder's management is necessary. In addition, SoS type (i.e., acknowledged, directed, or blended) and the kind of development (i.e., greenfield, brownfield, closed source or blended context) when possible, also should be predicted in this view. SoS type directly affects all processes of acquisition and management of the capability and requirements of the SoS as well as their evolution and architectural adaptation itself over time. Since, the type is related to the organization of CSs and their collaboration, while the kind of development is related to how the SoS will be implemented. Stakeholders' intentions are transformed in needs (i.e., HLN) which are often competitive or conflicting with different priorities that should be solved by SoS engineer, SoS architect, and/or SoS requirement engineer. This can occur for various reasons, like one stakeholder can participate in more than one organization that supervises a CSs and has different roles in each one, besides the same stakeholders may change their roles and responsibilities at different times in the whole SoS development lifecycle (POULOUDI; WHITLEY, 1997). These HLN are statements that require attendance and selected set of capabilities and/or requirements will be selected to be assigned or developed by each CS that interoperate to accomplish the M.

Mission level describes the core of the SoS, for which all SoS design is based on. At the boundary of this view, non-conflicting $HLNs$ are mapped to only one M (cf.Rule 5). Hence, SoS architects must identify clear intermediate missions, constraints, and tasks to the missions to be met. Mission level is made up of different levels, starting with M and finishing when refinement achieves the leaf level, in other words, when the derivation reaches the individual mission. Intermediate level can be met by one or more than CSs, however in an individual mission, it is necessary to apply the `Rule 9`, which is attributed by only one CS. This CS, in turn, can use more than one capability to fulfill this mission, as reported in ensuing property:

Rule 11. *Different CS or CO can be provided by one same CS to fulfill its IM.*

In addition, missions are accomplished by emergent behavior, which occurs at the level of SoS and by the collaboration of two or more CSs in all SoS lifecycle, such relation we defined as:

Rule 12. *EB exist if and only if there is collaboration between at least two CS.*

Hence, emergent behavior unforeseen at design time can emerge at runtime and needs will appear. These needs shall be analyzed and translated according to abstraction level and complexity and can be denoted as:

Rule 13. *All DM may evolve over time due to EB property, arising a ND, which will be translated in NM, NC, or NR.*

Capability Level concerns with the capabilities necessary to the achievement of M, sometimes defined based upon desired operational task or in missions identified at the mission level (LANE, 2017). Functions required by capabilities are established by SoS design team as well as variability points needed to their execution. Capability is also general and can evolve over time into a more specific way. Specific diagram as use cases or capability model can be used to refining capability to requirement considering the operational environment. Similarly, at the mission level, needs can emerge through time during the evolution of the SoS or of the CS itself. Therefore, they also shall be analyzed, and if such capability or one non-existent resource is essential to the accomplishment of the M, needs are acquired. Otherwise, this capability is translated to various requirements as denoted by property:

Rule 14. *A C can be compound by a set of technical requirements, functional, non-functional, and constraints, which can be from hardware or software.*

These requirements can be more specified and attributed to CSs to meet the SoS need. In general, the decomposition of a capability objective can be performed according to the ensuing constraint property:

Constraint 4. *CDecompose = if $CO_{i,j}$ can be decomposed into $C_{1,2},\ldots, C_{m,n}$ and $C_{i,j}$ can be decompose into $C_{1,2}, \ldots, C_{m,n}$ or $R_{1,2}, \ldots, R_{m,n}$ and R can be decomposed into $R_{1,2},\ldots, R_{m,n}$ the decomposition between $CO_{i,j}$ and $R_{1,2},\ldots, R_{m,n}$ can be described as CDecompose($CO_{i,j}$ ($C_{1,2}, \ldots, C_{m,n}$,$R_{1,2},\ldots, R_{m,n}$)). CO is the parent C that is parent of R_i, that is defined as Parent (CO, C_i, R_i), if and only if, $CO_{i,j}$ is the parent capability of ($C_{m,n}$, $C_{i,j}$) and it is the parent capability of $R_{m,n}$, $R_{i,j}$ it is the parent capability of $R_{m,n}$. R_i is the descendant requirements of C, which can be described as CDescendant (R_i, C) if only if $R_{m,n}$ is the descendant objective of ($R_{i,j}$, $R_{m,n}$) is the descendant abstract requirement of $CO_{i,j}$, and $CO_{m,n}$ is the descendant requirement of $M_{i,j}$, where $1 < i < m$, $1 < j < n$.*

Operational level is related to the operational activities, also known as tasks or system functions and associated resources. The main characteristic of this level is defining the R inside more focused technical requirement and showing their logical and operational characteristics model of the interoperation (e.g., interface model provided and required, standards, resources). A mission objective and mission capability are organized in operational activities, and these activities are responsible for supporting the attainment of the M or some business goals. Hence, mission and SoS team derived the capability requirement for units capable of being allocated to a single CS (cf. Rule 9). However, if any capability requirement is not satisfied by one capability solution, it can directly impact the M, so an analysis of the capability gap must be fulfilled (BALDWIN, 2008). In addition, all capability requirements need to be validated to their specific proposal and by the authority appropriated before being assigned to a CS. During the transformation process some rules and one proposition must be followed to preserve the integrity of the refinement as described below:

Proposition 5. *Each requirement must be defined in measurable terms, contain applicable criteria, and be sufficiently detailed to be used as design criteria and architectural synthesis activity.*

Rule 15. *A PR can have one or a set of DR and they may evolve over time due to EB and EDP.*

Rule 16. *Every DR that are leaf requirement must be allocated to at least one CS and this requirement cannot be refined further into other DR.*

Requirements are often conflicting, mainly when they have various stakeholders to be met with multiple interests. One form of minimizing such a problem is to identify these conflicts in initial stages and negotiate with stakeholders to solve or at least find one intermediate solution to cease it. This conflict is denoted by:

Constraint 5. *RConflict = If R_i conflicts with any other requirement, the relation between R_i and R_n can be defined as $Conflict$ (R_n, R_i).*

Constraint 6. *RNegotiation = if $RConflict \neq 0$, then R_i must be negotiated with (S), the relation between R_i and S can be defined as $RNegotiation$(S, R_i), $1 < i < n$.*

At the end, service level presents the *CS* identified through capabilities of services gathering in operational and capability level to the allocation boundary to furthermore meet one capability or service to design the SoS or the capability requirements to evolve the SoS itself. In addition, the services provided by the CS are presented and defined by the next properties:

Rule 17. *All CS to provide at least one S and one same CS can collaborate with more than one S.*

Rule 18. *A CS can provide more than one C (i.e., CS Capability) to a same SoS. SoS can receive more than one C or S of one same CS.*

5.4.2.1 Constraints and Pattern for ATLANTA Theoretical Framework

Inside of GORE context analysis, the main pattern used to refinement is AND/OR. Based on this pattern, we adapted and extended the preview patterns and principles proposed by Zhu *et al.* (2017) to apply in ATLANTA Theoretical Framework. Besides the patterns, constraints that can be limited to the development of SoS or the resources of technologies and infrastructure available were also designed to support the interviewees in the application of analysis of technical interoperability requirements during the SoS development. In this process of execution, dependence of activities is common and, in the rules, this dependency is also presented and described of the general way as:

Rule 19. *GDependence = if the fulfillment of ϑ_i is dependent on fulfillment ϑ_j, the relation between ϑ_i and ϑ_j there will be, if and only if, there is one dashed link from ϑ_i and ϑ_j and* *starttime and endtime of ϑ_j need be first that ϑ_i. This relation can be described as Dependence (ϑ_j, ϑ_i), where, ϑ cab be a task, mission, capability or requirement;* $i = \{1,2,3,\ldots m\}$*; and* $j = \{1,2,3,\ldots n\}$.

A refinement of these general rule to mission dependence is presented below and it can be always applied some element needs to wait another to continue the activity. Although we used the same element in different levels, Rule 19 can be also adopted to dependence of different elements, as showed in Rule 21.

Rule 20. *Mission Dependence = if the accomplishment of M_1 requires support of accomplishment of M_2 the relation between M_1 and M_2 can be defined as Dependence (M_2, M_1).*

Rule 21. *Capability Dependence = if the accomplishment of M_1 requires support of accomplishment of C_2 the relation between M_1 and C_2 can be defined as CDependence (C_2, M_1).*

We designed the AND pattern and adapted from the Zhu *et al.* (2017) the OR, parallel and sequential to COnMisOPS and Mission Engineering Transformation level. Understanding the pattern is important, because they are presented to our constraints of time, event, requirements, and so forth. The first pattern denoted is the OR, after AND, parallel, and sequence and all are described in a generic way as much as possible, since the most transformation property of refinement is similar to all levels.

Pattern 1. *OR Pattern: if Decompose (ϑ $\vartheta_{1,2}, \ldots, \vartheta_m$), and any of ϑ_i is achievement, ϑ is achievement, the decomposition is ORPattern;*

Pattern 2. *AND Pattern: If Decompose (ϑ $\vartheta_{1,2}, \ldots, \vartheta_m$), and two or more ϑ_i is achievement simultaneously, the decomposition is ANDPattern*

Pattern 3. *Parallel Pattern: if Decompose (ϑ $\vartheta_{1,2}, \ldots, \vartheta_m$), all of ϑ_i are achievement, ϑ is achievement, and any ϑ_i is accomplishment independently without sequence relation, the decomposition is ParallelPattern*

Pattern 4. *Sequence Pattern: if Decompose (ϑ $\vartheta_{1,2}, \ldots, \vartheta_m$), all of ϑ_i are achievement, ϑ is accomplishment, and the starttime of ϑ_i is later than endtime of m_{i-1}, the decomposition is SequencePattern.*

The priority principle is passed from parent to descendant in the transformation process and when it is violated, the model has a problem of reliability and integrity. Besides, the model presents a problem to consistency and completeness of data of mission needs that will affect gathering capabilities and requirements, technical interoperability requirements, and until constituents that better meet the concerns of stakeholders. Architecture cannot be designed not to meet all the stakeholders needs and business objectives and hence, the SoS mission can failure. So, we describe priority as a constraint that must be met throughout of SoS development lifecycle as:

Constraint 7. *Priority: if the priority of $\vartheta_1 > \vartheta_2$, the priority of $D\vartheta$ of ϑ_1 cannot be lower than of $D\vartheta$ of ϑ_2. The similar principle is applied to all kinds of needs, requirements and capabilities to decomposition.*

The process of prioritization support to constraints of time of processing from SoS mission to find the CSs and implementation of requirements in the evolution process of an SoS. Each time procedure is isolated, and because of it these two properties of constraints were designed as presented:

Constraint 8. *Time: The processing period of DM should be within the processing period of PM. ORPattern or ParallelPattern: the starttime of M_i cannot be earlier than the starttime of M, and the endtime of M_i cannot be later than endtime of M. In SequencePattern, the starttime of M_1 cannot be earlier than startime of M, and the endtime of M_n, cannot be later than the endtime of M (M_i is descendant of M). To ANDPattern, the starttime and endtime of two or more M_i cannot be latter than starttime and endtime of M.*

Constraint 9. *Time Requirement: The implement period of DR should be according to their prioritization and within the development period of PR. ORPattern or ParallelPattern: the start of CS Capability (CSC) cannot be earlier than the start of R_i, and the finalization of R_i cannot be later than the finalization of CSC. In SequencePattern, the start of CSC cannot be earlier than the start of R_i, and the finalization of R_i, cannot be later than the finalization of CSC. (CSC is dependent of R_i). To ANDPattern, the start and finalization of two or more CSC can be latter than the start and finalization of R_i.*

Other three important principles are related to the use of the resource, which usually is a technical interoperability requirement. The occurrence of an event and in what condition they can take place considering the *ORPattern* and *ParallelPattern*. Such event is concerned with environmental changes or any trigger that can initiate an SoS mission. And, conditions that are the technical infrastructures or any condition necessary to successfully execute the SoS mission. Each one was defined as following:

Constraint 10. *Resource: The union set of Resource of M_i is equal to the Resource of M, that is $\cup Resource_i$,$(1 \leq i \leq n)$= Resource)*

Constraint 11. *Event: In ORPattern or ParallelPattern, the union set of Event of M_i is equal to the Event of M, that is $\cup Event_i$,$(1 \leq i \leq n)$=Event. In SequencePattern or ANDPattern, the union set of Event of M_i is equal to the Event of M, that is $Event_i$=Event.*

Constraint 12. *Condition: The union set of condition of M_i is equal to the Condition of M, that is $\cup Condition_i$, $(1 \leq i \leq n)$=Condition.*

At the end, the complete and correct analysis of the models to design the architecture needs to be performed, because of this, the model is checked to its completeness to verify the definition and consistency to maintain the model free of conflict.

Definition 10. *Model Completeness: The modeling is considered complete in relation to the decomposition, if and only if each DM that is not IM refined to at least two M and IM refined until a set of R is unambiguous, consistent, feasible, traceable, and verified.*

Definition 11. *Model Consistency: The priority of each descendant can be higher than the priority of parents, and descendants cannot be conflicting.*

5.4.2.2 Analyzing Technical Interoperability Requirements

Interoperability requirements are specified into operational level with regards to architectural analysis when they are refined to technical interoperability requirements. Through them, SoS are able to physically communicate with each other and, if needed with other infrastructures. Each domain has a specific need that should be considered to meet the SoS's objectives and mission. The main goal is to achieve the SoS mission without losing its purpose and providing the values for the company or group of companies involved. For this, technical interoperability requirements must be identified together with the stakeholders in an integrated way inside the process of design and interoperation. For our interviewees, the isolated process can introduce failures and gaps between levels and to each attribute that is analyzed (i.e., mission objective, capability objective, capability, requirements). Furthermore, the threats and barriers (i.e., technical, architectural, economic, legal, and cultural) of technical interoperability requirements when not identified on time brings forth serious problems to interoperation and communication among the CSs. In either case, the systematic identification of the problem support in planning the possible continence, improve the interoperation and continuation from the overall mission.

Figure 36 illustrates three dimensions of the technical interoperability frame for addressing Theme 1 during system evolution with respect to strategy and its use is derived from the Triplet model. The first dimension is the **nature of the technical interoperability requirements** discussed in Section 5.4.1, which refers to stakeholders needs and understanding, beyond the enterprise analysis. The second dimension concerns to **technical interoperability requirements strategy** is regarded to multiple stakeholders views. They seek understanding together with mission and SoS team the importance of technical interoperability stakeholders and what value they are providing to the enterprise to achieve the SoS mission. The connection of this dimension is with the mission engineering. In working with this dimension, the stakeholders are clear about how each capacity or requirement contributes to the mission and with what value it is contributing, while maintaining the return to the organization itself. Consequently, the process of negotiation and prioritization becomes much simpler when the contributions for specific purposes are known. The last dimension is related to SoS architecture and concerns with **technical interoperability requirements in use**. Particularly, the analysis of the requirements

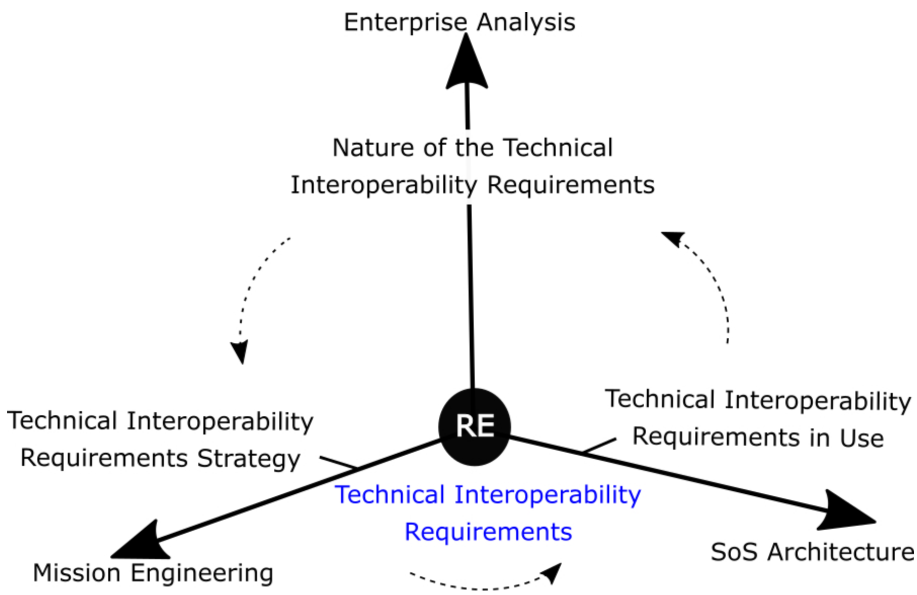


Figure 36 – Analyzing technical interoperability requirements using interoperability frames

Source: Elaborated by the author.

of other themes for the operational environment to optimize the accomplishment of SoS mission through technical infrastructure. In these situations, the SoS evolution offers new opportunities for renovating and diversifying system functionality over time (i.e., needs, capabilities, requirements emerged at runtime because of the emergent behavior and non-adaptation), building on information exchanged between collaborative or communicating systems. New needs can arise and they need to return at design time and this process is incremental interactive in all directions, as represented by the dotted arrows, representing the evolutionary development of the SoS itself. To better understand the technical interoperability requirements and how they are organized at design time and at runtime, we developed the ($QM2_{TI}$) that discussed in the next section.

5.4.2.3 Quality Model of analysis of Technical Interoperability Requirements driven by Mission Engineering ($QM2_{TI}$)

We design the abstract quality concept to support the requirements analysis in the context of ATLANTA Theoretical Framework and following the guidelines DUMOD (VILLALBA *et al.*, 2010), which is a systematic process to develop domain-oriented quality models based on empirical data. The phase 1 (i.e., context analysis and work on field) and phase 2 (i.e., the definition of relevant quality characteristics) of the guidelines DUMOD were addressed as proposed by Villalba *et al.* (2010). Phase 3 of the DUMOD is not completely applied to this moment since the incidents emerged of empirical analysis and completeness assessment was performed by comparing the data with the literature-driven analysis and with the standards ISO/IEC-25010 (2011), ISO/IEC/IEEE-24765 (2017), ISO/IEC/IEEE-25012 (2019), Abuhav (2017). This phase only was performed with experts' assessment of each version of the $QM2_{TI}$ to maintain the completeness and consistency of the definition and of the comparison performed by the empirical analysis and literature-driven analysis. However, we understood that more refinement is necessary and new experts and practitioner's assessment will be carried out as future work.

To design the $QM2_{TI}$, firstly, it was analyzed the mission-driven technical interoperability requirements by considering main stakeholders of SoS, SoS types, communication and data infrastructure, hardware, software, service workflow, mission decomposition, and any resource necessary to the running of the SoS. On the second phase, we seeked to recognize the relevant quality attributes pointed out by empirical population driven by ISO/IEC 25000 standard definitions, more specifically the ISO/IEC 25010 with regards to products and systems quality and ISO/IEC 25012 that refers to data quality (cf. Appendix P). In the first version, 42 sub-characteristics not existing at ISO/IEC 25010 and four new characteristics were identified. In this first comparison, we disregarded some sub-characteristics, for example `response-time` for being attended into from the sub-characteristics *time-behavior* of the ISO/IEC 25010 and the `reuse of resource` also met by *resource utilization*. We compared the definition of all the sub-characteristics with each other resulting in the third version in 32 sub-characteristics and five new characteristics, which were again compared along with the relevant literature. After the literature analysis and comparison to other standards, we reached to the final version with 73 sub-characteristics, being 33 sub-characteristics were reclassified from the ISO/IEC 25010 and 40 sub-characteristics of others fonts including empirical analysis[5]. The 73 sub-characteristics were divided in two set of, namely: (i) **design environment quality** and (ii) **quality in the operational environment** (cf. Appendix O). Figure 37 shows the technical infrastructure development and deployment (e.g., networks, data centers, hardware, software, and operate) as well as the development of the SoS at design time to identify, mission, capabilities, and particular interoperability requirements to support the interoperation.

$QM2_{TI}$ in the **design environment quality** is performed by ten categories and 48 sub-characteristics, of which 23 are new sub-characteristics and three as new categories (i.e., interoperability, understandability, and sustainability). To better meet the needs of analysis of technical interoperability requirements, each category was individually analyzed and reorganized. In some cases, the same category as functional suitability, and performance efficiency, and interoperability will be presented at design time and at runtime with different sub-characteristics being met. Particularly, according to the interviewees, functional suitability is a key category to SoS, since, in addition to interoperating systems, they need to adequately provide functionality so that the SoS mission can be achieved. The definition of each requirement is available in Appendix O, as well as the relation of each code identified in empirical analysis in Appendix M. However, an important differentiation between compliance and conformance is performed. Both regarded to attend the regulations, laws, conventions, standards and so on. Regulations met through compliance is voluntary by the organization, otherwise, i.e., by conformance, it is a

[5] There is a cross between the sub-characteristics of empirical analysis and ISO ISO/IEC 25010, when this happened, it was always maintained as ISO/IEC 25010.

Figure 37 – $QM2_{TI}$ in design environment quality

Source: Elaborated by the author.

mandatory attendance. For instance, safety-critical systems as avionics, train, autonomous car perform the conformance to the standards ISO/DIS 26262 (2009), EN 50126 (1999) because the need of certification. On the other hand, we can use the standard OW3 or Transmission Control Protocol-Internet Protocol (TCP-IP), Open System Interconnection (OSI) and in compliance with such standards and protocols, because are a voluntary choice to join them is not a constraint or an obligation.

Figure 38 shows the $QM2_{TI}$ in the operational environment and refers to the characteristics at runtime and how the requirements need to be satisfied to maintain the SoS ever available to comply with the concerns of the stakeholders and business objectives to the accomplishment of global mission. In addition, help to manage mission needs not met by adaptability and variability

Figure 38 – $QM2_{TI}$ in operational environment quality

Source: Elaborated by the author.

points. These are essential technical interoperability requirements to an operational environment to maintain the operations further safe, free from risk that is a special issue to interoperation failure and mission. It is worth noting that not all requirements will be applied to all SoS, but we seek to build a quality model that is generic so that it can be used for SoS domain and similar systems. Based on these arguments and statements, it was possible to come up with two more theoretical propositions

Proposition 6. *Enterprise and technical alignment issues emerge as consequence of the inability of their interaction for comprehending the technical environment and stakeholders' needs and experiences of aware communication, and domain terminology can influence improvements in this relation.*

Proposition 7. *Technical interoperability awareness issues support communication and collaboration between CSs, maintaining technical sustainability and resources available for decisions over time.*

5.4.3 Theme 3: Control and management of mission at runtime to achieve outcomes

The last theme concerns the culmination of issues in relation to operational environment and outcomes mission obtained through the control and management of mission and of the technical interoperability requirements that evolve at runtime. Needs can emerge at runtime (cf. Rule 13) because of emergent behavior and have to be transferred at design time to be properly handled and implemented. Figure 39 presents the relationship between categories of this theme. Although the

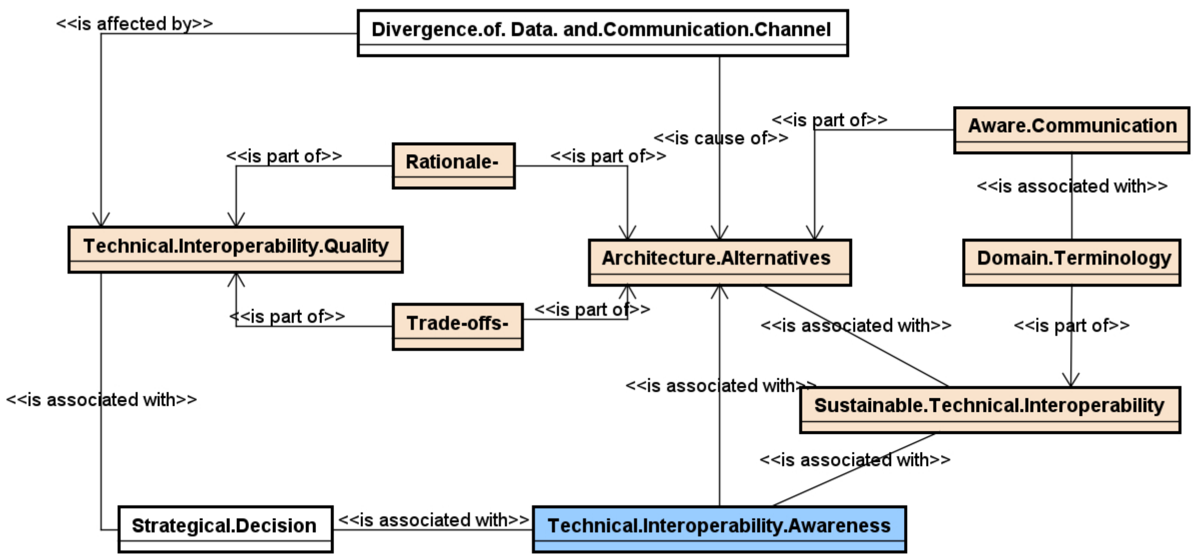


Figure 39 – Relationship between minor categories of the Theme 3

Source: Elaborated by the author.

minor categories of this theme have `divergence of data and communication channel`(cf. Section 4.2.3) and `strategical decision` (cf. Section 4.2.6), its kernel is architecture alternatives, since the concerns are related to the operation, by making the CSs communicate between them and exchange data and information to use such information to decision-making or provide one service or set of service to end-use with better quality. Indeed, all categories are directed to support the interoperation or as *part of* or as *associated with*. Only divergence of data is the cause of the choice of alternatives while also affect the quality attributes; however, to improve the mission outcomes the technical interoperability awareness needs to rise, plan the constituent systems, develop the technical infrastructure such as software, communication, and hardware and also the technical awareness in time of operation, control and management.

To support the building of this awareness, we present the ATLANTA$_{M3}$, a mission management model in Figure 40, which is split in four layers and our main focus is to requirements at runtime and how it can support the architectural evolution. The first layer of mission management is concerned with what purpose needs to achieved, and action plans to fulfill the business objectives and goals, and also to the current situation, which must be handled. This layer intends

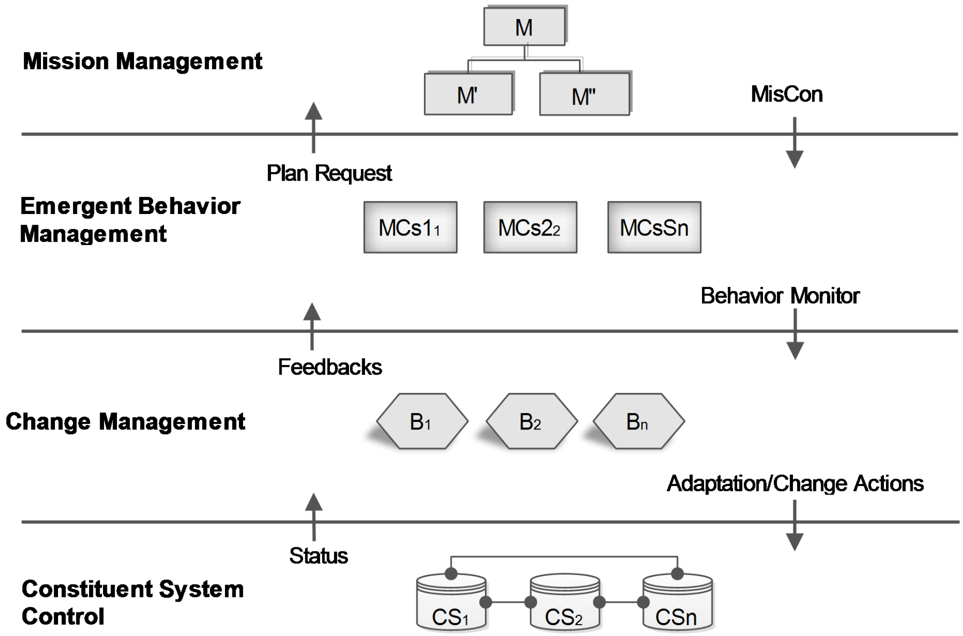


Figure 40 – Layer architecture model for multilevel mission management in mission engineering

Source: Elaborated by the author.

to describe the mission plan (i.e., MisCon) to perform the SoS mission, and business objective. The MisCon is created in response to the requisition of the emergent behavior management layer. In MisCon the foreseen emergent behavior is described as variability's points, in which the adaptation will occur according to external or internal modifications. Hence, requirements adaptation refers to the ability of the SoS to be modified by modification of their entities, assessing the process of monitoring and changing, which have different and complementary function inside of the evolution process.

In the context of requirements at runtime two areas (i.e., change requirements and monitor requirements), most of the time, are treated as similar, however, they have different and complementary functions in the evolution process. The former covers requirements used beyond guidelines to monitor at runtime, but as tactics plans (i.e., rules) on how the adaptation should take place to keep satisfying SoS requirements, whereas the latter is concerned with checking if SoS requirements defined at design time are being achieved at runtime (GOLDSBY *et al.*, 2008; BENCOMO *et al.*, 2010). Autonomic computing as applied in Affonso et al. (AFFONSO *et al.*, 2015) and (CHENG *et al.*, 2014) has been exploited to ensure the adaptability when constituents or mediators are modified, switched, inserted or removed. In addition, expected behavior at the SoS level are normally already identified inside this plan. Otherwise, due to properties inherent of autonomy of the constituents the emergent behavior needs to be monitored in the whole lifecycle of the SoS to control different organizations even as deal with exceptions cases, like violation or failure of a requested capability (ESTAHANI *et al.*, 2013) This monitoring permits information to identify which information is not adapted for SoS level bringing forth new needs to analysis and when it is needed to implement changes, the request is sent on the planning stage. This layer establishes as artifact a report of behavior monitor, which registers the constituent's configurations and/or describes its internal or external behavior required for fixing or addition in the SoS.

The change management layer is accountable for receiving the solicitation of changes and executing them in reaction the environment modification (i.e., flexibility), needs reported of the mission operational (i.e., lower layer), or still internal changes (i.e., adaptability) with alteration of mission, capabilities, or requirements to preserve or enhance value delivery in the attendance of variable environment, capabilities, and requirements. In addition, stated reported from the higher layer and from the lower layer are also verified here, since this state can be changed in function of other needs. For example, when a component of a constituent fails and there is no fault tolerance control, if that component was essential to provide capacity to SoS, that capacity is not available and the SoS mission may also fail. In this case, it would be necessary to work with the redundancy of this component, so if it fails, a duplicate is immediately switched from idle to active mode (KRAMER; MAGEE, 2007). We consider this layer as an organization of tactics plans responsible to answer to changes in SoS level that alter negative fashion the achievement of global mission as well as an improvement the SoS or its adequacy to external environmental like laws, regulations, or agreements. Changes in SoS should be performed on two levels, i.e., considering the changes in adaptation and flexibility, which most of the time will be implemented by the constituents or SoS architecture. Constituents distribution induce to issues of latency, concurrency, robustness, and reliability carrying out the necessity of maintaining part or complete autonomy to the constituent, whilst retaining the consistency of the SoS even in operational scenario changes. For instance, during the interoperation between constituents, in its adaptation process it is possible the introduction, swap, or removal of other constituents, capabilities, or even new requirements to adequate the SoS the stakeholder's needs and the value, which the user and customer desire to receive (GEORGIADIS *et al.*, 2002).

At the end, the control layer is a lower one and consists of physical systems, called constituents including the physical layer, which covers the issues to technical interoperability between and among constituents. This layer receives request to changes to comply with SoS necessities to perform its mission and also sends to higher layer events and status of the constituents. Events and status could be monitored by a supervisor system that reports the modification to a higher layer according to requisitions, as well as traces them from constituents to SoS mission and business goal. By monitoring the capabilities of constituents, the evolution of these capabilities provided by that constituent can be controlled, or at least identified before leading to mission failure. For this, MAPE-K model and machine learning become essential to reduce human effort in controlling behavior and minimal conditions to the interoperation of constituents. From this discussion, one can define yet another proposition.

Proposition 8. *The greater the awareness of collaboration for technical interoperability, the* *easier the analysis of technical interoperability requirements.*

5.5 Summarizing the Abstract Model of the ATLANTA Theoretical Framework

In this section, we summarize the relationship of the three themes that are concerned with the findings of our research question that is again presented in the gray box.

Research Question: *How can we specify a theory to support the analysis of technical interoperability requirements with the use of SoS mission engineering?*

ATLANTA Theoretical Framework has been developed based on empirical analysis with the support of literature-driven analysis as described and justified in Chapter 3. The Analyses of technical interoperability requirements driven by mission engineering were the main concern raised in our investigations into the context of SoS and similar systems, since they are two fields considered essential to development of these systems. By considering the mission engineering concept, we portray in the SoS domain a reality strategical knowledge known by the interviewees enabling more interaction between them and their needs to transform that tacit knowledge in a theoretical knowledge for SoS. The data analysis driven by an interaction and feedback with the interviewees provided the possibility of developing a theory that now enables analysis not possible before, such as: barriers of technical interoperability requirements; analysis of technical interoperability requirements sustainable; and specifics approaches to perform the situational analysis to SoS domain. ATLANTA is a theory that can be used as any SoS that have central control, or only for specific parts to meet the essential needs of the mission team, SoSE team, architects, requirements engineers, and so on. We are aware that processes, approaches, tools, among others are necessary to apply such theory; however, in this research the focus was to identify the problem and mechanics to support it. As in our literature there is a lack of theory that supports the understanding of the analysis of technical interoperability requirements driven by mission engineering, we decided by advancing in this direction.

Figure 41 shows the difference surrounding the three themes. Enterprise and technical alignment issues receive positive and negative influence from Theme 1 and connect themes one `mission engineering influence` and two `enterprise and technical alignment issues`, as well as link, themes two and three `increasing awareness to control and management`. Both positive and negative perception for each theme in relation to each other is shown into the abstract theoretical model from Figure 30 in Section 5.3. The negative perceptions are presented in red mark while the positive is blue. From `Theme 1` to `Theme 2`, the negative viewpoint is focused by not understanding the needs of stakeholders or not meeting the main needs of the most influential stakeholders, which brings problems for all themes and operational alignment. Indeed, SoS have many stakeholders, therefore, knowing how to choose the most influential ones for decision-making can bring great gains for the whole SoS and for the stake-

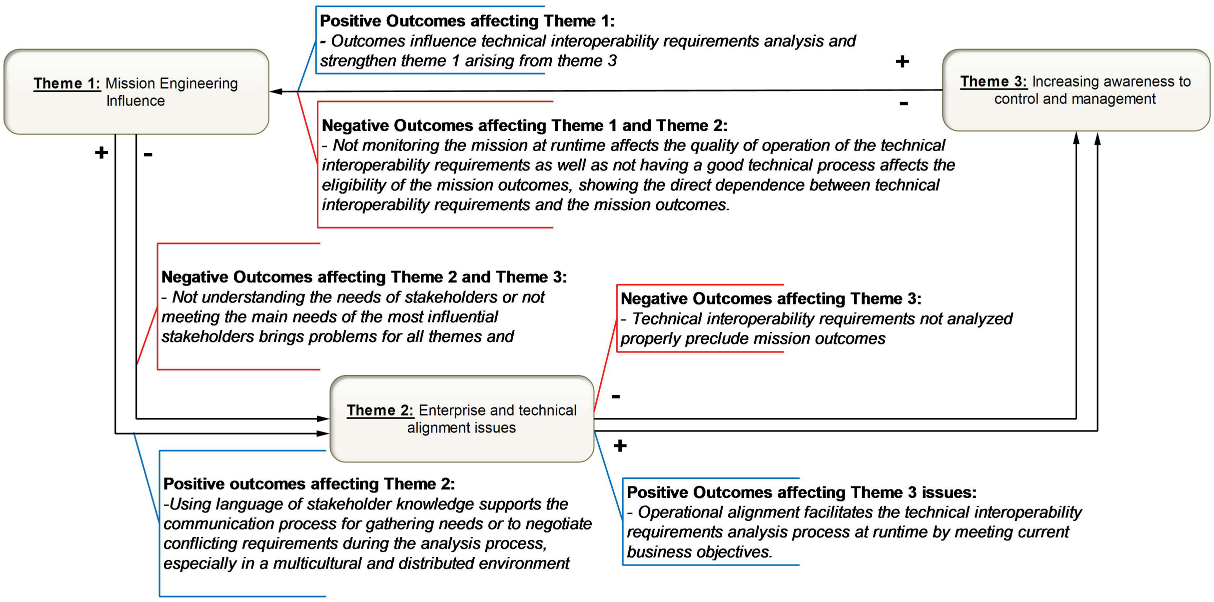


Figure 41 – Abstract model of the ATLANTA Theoretical Framework

Source: Elaborated by the author.

holders themselves. Such a procedure influences the technical interoperability requirements that when not analyzed properly preclude mission outcomes, affecting negatively the link between theme 2 and theme 3. Lastly, the Theme 3 in operational level affecting Theme 1 and theme 2 by not monitoring accordingly the mission at runtime, because it affects the quality of operation of the technical interoperability requirements as well as not having a good technical process affects the eligibility of the mission outcomes, showing the direct dependence between technical interoperability requirements and the mission outcomes.

In either case, positive perceptions support the applicability of the analysis of technical interoperability requirements and the evolution of SoS over time without any opposition. Between Theme 1 and Theme 2 the use of language, stakeholder knowledge help the communication process for gathering needs or to negotiate conflicting requirements during the analysis process, especially in a multicultural and distributed environment. Hence, this environment maintains further coordination to the operational needs facilitating the analysis of technical interoperability requirements process at runtime by meeting current business objectives and a positive interaction from Theme 2 to theme 3. Conversely, outcomes influence the analysis and strengthen Theme 1 arising from Theme 3.

5.5.1 Conceptual Implication of the ATLANTA Theoretical Framework in the State-of-the-Art and State-of-the-Practice

One of the main motivations to perform this work was the difficulty to find in literature studies that connect the areas of Enterprise (or organizational) (CHEN; DACLIN, 2006), SoS Mission (KUMAR; MERZOUKI R.AND BOUAMAMA, 2017), SoS (LANE, 2017), RE (NCUBE; LIM, 2018), interoperability (TOLK, 2003), technical interoperability requirements, architectural analysis (CHEN *et al.*, 2013), mission engineering (SOUSA-POZA, 2015), and operational environment (ESTAHANI *et al.*, 2013) to treat the stakeholders' needs at design time and at runtime due to emergent behavior of SoS and evolution of CSs. Since SoS rely on technical infrastructure and technical interoperability requirements to support the interoperation, such requirements and infrastructure must meet specific quality attributes, beyond the protocols, standards, networks, interfaces, and others in a crosscutting way to keep life and the assets safe and security.

The choice of ground theory to the construction of a theoretical interpretation of the investigated phenomenon is justified, because it allows to look for the problem by different viewpoints and perspectives (MILES *et al.*, 2014; WüRFEL *et al.*, 2016). After the first systematic mapping (cf. Section 3.5.1.1), it was noted the need of a deeper exploratory investigation to satisfy the gap existing in the literature. When we finalized our survey it was evident the difficulties to perform the analysis at the interoperability level, beyond making clearer the divergence of the concepts among industrial and academics respondents. In general, few were the studies found that addressed the thematic of this Thesis as mentioned in Section 2.6 and Section 3.5.1. In this sense, this research advances the states by integrating mission, requirements, and architecture (i.e., Triplet Model) using the concepts of the mission engineering discipline that inserted a process from enterprise analysis at runtime and from runtime to enterprise analysis.

Figure 1 (cf. Chapter 1) shows the diverse isolated aspects that were treated separately of the 11 categories and integrated into ATLANTA Theoretical Framework. Therefore, the theory for analysis of technical interoperability requirements driven by mission engineering *dealing with multilevel of technical interoperability awareness* emerged of the data incidents, showing representativeness of the phenomenon. Thereby, only this type of methodology captures the essence in a substantive area to meet the most relevant needs and aspects involved, as described by the interviewees in their verification process. The categories of the ATLANTA were organized in three phases i.e., **the enterprise analysis process**, **mission refinement and interoperability analysis**, and **technical interoperability analysis**, to better support its application to the analysis of technical interoperability requirements as a detailed process of the Figure 32:

- **Enterprise Analysis:** The core of the ATLANTA is the Triple Model, which was refined to $Triplet_T$. To initiate the analysis process it is necessary to identify the stakeholders and the type of SoS. We are working with the blended SoS. Enterprise analysis is performed by the user the Forces of Porter and 5Cs model, other approaches such as Swot also can be applied when necessary. After finalizing the identification of the needs and opportunities and threats and weaknesses of the planning of the ConMisOps, it is recommended to perform the description of the MisCon that will go at runtime. The planning and organization for the prioritization of the SoS mission can be discussed among stakeholders and only after the identification of the mission that new activities will be performed. Next, the

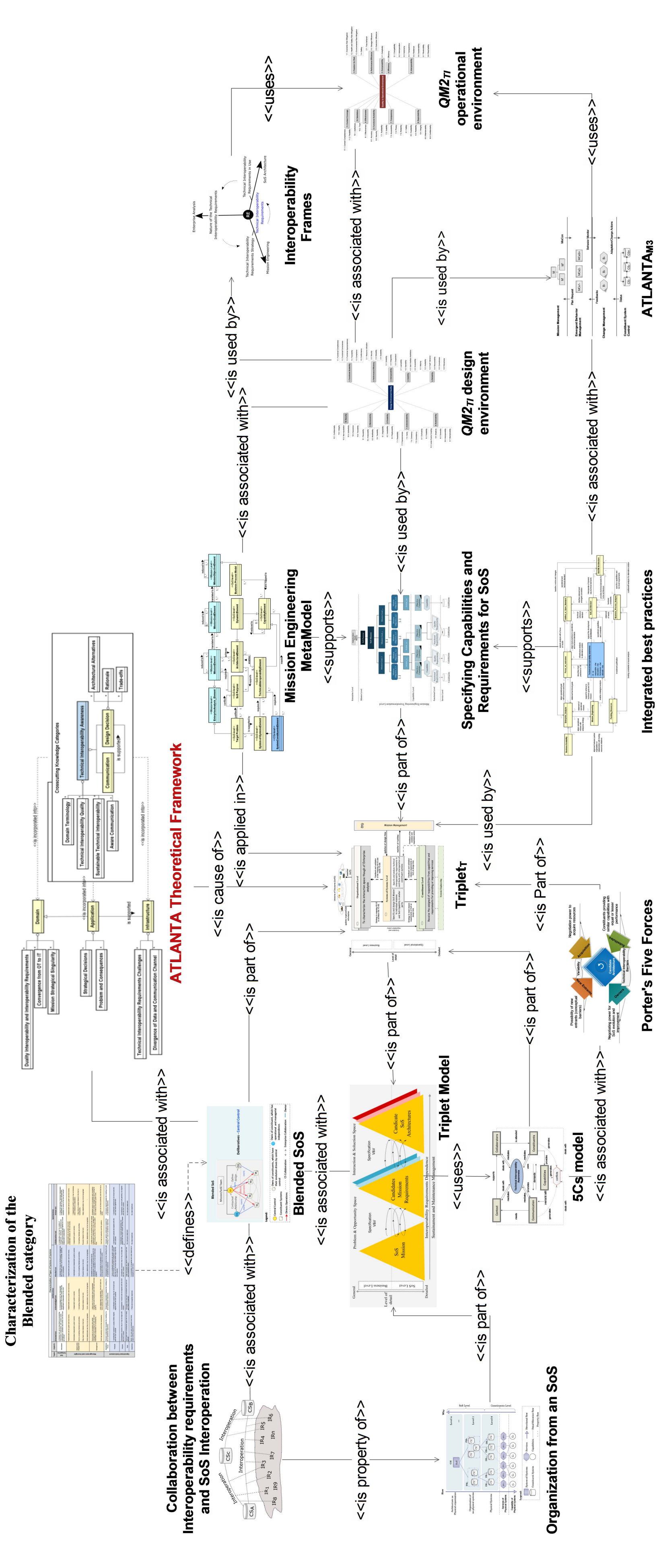


Figure 1 – Contributions and its relationships

Source: Elaborated by the author.

analysis of mission threats initiate that also are registered in MisCon, the strategical planning of acquisition is verified and constraints are also raised in strategical level. All strategical planning of SoS and its mission is performed in enterprise analysis to other stages, decisions are more technical than strategical, except at runtime.

- **Mission refinement and interoperability analysis**: the second group is related to the mission and initial gathering of the requirements. In this phase, the metamodel and mission engineering transformation are used, beyond the $QM2_{TI}$ model. During the mission refinement, specification of capability or requirements will occur in interaction with enterprises, specifically for interoperability requirements. The transformation property, constraints, and definitions can also be applied to support the specification that uses the scenario proposed in Section 4.2.6. Moreover, the interoperability requirements should be gathered by using multiples view to better meet the stakeholders' needs and the communication process, and the requirements model makes the procedure clearer and reliable; beyond improving the requirements architecture quality (cf. Section 4.2.6).

- **Technical interoperability analysis**

  The last stage started at design time to maintain a long time at runtime until emerging new need, mission, capability or requirements. Interoperability requirements are refined to achieve the technical interoperability requirements not identified yet. New ones can be gathered and others can be changed because of the complexity of systems. The interaction is still constant to both levels, verifying the design decision and the rationales that had already been registered since enterprise analysis. Technical interoperability frame interacts with all other findings of the analyses of technical interoperability requirements including mission management in mission engineering (cf. Section 5.4.3) and $QM2_{TI}$ in the operational environment. Although all level interacts with each other, this one has more interaction, because of emergent behavior and evolutionary development that often provokes the appearance of new needs to map or threats to mitigate.

5.6 Final Remarks

This Chapter presented the characterization of central category and also the construction of the emergent theory, named ATLANTA Theoretical Framework, which relates to other ten categories addressed in Chapter 4. We structured and described an abstract model for ATLANTA and answered our research question. The abstract model introduces three major themes established from ten categories emerged of the empirical analysis and literature-driven analysis, which support the process of technical interoperability requirements analysis. By presenting new concepts, metamodels, transformation properties, the technical interoperability frame was proposed and discussed together with $QM2_{TI}$ and $ATLANTA_{M3}$. In the next chapter, the discussion and conclusion will be addressed.

CHAPTER 6

Conclusions

6.1 Initial Remarks

SoS emergent behavior results from the interoperation of CSs and their inherent and continuous evolution, which often leads to needs and requirements (including technical interoperability requirements ones) in the operational environment due to environmental changes. For the SoS viewpoint addressed in this research, technical interoperability requirements are interoperability elements that support the interoperation, and although they have been widely studied (FORD *et al.*, 2007a), their analysis process faces several problems and challenges, mainly because RE is met individually for each type of SoS (NCUBE; LIM, 2018), including blended SoS. Most analysis processes have described technical interoperability as either a level of maturity (TOLK; MUGUIRA, 2003), or synonym of interoperability requirement (MALLEK *et al.*, 2012) that can include other interoperability elements (e.g., semantics and syntactic). In order to comprehend the analysis of technical interoperability requirements and support enterprises and interviewees, we have explored the analysis process and how it can be applied from the enterprise context to SoS and similar systems using mission engineering. The business and technical stakeholders must be considered in initial stages (KNODEL; NAAB, 2016), as well as the architectural decision and rationale (BASS *et al.*, 2012). Moreover, quality attributes (BIANCHI *et al.*, 2015), modeling languages (LANA *et al.*, 2018), and evolution of the SoS process that addresses such technical analysis process (LANA *et al.*, 2016) can be considered for the accomplishment of the SoS mission, as discussed in *QM2_TI* (cf. Section 5.4.2.3). This thesis defines the ATLANTA Theoretical Framework, a theory for the analysis of technical interoperability requirements driven by mission engineering from the perspectives of industrial practitioners and academics collaborating in industrial projects towards facilitating the understanding of the process of technical analysis. The remaining of the chapter is organized as follows: Section 6.2 synthesizes the contributions of the thesis and Section 6.4 provides the threats to validity found in the development of ATLANTA. Sections 6.5 and 6.6 discuss the limitations of the ATLANTA Theoretical Framework and detail the way they can be overcome and possible future research and extensions of the thesis, respectively.

6.2 Synthesis of Theoretical Contributions

The data collect and analysis were guided by the grounded theory together with the literature-driven analysis illustrated and discussed in previous chapters to present the construction of the substantive theory. The grounded theory supported the 11 categories empirically reasoned with rich information from at least 14 countries, in which the categories, sub-category, codes, memos, interview observations, and discussions with experts were performed and conducted. Table 15 displays the contributions of the research, besides knowledge about ATLANTA, including the Triplet Model, technical quality attributes, technical sustainability models, development of best practices, and transformation properties that can support the interviewees in the analysis in both early stages and operational environment. Nine propositions based on findings were established during the construction of the ATLANTA Theoretical Framework. They addressed the progress and evaluation of the internal reliability degree for the systematization of the grounded theory, as well as of the aim achieved, i.e., *build a substantive theory in technical interoperability requirements analysis driven by mission engineering from the perspective of the industry professional and researchers with partnership in industrial projects*; which is related to propositions that explain the analysis of technical interoperability requirements (see Table 17).

Figure 41 shows a way of improving the technical interoperability requirements analysis by means of challenge evaluation as outlined in Chapter 5, which focuses on the interaction among the main themes of the substantive theory. A similar situation arises in the definition of *technical, architectural, economic, legal, and cultural* barrier types (cf. Section 4.2.9), and *social, economic, technical, environmental, and individual* technical sustainability perspectives (cf.Section 4.2.5), which are challenges faced by the interviewees to perform the interoperation and maintain the technical sustainability. Other barriers were identified by Chen and Daclin (2006) in the INTEROP project for enterprise interoperability (i.e., technological, conceptual, and organizational). However, they are very comprehensive types of technical interoperability requirements that have a more specific one. Our propositions are particularly related to technical interoperability awareness for improving the analysis over time maintaining the resources and services available to end-user. Services provided by SoS add value to stakeholders and end-users, and the whole service infrastructure supports providers, thus, facilitating the process of awareness.

A comparison between our results with the literature revealed the academy is focused on understanding the relationship of SoS mission with the technological level and its impact, as addressed in Section 2.6 and discussed in Section 4.2.10. Because of its importance, if the focus is only on one part of the development process, mission threats can be raised in the other

Table 15 – Main contributions of the constitutive elements of the ATLANTA

Source: Elaborated by the author.

Constitutive Elements	Main Contributions	Page
Duality between Interoperation and Interoperability Requirements	“Duality between interoperation and interoperability requirements" discusses concepts related to interoperability and defines technical interoperability requirements and MEDINA. It provides an overview of both SoS structure level ((Figure 16)) and collaboration between interoperation and interoperability requirements (Figure 17). As a theoretical category, it is a partial, but essential representation of interoperability occurring throughout the SoS development lifecycle.	66
Convergence between Operational Technology (OT) and Information Technology (IT)	This category has emerged from the necessity of more flexibility between OT and IT in the interoperability process and development of complex systems for industry 4.0. Such a collaboration poses reliability and dependability challenges for the SoS development and influence on the performance of SoS, its interoperability, tradesoffs with interoperability requirements, and accomplishment of the global mission. In this context, a new SoS type, called blended (Figure 18), has been designed and used to characterize of not only this category, but other of SoS ones considering the descriptions, management, oversight, operational environment, SoS design, and engineering and design considerations.	71
Divergence of Data and Communication Channel	Interviewees indicated the clear necessity of more quality data for the generation and knowledge of information. Organization connections must be available for the internal environment; however, problems arise if they are external or distant from urban areas due to the lack of an optimal coverage in several regions (some have blind spots). "Divergence", as a theoretical category, represents a reality of many domains in Brazil, such as agricultural, health, vehicles, tractors, etc.	80
Technical Interoperability Quality	This category supports the identification of the main technical quality attributes considered essential by interviewees during an analysis process of software and hardware. Some points, such as security, convergence challenges, technical challenges, and technical sustainability, which cover particular quality attributes to be analyzed and verified in ATLANTA were specified.	81
Sustainable Technical Interoperability	Sustainability is a difficult concept for technical interoperability requirements, starting from the dilemma of openness versus standardization of environments, protocols, and standards, which are the greatest current problem regarding interoperation. Towards dealing with it, we provide an overview of a generic model for technical infrastructure (Figure 20) for vertical and horizontal interoperability. Such a category was discussed in five perspectives (i.e., social, economic, technical, environmental, and individual) considering the development of SoS and technical issues to be addressed at both design time and runtime.	82
Strategical decisions	As a theoretical category, strategic decisions is related to a long process that starts with business alignment and misalignment between problem and solution and their intersection. It involves two viewpoints, namely, design time and runtime, and addresses the way problem and solution and their interaction should work in each of them. Facing such a scenario, we propose the Triplet Model (Figure 21), which enables requirements to act as a bridge between SoS mission and SoS architecture. Figure 22 shows the model and its relation with the architecture, and requirements are defined for a reliable requirements architecture. Figure 23 illustrates the architecture hierarchy and the relation for the identification and specification of the SoS mission in the enterprise analysis.	88

development stages, thus, compromising all other activities, including analysis and specification of technical interoperability requirements driven by mission.

Table 14 – Main contributions of the constitutive elements of the ATLANTA (Continuation)

Source: Elaborated by the author.

Constitutive Elements	Main Contributions	Page
Problems and consequences	Technical problems emerged and solutions were described during the analysis of data and interviews. Other consequences were addressed, due to the lack of solutions, which is common regarding technical interoperability. Some interviewees realized the urgency and importance of technical interoperability requirements analysis; however, its application is hampered by the high costs. Preference was given to relying on the quality of constituents or using of other approaches whose consequences are known.	96
Aware communication	This category is essential for the whole SoS development lifecycle, and, as a theoretical element, it connects both mission and SoS teams, so that they find a way addressed according to the stakeholders' needs specified in inter and intra-culture. Such an understanding can lead to errors found only at runtime, which implies higher costs. Stakeholders change during the SoS lifespan without knowing the rationale and nature of the requirements; therefore, the treatment can be costly and difficult. Moreover, improvements in the current languages, or development of a new one that covers a specific problem are discussed by the interviewees and other industrialists.	97
Technical Interoperability Requirements Challenges	This category identifies the challenges of technical interoperability, classified into three sets, namely technical interoperability threats, technical interoperability barriers, and SoS properties, which influence the interoperability requirements. Threats must be recognized and mitigated as soon as possible; otherwise, they affect the interoperation already established. Barriers hinder interoperation; therefore, one must work overcome them. The following five types of barriers were considered relevant by interviewees: technical, architectural, economic, legal, and cultural. Collaboration, evolutionary development, and distributed systems mostly influence technical interoperability. On the other hand, although architecture is considered less influenced, it is important for researchers, since it is the core of SoS and responsible for the interoperation process.	98
Singularity mission strategy	A small change in this category can significantly affect the entire SoS lifecycle. It discusses the mission planning and identifies the SoS mission through an enterprise analysis. A situational analysis evaluates the current conditions, after which, strategies and plans can be elaborated to meet the needs of the stakeholders. Figure 24 shows Porter's five Forces for awareness Interoperability requirements that enable the integration of new organizations, constituents of similar capabilities, owners, acquisition of an organization, and possible architectures generated from the variability and impacts of the barriers. ConMisOps plan and MisCon were structured and discussed for the planning of the SoS mission, and the interviewees reported a concern over such planning based on the enterprise's strategical plan. OKR was adopted because of its flexibility, scalability, compatibility, among other features.	100

6.3 Revisiting the Thesis Contributions

The main proposal of this thesis was the design of a substantive theory from experiences of industrial practitioners and academics involved in industrial projects for SoS and similar systems. Our aim was to fill a gap in the body of knowledge of studies that analyze technical interoperability requirements driven by mission engineering for those systems by understanding a phenomenon little explored in the relevant literature. In general, the research achieved the objectives outlined (cf. Table 1) for having answered the following problem question proposed:

Research Question: *How can we specify a theory to support the technical interoperability requirements analysis with the use of SoS mission engineering?*

Table 17 – Relation between aim and propositions

Source: Elaborated by the author.

Items	**Descriptions**
Aim	To build a substantive theory that support the process of technical interoperability requirements analysis driven by mission engineering for the design and development of SoS from the perspective of the industry professional and researchers with partnerships in industrial projects.
Proposition 1	The knowledge of the nature of technical interoperability requirements is a consequence of an aware communication between teams and stakeholders
Proposition 2	The nature of technical interoperability requirements supports the operational outcomes.
Proposition 3	The greater the strategic alignment in the operational context, the better the control and management of SoS mission at runtime.
Proposition 4	Mission engineering substantially influences the analysis of technical interoperability requirements and both contribute to SoS interoperation at design time and runtime.
Proposition 5	Each requirement must be defined in measurable terms, contain applicable criteria, and be sufficiently detailed to be used as design criteria and architectural synthesis activity.
Proposition 6	Enterprise and technical alignment issues emerge as consequence of the inability of their interaction for comprehending the technical environment and stakeholders' needs and experiences of aware communication, and domain terminology can influence improvements in this relation.
Proposition 7	Technical interoperability awareness issues support communication and collaboration between CSs, maintaining technical sustainability and resources available for decisions over time.
Proposition 8	The greater the awareness of collaboration for technical interoperability, the easier the analysis of technical interoperability requirements.

Table 1 (cf. Chapter 1) shows each specific objective and the section(s) that correspond to the findings that guided the achievement of the aim. For instance, objective (i) was addressed in Section 4.2.6, whereas (iv) can be considered crosscutting objective being addressed in more than one section, i.e., Sections 4.2.3, 4.2.7, 4.2.8, 4.2.9, and 5.5. Building of theory was not only grounded through the interviewees statements, as they analyzed the requirements, but on the understanding of their reality in both comprehensive and peculiar ways towards a generic theory that would suit most of them. Besides internal analysis, which checked the relationship between our aim and the proposition, the theory validity was also analyzed by triangulation. The following three types of triangulation were applied to our process: (i) data (technical literature, observations, non-technical literature, discussions with experts, memos, and codes), which concerns time, space and persons involved; (ii) methodology (51 survey respondents, 27 interviews, and four systematic mappings), which refers to the use of multiple methods, and (iii) environment (14 countries), which concerns the use of different realities that involve environmental factors, such

Table 1 – The relation between specific objectives and sections

ID	Objective	Section
(i)	Investigate and map the relationship between mission, requirements, and architecture for the SoS field	(cf. Section 4.2.6)
(ii)	Investigate and describe the way technical interoperability requirements are specified and analyzed for the SoS field	(cf. Sections 4.2.1, 4.2.4, and 5.4.2.3)
(iii)	Identify and describe the relationship between mission and requirements, and the way requirements are specified and analyzed by mission engineering	(cf. Sections 4.2.10 and 5.4
(iv)	Provide a comprehensive explanation of challenges and design of issues to support the analysis of technical interoperability requirements	(cf. Sections 4.2.3, 4.2.7, 4.2.8, 4.2.9, and 5.5)
(v)	Analyze and describe the barriers and sustainability perspectives for technical interoperability requirements based on the industry	(cf. Sections 4.2.9, and 4.2.5)

as domain, size, and economy. The identification of empirical data provided adequate ones for analysis, which, together with technical and non-technical literature led to the findings reported in the previous chapter. Furthermore, all steps of the theory were evaluated by 10 interviewees through constant feedback, and by experts on the areas.

The main conclusion of this research is that the ATLANTA Theoretical Framework must be considered for each enterprise and application domain, since a generic model was created. Moreover, the theory that deals with technical interoperability awareness was based on blended SoS, composed of an individual system that has managerial and operational controls, and others that only have an operational control, but whose management is subordinate to a central control. Regarding RE and architectural level, requirements are treated in different forms for each SoS category. Therefore, the type must also be considered when the analysis of technical interoperability requirements driven by mission engineering is planned in MisConOps or MisCon. Stakeholders can support both mission and SoS team; however, according to them, the languages are usually complex. We, therefore, suggest the use of multiple views for an easier communication process between business stakeholders and technical team, and the start of decision-making and the rationale in the early stages of an enterprise analysis. Such activities satisfy the stakeholders' needs and support the traceability process from mission to architecture, and from the operational environment to the mission. Such conclusions are in agreement with the literature since, although technical interoperability requirements analysis, is not related in it, there are descriptions of challenges of SoS about analysis, as shown in Jamshidi (2008) and Crowder *et al.* (2016). Findings are aligned with the diffusion theory, which highlights a novelty to be accepted, a relevant benefit must be compared to current procedures (ROGERS, 1983).

Thereby, the theory shows that ATLANTA directs the whole environment to deal with technical interoperability awareness, impacting even the business model to know awareness changes and interoperation of individual systems when it comes to the technical part of the standards, protocols, communication networks, mediators, interfaces, and others. The application of grounded theory was relevant to this context by allowing interaction between various complementary disciplines of SoS development, by expanding this process to enterprise analysis, incorporating methods of strategic analysis to SoS, and by addressing an activity of analysis driven by mission engineering, which is not still known, especially in terms of its threats to the development lifecycle and consequences. Such substantive theory supports the mission and SoS team and stakeholders to define strategies, evaluate the quality attributes related to technical interoperability requirements, verify barriers, and others.

6.4 Threats to Validity

The findings of the research were proposed to build a substantive theory that specifies a particular context in a specific area and it must be able to satisfy quality criteria to achieve the scientific severity of an exploratory-descriptive and qualitative study, of applied and non-applied nature. The criteria were described in Section 3.8. In this section, we discuss issues of validity and reliability of this research based on the works of Charmaz (2005) Wohlin *et al.* (2012) and Merriam and Tisdell (2015). The validity of quality researches has modified their concepts to better match the methods that are being adopted as describes Merriam and Tisdell (2015). However, in general, validity and reliability are related for rigor of the process that is conducted until a result that provides trustworthiness for users is achieved. We also maintain the ATLANTA aligned with the diffusion theory (ROGERS, 1983) addressed in Section 6.3.

6.4.1 Credibility or Internal Validity

Credibility is concerned with the relationship between substantive theory and empirical analysis or how the research findings match reality, since qualitative research comprehends that is even changing is built of a multidimensional way. Therefore, a given reality is interpreted as not measured. Guaranteeing the credibility (**or internal validity**) of the research means that findings are valid to the reality of the quality paradigm adopted. Such guarantee is achieved by means of triagulation (cf. Section 6.3) and methodological process rigor (cf. Chapter 3). All categories emerged from the data as discussed in Chapter 3. A range of empirical data was raised through the survey and interviews that were coded. Open and color codes were used as well as axial and selected codes to generate the category, sub-categories, and the codes are displayed in Appendix M. Relationships between categories generated a generic model using theoretical codes and their relation to the core category and the basis of technical interoperability requirements identified in empirical data and compared to literature. Comparison with the literature was performed based on systematic mapping to each correlated area and non-technical literature and findings are shown in Chapters 4 and 5.

6.4.2 Originality, Reliability, Consistency, or Construct Validity

Originality refers to verifying whether the categories established provide the possibility of new ideas in the research area, i.e., the findings can be replicated many times if the phenomenon is considered stable and reliable. This research claims to be of theoretical expressiveness; as the SoS poses great challenges, because there are not specifics ways of dealing with technical interoperability requirements, we defined such perspectives and linked enterprise to technology at runtime by adopting mission engineering to guide the technical interoperability requirement analysis process (cf. Section 4.2.5 Section 4.2.9, and Section 4.2.10; Chapter 5). Such requirements are based on the interoperation process and failure of any technical interoperability requirements that can impair mission satisfaction and hinder the achievement of mission outcomes as even business objectives. The qualitative research understands the reality of a viewpoint. Although there are many viewpoints, originality is possible in a specific case; however, the replication may generate different results. Quality research could ensure that results be consistent with collected data.

6.4.3 Resonance or Conclusion Validity

Our substantive theory was designed using empirical data from practitioners and academics involved in the viewpoint of the industrial project. The **resonance** regarded assessing the relation of the categories with the experience of respondents, and whether such a grounded category makes sense to interviewees. The findings of the categories were designed with theoretical sensibility based on meaning that the interviewees provide to the questions that they answer. We gathered data in an iterative and incremental way, and to each one, we sought at least two pieces of feedback. For the final of category, it was always necessary to consult 10 interviewees to clear doubts and make the analysis process closest to them. One exception of the feedback was displayed in Section 3.6.

6.4.4 Usefulness or External Validity

Lastly, we addressed the **usefulness** that aims to verify the applicability to an individual's daily life and it is possible to transfer it to other research areas. As described in Section 5.5.1 and Chapter 5, the theory can be used in other areas and also in day-to-day business work, since it has information that supports the identification of SoS mission, which can be conducted by the mission, requirements or architect engineer. An interoperability engineer or SoS architect can use specification of technical interoperability requirements at both, design time and runtime, identification of main technical barriers or sustainable measure can use for decision-making, and the owner Triplet model to design the SoS. However, guidelines were developed with transformation properties to be possible to apply in SoS with central control. To the SoS without control, it would be necessary to develop testing to verify the applicability, especially if it is the virtual type.

6.5 Limitations and Future Works

This section describes the limitations of this research and how these can be addressed in future research.

Overcome theory limitations: Despite the substantive theory provides knowledge to understand and make the analysis of technical interoperability requirements through the ATLANTA Theoretical Framework easier, some challenges still remain and must be considered in requirements engineering and architectural analysis. We focused on the analysis, but in both levels, there are essential activities that need to be performed to increase the quality of technical interoperability requirements. Our research shows (cf. (LANA *et al.*, 2018)) that formality supports to increased the quality, but that many industries often have used one combination between formal and semi-formal languages, especially in early stages of SoS development. In addition, semi-formal to structural specification facilitates the communication and the comprehension of stakeholders in the development phase. For this, ontologies are essential to improve the ambiguity of artifacts generated to all the levels and models, facilitating design level, and the mapping among standards and protocols.

Evaluate quality attributes:

We detailed the $QM2_{TI}$ in Chapter 5 and, although it has been elaborated with empirical data and evaluated by the interviewees and experts, providing reliability and attending the interviewees of the various domains. In future work, we plan to perform an external assessment with other industry practitioners and academics to refine and consolidate each description and its current organization.

Evaluate the theory with a case study:

The theory reported in this Thesis was designed based on empirical data and literature-driven analysis together with experts discussion presented in Chapter 4 and Chapter 5. In future work, we plan to investigate a case study to assess the application in a real environment and perform the refinements or extensions necessary to a specific domain. In this case study, we intend to address a blended SoS to evaluated the Mission Specification initially and the enterprise analysis; extending to the complete ATLANTA Theoretical Framework.

Automate the processes of the ATLANTA :

The theory presented in this Thesis was developed in a semi-manual way. In future work, we intend to automate the processes to make them easier to execute, considering they are complex and the artifacts generated over time will be many. Such a procedure in the manual way is impracticable and makes it difficult to achieve an important requirement, backward and forward traceability.

6.6 Possible Extensions

Many were the opportunities for research to emerge during the development of this Thesis, and they represent possible extensions that can contribute to the areas of SoS, ER, mission engineering, and architectural analysis. Four of these are described as follows:

Evolve the theory to other activities of requirements engineering and architectural analysis: We established the theory specifically considering the analysis activity. However, to ensure quality to a critical system all the activities of the RE process and architectural analysis must be performed. Therefore, such investigation allows the extension of the theory to each activity, which can be then grouped in one RE process for SoS at both levels, requirements engineering and RE architecture. The specific process of RE architecture will assist the SoS team of SoSE, which many times start the requirements gathering at the architectural level.

Interoperability Language for modeling mission engineering: In Chapter 4.2.1 describes the incapacity of the "current languages to perform the necessary analysis of interoperability requirements achieving the interactions and comparison between models and view to multilevel of development, making traceability difficult". Interoperability and mission engineering have particularities that, according to interviewees make current languages end up being incomplete to serve them. Therefore, similar system areas approaches can be used to enable comparison analysis to adjust the specific necessities to a language that could meet the necessities of interoperability in a general way, and of mission engineering in its specification. Such investigation is required to allow the language to be used by the enterprise at runtime changes.

Improve transformation properties and constraints specification: In Chapter 5, we define various transformation properties and constraints rules to specify from mission to requirements using the mission engineering transformation. This rule has a mathematical formality that can be optimized for each set of specification aiming to improve the quality of application of ATLANTA in an automated fashion.

Ontology for technical interoperability requirements: In Chapter 4.2.10, we described the necessity of design-decision and rationale starting in initial stages to improve the achievement of technical interoperability requirements, and to reduce the misunderstanding of information. Another way to improve the quality of information is to reduce its ambiguity over time. Therefore, designing an ontology for technical interoperability requirements driven by mission engineering will enable the consolidation of the specification of ATLANTA and to language, supporting to specify and analyze further SoS in several domains.

Evolve multilevel mission management with machine learning: In $ATLANTA_{M3}$, we created a model of management of mission at runtime. Machine learning and control can improve the process of technical interoperability requirements at runtime, keeping the mission always active, for example, by monitoring failure, or checking emergent behavior or CSs evolution. Therefore, future research should focus on mechanisms to automatically perform the $ATLANTA_{M3}$ ensuring the reliability of the process and the SoS mission should always be active because of technical interoperability requirements.

References

ABBOTT, R. Open at the top; open at the bottom; and continually (but slowly) evolving. In: **IEEE/SMC International Conference on System of Systems Engineering**. Los Angeles, California: IEEE, 2006. p. 41–46. Citation on page 16.

ABDALLA, G. **Establishment of an Ontology for Systems-of-Systems**. Master's Thesis (Master) — Institute of Mathematics and Computational Sciences, University of São Paulo, São Carlos, 2017. Citation on page 49.

ABDALLA, G.; DAMASCENO, C. D. N.; NAKAGAWA, E. Y. **A systematic literature review on systems-of-systems knowledge representation**. 2015. Citation on page 32.

ABUHAV, Y. **ISO 9001:2015: A Complete Guide to Quality Management Systems**. : CRC Press, 2017. Citation on page 139.

ACHESON, P. Methodology for object-oriented system architecture development. In: **4th Annual IEEE international Systems Conference (SysCon 2010)**. San Diego, California: IEEE, 2010. p. 643–646. Citation on page 35.

ACHESON, P.; DAGLI, C.; KILICAY-ERGIN, N. Model based systems engineering for system of systems using agent-based modeling. In: **CSER**. 2013. p. 11–19. Citation on page 212.

ACKOFF, R. L. Towards a system of systems concepts. **Management Science**, INFORMS, v. 17, n. 11, p. 661–672, 1971. Citation on page 14.

ADOLPH, S.; HALL, W.; KRUCHTEN, P. Using grounded theory to study the experience of software development. **Empirical Software Engineering**, Kluwer Academic Publishers, v. 16, n. 4, p. 487–513, 2011. Citations on pages 44 and 50.

AFFONSO, F. J.; LEITE, G.; OLIVEIRA, R. A. P.; NAKAGAWA, E. Y. A framework based on learning techniques for decision-making in self-adaptive software. In: **Software Engineering and Knowledge Engineering (SEKE 2015)**. Pittsburgh, USA: http://ksiresearchorg.ipage.com/, 2015. p. 1–6. Citation on page 144.

AHMED, R.; ROBINSON, S. Simulation in business and industry: How simulation context can affect simulation practice? In: **Spring Simulation Multiconference (SpringSim 2007)**. Norfolk, Virginia: Society for Computer Simulation International, 2007. p. 152–159. Citation on page 27.

AL-NASHIF, Y.; MYTHILI, S. V. **Autonomia : Autonomic Control and Management Environment**. 2007. http://acl.ece.arizona.edu/projects/old/Self-Healing/index.html. [Online, Acessed: september 13, 2019]. Available: <http://acl.ece.arizona.edu/projects/old/Self-Healing/index.html>. Citation on page 76.

ALBERS, A.; PEGLOW, N.; POWELSKE, J.; BIRK, C.; BUSAC, N. Coping with complex systems-of-systems in the context of pge – product generation engineering. In: **Proceedings of the 28th CIRP Design Conference**. Nates, France: Elsevier, 2018. p. 457–462. Citations on pages 16, 17, and 212.

ALON, N.; YUSTER, R.; ZWICK, U. a. Color-coding. **Jornal of ACM**, ACM, v. 42, n. 4, p. 844–856, 1995. Citation on page 47.

ALSHAZLY, A. A.; ELFATATRY, A. M.; ABOUGABAL, M. S. Detecting defects in software requirements specification. **Alexandria Engineering Journal**, Elsevier B.V., v. 53, n. 3, p. 513–527, 2014. Citation on page 13.

AMANOWICZ, M.; GAJEWSKI, P. Military communications and information systems interoperability. In: **IEEE Military Communications Conferece (MILCM 1996)**. McLean, USA: IEEE, 1996. p. 280–283. Citation on page 37.

AMYOT, D.; GHANAVATI, S.; HORKOFF, J.; MUSSBACHER, G.; PEYTON, L.; YU, E. Evaluating goal models within the goal-oriented requirement language. **International Journal of Intelligent Systems**, John Wiley & Sons, Inc., v. 25, n. 8, p. 841–877, 2010. Citation on page 34.

ANTHONY, R. J. Systems programming: Designing and developing distributed applications. In: ANTHONY, R. J. (Ed.). University of Greenwich, UK: Elsevier, 2016. chap. Case Studies: Putting it All Together, p. 138–157. Citation on page 33.

ARTS, T.; HUGHES, J.; NORELL, U.; SVENSSON, H. Testing autosar software with quickcheck. In: **IEEE 8th International Conference on Software Testing, Verification and Validation Workshops (ICSTW 2015)**. Gaz, USA: IEEE, 2015. p. 1–4. Citation on page 26.

ATKINSON, W.; CUNNINGHAM, J. Proving properties of a safety-critical system. **Software Engineering Journal**, IET, v. 6, n. 2, p. 41–50, 1991. Citation on page 28.

ATSB. **The Operational Search for MH370**. 2017. Citation on page 21.

AVIZIENIS, A.; LAPRIE, J.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. **IEEE Transactions Dependable Security Compututer**, IEEE Computer Society Press, v. 1, n. 1, p. 11–33, Jan. 2004. Citation on page 16.

AXELSSON, J. A systematic mapping of the research literature on system-of-systems engineering. In: **10th System of Systems Engineering Conference (SoSE 2015)**. San Antonio, USA: IEEE, 2015. p. 18–23. Citation on page 26.

AXELSSON, J. **Systems-of-systems for border-crossing innovation in the digitized society: a strategic research and innovation agenda for Sweden**. 2015. Citations on pages 16, 17, and 26.

AYALA, G. Supervision for system of systems engineering based on fuzzy logic approach. In: **IEEE International Conference on Robotics and Biomimetics (ROBIO 2015)**. Zhuhai, China: IEEE, 2015. p. 2591–2596. Citations on pages 22, 125, and 126.

BAIMYRZAEVA, M. **Beginners' Guide for Applied Research Process: What Is It, and Why and How to Do It?** 2018. Citation on page 44.

BALDWIN, K. **System engineering guide for systems-of-systems engineering**. 2008. Citations on pages 132, 134, and 193.

BALLAGNY, C.; HAMEURLAIN, N.; BARBIER, F. MOCAS: A state-based component model for self-adaptation. In: **SASO**. 2009. p. 206–215. Citation on page 212.

BARKER, S. G.; SUMMERS, M. P. Robust decision making for agile systems development part 1: Exploring the paradigm. In: **9th International Conference on Axiomatic Design (ICAD 2015)**. Wiltshire, United Kingdom: Elsevier, 2015. p. 44–49. Citation on page 212.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. : Addison-Wesley, 2012. Citations on pages 3, 14, 34, 62, and 151.

BEALE, D.; HARRIS, D.; BONOMETTI, J.; MUELLER, R.; MURPHY, G. Chapter 2: Systems engineering (se) – the systems design process. In: BEALE, D.; HARRIS, D.; BONOMETTI, J.; MUELLER, R.; MURPHY, G. (Ed.). : Auburn University Office of Information Technology, 2004. chap. A Rrich services approach to CoCoME, p. http://www.eng.auburn.edu/ dbeale/ES-MDCourse/Chapter2.htm. Citations on pages 2 and 194.

BENALI, H.; SAOUD, N. B. B.; AHMED, M. B. Context-based ontology to describe system-of-systems interoperability. In: **11th International Conference on Computer Systems and Applications AICCSA 2014**. Doha, Qatar: IEEE, 2014. p. 64–71. Citation on page 49.

BENBASAT, I.; GOLDSTEIN, D. K.; MEAD, M. The case research strategy in studies of information systems. **MIS Quarterly**, Society for Information Management and The Management Information Systems Research Center, v. 11, n. 3, p. 369–386, 1987. Citation on page 44.

BENCOMO, N.; WHITTLE, J.; SAWYER, P.; FINKELSTEIN, A.; LETIER, E. Requirements reflection: Requirements as runtime entities. In: **Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 2 (ICSE 2010)**. New York, USA: ACM, 2010. p. 199–202. Citation on page 144.

BENDOV, N. **Designing for adaptability and evolution in system of systems engineering: Characterization of SoS D 4.1**. 2009. Citation on page 31.

BHASIN, K. B.; WARNER, J. D.; ANDERSON, L. M. Lunar communication terminals for nasa exploration missions: Needs, operations concepts and architectures. In: **26th International communication Satellite Systems Conference (ICSSC 2008) American Institute of Aeronautics and Astronautics (AIAA 2008)**. San Diego, CA: American Institute of Aeronautics and Astronautics, 2008. p. 1–24. Citation on page 54.

BIANCHI, T.; SANTOS, D. S.; FELIZARDO, K. R. Quality attributes of systems-of-systems: A systematic literature review. In: **International Workshop on Software Engineering for Systems-of-Systems (SESoS 2015)**. Florence, Italy: , 2015. p. 23–30. Citation on page 151.

BICKMAN, L.; ROG, D. J. E. **The Sage Handbook of Applied Research Methods**. : SAGE Publications, Inc., 2009. Citation on page 44.

BILA, H. A.; ILYAS, M.; TARIQ, Q.; HUMMAYUN, M. Requirements validation techniques: An empirical study. **International Journal of Computer Applications**, Foundation of Computer Science, v. 148, n. 14, p. 5–10, 2016. Citations on pages 3 and 27.

BILLAUD, S.; DACLIN, N.; CHAPURLAT, V. Enterprise interoperability. In: SINDEREN, M. van; CHAPURLAT, V. E. (Ed.). Berlin: Springer-Verlag Berlin Heidelberg, 2015. chap. Interoperability as a Key Concept for the Control and Evolution of the System of Systems (SoS), p. 53–63. Citations on pages 38 and 67.

BILTGEN, P. T. **A Methodology for Capability-based Technology Evaluation for Systems-of-Systems**. Master's Thesis (Thesis) — School of Aerospace Engineering Georgia Institute of Technology, 2007. Citation on page 212.

BOARDMAN, J.; SAUSER, B. System of systems – the meaning of of. In: **IEEE/SMC International Conference on System of Systems Engineering**. Los Angeles, USA: IEEE, 2006. p. 118–123. Citations on pages 15, 16, and 201.

BOEHM, B.; LANE, J. 21 st century processes for acquiring 21 st century software-intensive systems of systems. **The Journal of Defense Software Engineering**, v. 19, n. 5, p. 4–9, 2006. Citation on page 212.

BOEHM, B. W. **Software engineering economics**. Upper Saddle River, USA: Prentice Hall PTR, 1981. Citations on pages 26 and 33.

BOEHM, B. W. Verifying and validating software requirements and design specifications. **IEEE Software**, IEEE Computer Society Press, v. 1, n. 1, p. 75–88, 1984. Citations on pages 3 and 27.

BOER, R. C.; VLIET, H. On the similarity between requirements and architecture. **Journal of Systems and Software**, Science Direct, v. 82, n. 3, p. 544–550, 2009. Citation on page 92.

BOULDING, K. E. General systems theory: The skeleton of science. **Management Science**, INFORMS, v. 2, n. 3, p. 197–208, 1956. Citation on page 14.

Bradbury, J. S. **Organizing Definitions and Formalisms for Dynamic Software Architectures**. 2004. Citation on page 76.

BRAGA, R. T. V.; BRANCO, K. R. C.; JR., O. T.; MASIERO, P. C.; NERIS, L. O.; BECKER, M. The ProLiCES approach to develop product lines for safety-critical embedded systems and its application to the unmanned aerial vehicles domain. **Clei Eletronic Journal**, v. 15, n. 8, p. 1–13, 2012. Citation on page 212.

BRERETON, P.; KITCHENHAM, B.; BUDGEN, D.; LI, Z. Using a protocol template for case study planning. In: **12th International Conference on Evaluation and Assessment in Software Engineering (EASE 2008)**. Swindon, UK: BCS Learning & Development Ltd., 2008. p. 41–48. Citation on page 46.

BUTLER, G.; WOODY, C. **Mission Threads: Bridging Mission and Systems Engineering**. 2017. Citation on page 128.

BUTTERFIELD, M.; PEARLMAN, J.; VICKROY, S. A system-of-systems engineering geoss: architectural approach. **IEEE Systems Journal**, v. 2, n. 3, p. 321–332, 2008. Citation on page 212.

CALINESCU, R.; KWIATKOWSKA, M. Software engineering techniques for the development of systems of systems. In: CHOPPY, C.; SOKOLSKY, O. (Ed.). **15th Monterey Workshop on Foundations of Computer Software**. Vanderbilt, USA: Springer Verlag, 2008. p. 59–82. Citation on page 207.

CALINESCU, R.; KWIATKOWSKA, M. Software engineering techniques for the development of systems of systems. In: **Monterey**. 2008. p. 59–82. Citation on page 212.

CAO, J.; MAO, X.; YAN, H.; HUANG, Y.; WANG, H.; LU, X. Capability as requirement metaphor. In: **IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)**. Changsha, China: IEEE, 2011. Citation on page 120.

CARBON, R.; JOHANN, G.; MUTHIG, D.; NAAB, M. A method for collaborative development of systems of systems in the office domain. In: **12th International IEEE Enterprise Distributed Object Computing Conference (ECOC 2008)**. Munich, Germany: IEEE, 2008. p. 339–345. Citation on page 207.

CARBON, R.; JOHANN, G.; MUTHIG, D.; NAAB, M. A method for collaborative development of systems of systems in the office domain. In: **ECOC**. 2008. p. 339–345. Citation on page 212.

CARLOCK, P.; LANE, J. A. System of systems enterprise systems engineering, the enterprise architecture management framework, and system of systems cost estimation. In: **21st International Forum on Systems, Software, and COCOMO Cost Modeling**. Herndon, VA: University Southern California, 2006. p. 1–12. Citation on page 23.

CARVALHO, L.; SCOTT, L.; JEFFEY, R. **Exploring the Use of Techniques from Grounded Theory in Process Engineering**. 2003. Citation on page 43.

CAVALCANTE, E.; OQUENDO, F.; BATISTA, T. Architecture-based code generation: From π-adl architecture descriptions to implementations in the go language. In: **8th European Conference Software Architecture (ECSA 2014)**. Vienna, Austria: Springer International Publishing, 2014. p. 130–145. Citation on page 35.

CAVALCANTE, E. R. S. **A Formally Founded Framework for Dynamic Software Architectures**. Master's Thesis (PhD Thesis) — Department of Computers and Applied Mathematics - University of Rio Grande do Norte, Natal, Brazil, 2016. Citations on pages 35 and 36.

CERVANTS, H.; KAZMAN, R. **Designing Software Architectures: A Practical Approach**. : Pearson Education, Inc., 2016. Citation on page 92.

CHAKRABORTY, A.; BAOWALY, M. K.; AREFIN, U. A.; BAHAR, A. N. The role of requirements engineering in software development life cycle. **Journal of Emerging Trend in Computing and Information Sciences**, Université du Québec, v. 3, n. 5, p. 723–729, 2012. Citation on page 13.

CHAKRABORTY, S.; DEHLINGER, J. Applying the grounded theory method to derive enterprise system requirements. In: **10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing**. Toyama, Japan: IEEE Computer Society, 2009. p. 333–338. Citation on page 43.

CHAPURLAT, V.; KAMSU-FOGUEM, B.; PRUNET, F. A formal verification framework and associated tools for enterprise modeling: Application to ueml. **Computers in Industry**, Elsevier Science Publishers B. V., v. 57, n. 2, p. 153–166, 2006. Citation on page 26.

CHARMAZ, K. **Grounded Theory in the 21st Century: Applications for Advancing Social Justice Studies**. : Sage Publications Ltd, 2005. Citations on pages 7, 44, 46, 47, 48, 51, 52, 59, 64, 110, 119, and 157.

CHARMAZ, K. **Constructing Grounded Theory**. 2. ed. : Sage Publications Ltd, 2014. Citations on pages 44, 49, 51, and 60.

CHEN, D.; DACLIN, N. Interoperability for enterprise software and application. In: PANNETO, H.; BOUDJLIDA, N. (Ed.). : Willey Online Library, 2006. chap. Framework for Enterprise Interoperability, p. 77–88. Citations on pages 41, 147, and 152.

CHEN, D.; VERNADAT, F. B. Enterprise inter- and intra-organizational integration. In: KOSANKE, K.; JOCHEM, R.; NELL, J. G.; BAS, A. O. (Ed.). : Springer Link, 2002. chap. Enterprise Interoperability: A Standardisation View, p. 273–282. Citations on pages 36 and 37.

CHEN, L.; ALIBABAR, M.; NUSEIBEH, B. Characterizing architecturally significant requirements. **IEEE Software**, IEEE Computer Society Press, v. 30, n. 2, p. 38–45, 2013. Citations on pages 33, 89, 92, and 148.

CHENG, B. H. C.; ATLEE, J. M. Research directions in requirements engineering. In: **Future of Software Engineering (FOSE 2007)**. Washington, USA: IEEE Computer Society, 2007. p. 285–03. Citations on pages 13 and 43.

CHENG, B. H. C.; EDER, K. I.; GOGOLLA, M.; GRUNSKE, L.; LITOIU, M. L.; MULLER, H. A.; PELLICCIONE, P.; PERINI, A.; QURESHI, N. A.; RUMPE, B.; SCHNEIDER, D.; TROLLMANN, F.; VILLEGAS, N. M. Modelsrun.time. In: BENCOMO, N.; FRANCE, R.; CHENG, B. H. C.; AßMANN, U. (Ed.). : SpringerLink, 2014. chap. Using Models at Runtime to Address Assurance for Self-Adaptive Systems, p. 101–136. Citation on page 144.

CHIGANI, A.; BALCI, O. The process of architecting for software/system engineering. **International Journal of System of Systems Engineering**, Department of Computer Science, VirginiaTech, v. 3, n. 1, p. 1–23, 2012. Citation on page 35.

CHRISTEL, M.; KANG, K. **Issues in requirements elicitation**. 1992. Citation on page 98.

CHUNG, K. S. K.; CRAWFORD, L. The role of social networks theory and methodology for project stakeholder management. In: **29th World Congress International Project Management Association (IPMA 2015)**. Westin Playa Bonita, Panama: Science Direct, 2016. p. 372==380. Citation on page 76.

CIMATTI, A.; ROVERI, M.; SUSI, A.; TONETTA, S. Formalizing requirements with object models and temporal constraints. **Software and Systems Modeling**, Springer-Verlag New York, Inc., v. 10, n. 2, p. 147–160, 2011. Citation on page 30.

CLARK, T.; JONES, R. Organisational interoperability maturity model for c2. In: **Command and Control Research and Technology Symposium (C2RTS 1999)**. Rhode Island, USA: Software Engineering Institute - Carnegie Mellon University, 1999. p. 1–13. Citation on page 38.

CLELAND-HUANG, J.; HANMER, R. S.; SUPAKKUL, S.; MIRAKHORLI, M. The twin peaks of requirements and architecture. **IEEE Computer Society**, IEEE Software, v. 30, n. 2, p. 24–29, 2013. Citations on pages 33, 89, and 90.

CLEMENTS, P.; BACHMANN, F.; BASS, L. **Documenting Software Architectures: Views and Beyond**. 2. ed. : Addison-Wesley, 2011. Citations on pages 34 and 36.

COFER, D.; MILLER, S. P.; COLLINS, R.; IOWA, C. R. **Formal methods case studies for DO-333**. 2014. Citations on pages 32 and 36.

COLEMAN, J.; O'CONNOR, M. Using grounded theory to understand software process improvement: A study of irish software product companies. **Information and Software Technology**, Science Direct, v. 46, n. 6, p. 654–667, 2007. Citation on page 44.

COMMUNITIES, E. **European Interoperability Framework for Pan-European eGovernment Services**. 2004. Citations on pages 36 and 37.

CONGRESS, S. C. o. A. S. U.; SERVICES, H. C. on A. **Restructuring of the Strategic Defense Initiative (SDI) Program**. : University of Michigan Library, 1989. Citation on page 14.

CORBIN, J.; STRAUSS, A. **Basics of qualitative research: Techniques and procedures for developing grounded theory**. 3. ed. : Sage Publications, Inc, 2008. Citations on pages 47, 60, 64, and 120.

CORDES, D. W.; CARVER, D. L. Generating a requirements specifications knowledge-base. In: **ACM Sixteenth Annual Conference on Computer Science (CSC 1988)**. Atlanta, Georgia, USA: ACM, 1988. p. 727–. Citation on page 32.

CROW, J.; VITO, B. D. Formalizing space shuttle software requirements: Four case studies. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, ACM, v. 7, n. 3, p. 296–332, 1998. Citation on page 30.

CROWDER, J. A.; CARBONE, J. N.; DEMIJOHN, R. **Multidisciplinary Systems Engineering: Architecting the Design Process**. : Springer International Publishing Switzerland, 2016. Citations on pages 22, 34, 35, 130, and 156.

DACLIN, N.; DACLIN, M. S.; VALLESPIR, C. B. Writing and verifying interoperability requirements: Application to collaborative processes. **Computers in Industry**, Elsevier, v. 82, n. C, p. 1–18, 2016. Citation on page 67.

DAHMANN, J. **Systems of Systems Characterization and Types**. 2015. Citations on pages 72 and 74.

DAHMANN, J. **Mission Engineering: Systems of Systems Engineering in Context**. 2019. Citations on pages 4, 23, 24, 34, 41, and 100.

DAHMANN, J.; BALDWIN, K. Understanding the current state of us defense systems of systems and the implications for systems engineering. In: **2nd Annual IEEE International Systems Conference (SysCon 2008)**. Montreal, Canada: IEEE, 2008. p. 1–7. Citations on pages 1, 18, 24, and 75.

DAHMANN, J.; LANE, J.; REBOVICH, G.; LOWRY, R. Systems of systems test and evaluation challenges. In: **5th International Conference on System of Systems Engineering (SoSE 2010)**. Loughborough, UK: IEEE, 2010. p. 1–6. Citation on page 33.

DAHMANN, J.; MARKINA-KHUSID, A.; KAMANETSKY, J.; ANTUL, L.; JACOBS, R. **Systems of Systems Engineering Technical Approaches as Applied to Mission Engineering**. 2017. Citations on pages 2 and 41.

DAHMANN, J.; REBOVICH, G.; LANE, J.; LOWRY, R.; BALDWIN, K. An implementers' view of systems engineering for systems of systems. In: **IEEE International Systems Conference (SysCon 2011)**. Montreal, Canada: IEEE, 2011. p. 212–217. Citation on page 212.

DAUN, M.; TENBERGEN, B.; WEYER, T. Model-based engineering of embedded systems. In: POHLHARALD, K.; HöNNINGERREINHOLD; BROY, A. (Ed.). : SpringerLink, 2012. chap. Requirements Viewpoint, p. 51–68. Citation on page 79.

DAZHI, X.; XIAOZHONG, W. A practical approach for phased mission analysis. **Reliability Engineering & System Safety**, ScienceDirect, v. 25, n. 1, p. 333–347, 1988. Citations on pages 2, 19, and 20.

DECHER, P. Requirements driven development from contract win to customer sign-off. In: **15th Monterey Workshop on Foundations of Computer Software**. Manhattan Beach,USA: IEEE, 2010. p. 1–8. Citation on page 13.

DECHER, P. Requirements driven development from contract win to customer sign-off. In: **Monterey**. 2010. p. 1–8. Citation on page 212.

DELAURENTIS, D. A. A taxonomy-based perspective for systems of systems design methods. In: **IEEE International Conference on System, Man and Cybernetics (SMC 2005)**. Waikoloa, USA: IEEE, 2005. p. 86–91. Citation on page 13.

DERSIN, P.; TRANSPORT, A. **Systems of Systems**. 2014. http://rs.ieee.org/component/content/article/9/77-system-of-systems.html. [Online, Acessed: December 18, 2017]. Available: <http://rs.ieee.org/component/content/article/9/77-system-of-systems.html>. Citation on page 1.

DIALLO, S.; MITTAL, S.; TOLK, A. Emergent behanvior in complex systems engineering: A modeling and simulation approach. In: MITTAL, S. D. S.; (EDS.), A. T. (Ed.). : John Wiley & Sons, Inc, 2018. chap. Research Agenda for Next-Generation Complex System Engineering, p. 379–398. Citations on pages 22 and 36.

DIALLO, S. Y. **Towards a Formal Theory of Interoperability**. Master's Thesis (PhD Thesis) — Old Dominion University, Virginian, 2010. Citation on page 3.

DIAZ, J.; PéREZ, J.; PéREZ, J.; GARBAJOSA, J. Conceptualizing a framework for cyber-physical systems of systems development and deployment. In: **10th European Conference on Software Architecture Workshops (ECSAW 2016)**. Copenhagen, Denmark: ACM, 2016. p. 1–7. Citation on page 212.

DOD. **Business Capability Acquisition Cycle**. 2018. http://acqnotes.com/acqnote/acquisitions/business-capability-acquisition-cycle. [Online, Acessed: June 14, 2018]. Available: <http://acqnotes.com/acqnote/acquisitions/business-capability-acquisition-cycle>. Citation on page 20.

(DOD), D. o. D. **System engineering guide for systems-of-systems engineering**. 2008. Citation on page 3.

DOERR, J. **Measure What Matters:How Google, Bono, and The Gates Foundation Rock the World With OKRs**. : Portifolio, 2018. Citations on pages 105, 106, and 107.

DOLLING, J. Ontological domains, semantic sorts and systematic ambiguity. **International Journal of Human-Computer Studies**, Elsevier, v. 43, n. 5, p. 785–807, 1995. Citation on page 49.

DORAN, G. T. **There's a S.M.A.R.T. way to write management's goals and objectives**. 3. ed. Hawthorne, CA: Microcosm Press, 1981. Citation on page 20.

DRUCKER, P. **The Practice of Management**. : HarperCollins Publisher, 2010. Citation on page 105.

DYBå, T.; PRIKLADNICKI, R.; RöNKKö, K.; SEAMAN, C.; SILLITO, J. Qualitative research in software engineering. **Empirical Software Engineering**, IEEE, v. 16, n. 4, p. 425–429, 2011. Citation on page 43.

EASTERBROOK, S.; LUTZ, R.; COVINGTON, R.; KELLY, J.; AMO, Y.; D., H. **Experiences Using Formal Methods for Requirements Modeling**. 1998. Citations on pages 3 and 27.

EISNER, H. Rcasse: Rapid computer-aided system of systems (s2) engineering. In: **3dr International Council on Systems Engineering (INCOSE 1993)**. Chystal, Va: Wiley Online Library, 1993. p. 267–273. Citation on page 14.

EISNER, H.; MARCINIAK, J.; MCMILLAN, R. Computer-aided system of systems (c2) engineering. In: **IEEE International Conference on Systems, Man & Cybernetics**. Charlottesville, Va: IEEE, 1991. p. 1–12. Citation on page 14.

EN 50126. **The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)**. 1999. Citation on page 141.

ESARY, J. D.; ZIEHMS, H. **Reliability Analysis of Phased Missions**. 1975. Citations on pages 2, 19, and 20.

ESTAHANI, N.; ELKHODARY, A.; MALEK, S. A learning-based framework for engineering feature-oriented self-adaptative software systems. **IEEE Transactions on Software Engineering**, IEEE, v. 39, n. 11, p. 1467–1493, 2013. Citations on pages 144 and 148.

ESTEFANIA, L. D. E. **Space Mission Design and Operations**. 2010. Citations on pages 23 and 93.

ETZIEN, C. **Danse Modeling Formalism, including Domain Metamodel & Semantics: Focused on Support for Analysis and Optimization D_6.2.3**. 2014. Citation on page 40.

FABBRI, S.; HERNANDES, E. M.; THOMMAZO, A. D.; BELGAMO, A.; ZAMBONI, A.; SILVA, C. Managing literature reviews information through visualization. In: **14th International Conference on Enterprise Information Systems (ICEIS 2012)**. Wroclaw, Poland: Springer, 2012. p. 36–45. Citation on page 210.

FAISANDIER, A. **System Notion and Engineering of System: Engineering and Architecting Multidisciplinary Systems - Volume 1**. : Sinergy'com, 2015. Citations on pages 1, 105, and 194.

FARCAS, C.; FARCAS, E.; KRUEGER, I.; MENARINI, M. Addressing the integration challenge for avionics and automotive systems: from components to rich services. **Proceedings of the IEEE**, v. 98, n. 4, p. 562–583, 2010. Citation on page 212.

FARINHA, C.; SILVA, M. M. Requirements elicitation with focus groups: lessons learnt. In: **21st European Conference on Information Systems (ECIS 2013)**. Utrecht, Netherlands: Association for Information Systems, 2013. p. 1–12. Citation on page 97.

FARROHA, D.; FARROHA, B. Agile development for system of systems: cyber security integration into information repositories architecture. In: **IEEE International Systems Conference (SysCon 2011)**. Montreal, Canada: IEEE, 2011. p. 182–188. Citations on pages 35 and 207.

FARROHA, D.; FARROHA, B. Agile development for system of systems: cyber security integration into information repositories architecture. In: **SysCon**. 2011. p. 182–188. Citation on page 212.

FIRESMITH, D. **Profiling Systems Using the Defining Characteristics of Systems of Systems (SoS)**. 2010. Citation on page 16.

FISHER, D. A. **An Emergent Perspective on Interoperation in Systems of Systems**. 2006. Citation on page 3.

FORD, C. T.; COLOMBI, M. J.; GRAHAM, S. R.; JACQUES, R. D. A survey on interoperability measurement. In: **12th International Command and Control Research and Technology Symposium**. Washington, DC: Command and Control and Cyber Research Portal, 2007. p. 1–28. Citations on pages 36, 37, 67, and 151.

FORD, T.; COLOMBI, J.; GRAHAM, S.; JACQUES, D. The interoperability score. In: **5th Annual Conference on System Engineering Research (CSER 2007)**. Hoboken, USA: Crown Copyright, 2007. p. 1–10. Citations on pages 37 and 38.

FREITAS, A. S. **A Implementação do E-learning nas Escolas de Gestão: Um Modelo Integrado para o Processo de Alinhamento Ambiental**. Master's Thesis (Thesis) — Department of Administration, Pontíficia Universidade Católica do Rio de Janeiro (PUC-RIO), 2009. Citation on page 60.

FRENG, P. B. On the nature of sytems of systems. In: **26th Annual INCOSE International Symposium (IS 2016)**. Scotland, UK: INCOSE, 2016. p. 1–17. Citation on page 15.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A Practical Guide to SysML: the systems modeling language**. : Elsevier, 2008. Citation on page 27.

GALSTER, M.; MIRAKHORLI, M.; CLELAND-HUANG, J.; FRANCH, X.; BURGE, J. E.; ROSHANDEL, R.; AVGERIOU, P. Towards bridging the twin peaks of requirements and architecture. **ACM SIGSOFT Software Engineering Notes**, ACM, v. 39, n. 5, p. 30–31, 2014. Citations on pages 33 and 89.

GARCÉS, a.; NAKAGAWA, E. Y. A process to establish, model and validate missions of systems-of-systems in reference architectures. In: **Symposium on Applied Computing (SAC 2017)**. : ACM, 2017. p. 1–8. Citations on pages 40 and 54.

GASSER, U.; PALFREY, J. **When and How ICT Interoperability Drives Innovation**. 2007. Citation on page 114.

GEORGIADIS, I.; MAGEE, J.; KRAMER, J. Self-organising software architectures for distributed systems. In: **Proceedings of the First Workshop on Self-healing Systems (WOSS 2002)**. New York, USA: ACM, 2002. p. 33–38. Citation on page 145.

GHAZEL, M.; KOURSI, E. M. el. Automatic level crossings: From informal functional requirements' specifications to the control model design. In: **IEEE International Conference on System of Systems Engineering (SOSE 2007)**. San Antonio, USA: IEEE, 2007. p. 1–6. Citation on page 30.

GHEZZI, C.; JAZAYERI, M.; MANDRIOLI, D. **Fundamentals of Software Engeneering**. : Print-Hall of India, 2002. Citations on pages 28, 30, and 32.

GIDEY, H. K.; MARMSOLER, D.; ECKHARDT, J. Grounded architectures: Using grounded theory for the design of software architectures. In: **IEEE International Conference on Software Architecture Workshops (ICSAW 2017)**. Gothenburg, Sweden: IEEE, 2017. p. 141–148. Citation on page 43.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. : São Paulo: Atlas, 2010. Citation on page 44.

GLASER, B. G.; STRAUSS, A. L. **The Discovery of Grounded Theory: Strategies for Qualitative Research**. : Aldine Publishing Company, 1967. Citations on pages 43, 44, 46, 48, 49, 62, and 63.

GOKHALE, A.; BALASUBRAMANIAN, K.; KRISHNA, A.; BALASUBRAMANIAN, J.; EDWARDS, G.; DENG, G.; TURKAY, E.; PARSONS, J. Model driven middleware: a new paradigm for developing distributed real-time and embedded systems. **Science of Computer Programming**, Science Direct, v. 73, n. 1, p. 39–58, 2008. Citation on page 35.

GOKHALE, A.; BALASUBRAMANIAN, K.; KRISHNA, A.; BALASUBRAMANIAN, J.; EDWARDS, G.; DENG, G.; TURKAY, E.; PARSONS, J. Model driven middleware: a new paradigm for developing distributed real-time and embedded systems. **Science of Computer Programming**, v. 73, n. 1, p. 39–58, 2008. Citation on page 212.

GOLDSBY, H. J.; SAWYER, P.; BENCOMO, N.; CHENG, B. H. C.; HUGHES, D. Goal-based modeling of dynamically adaptive system requirements. In: **15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008)**. Washington, USA: IEEE Computer Society, 2008. p. 36–45. Citations on pages 31 and 144.

GOROD, A.; SAUSER, B.; BOARDMAN, J. System-of-systems engineering management: A review of modern history and a path forward. **IEEE Systems Journal**, IEEE, v. 2, n. 4, p. 484–499, 2008. Citations on pages 14 and 199.

GUESSI, M. **Synthesis of Software Architectures for Systems-of-Systems: an automated method by constraint solving**. Master's Thesis (PhD Thesis) — Institute of Mathematics and Computational Science - ICMC - University of São Paulo (USP), São Carlos, Brazil, 2017. Citation on page 35.

GUESSI, M.; GRACIANO, V. V. N.; BIANCHI, T.; FELIZARDO, K. R.; OQUENDO, F.; NAKAGAWA, E. Y. A systematic literature review on the description of software architectures for systems of systems. In: **30th Symposium on Applied Computing ACM (SAC 2015)**. Salamanca, Spain: ACM, 2015. p. 1–8. Citation on page 207.

GUTTAG, J. V.; HORNING, J. J. **Larch: Languages and Tools for Formal Specification**. : Springer-Verlag New York, Inc., 1993. Citation on page 27.

HAIMES, Y. Y. Modeling complex systems of systems with phantom system models. **Systems Engineering**, John Wiley and Sons Ltd., v. 15, n. 3, p. 333–346, 2012. Citation on page 30.

HALAWEH, M. Integration of grounded theory and case study: An exemplary application from e-commerce security perception research. **Journal of International Technology Theory and Application**, JITTA, v. 13, n. 1, p. 31–51, 2011. Citation on page 63.

HALAWEH, M. Using grounded theory as a method for system requirements analysis. **JISTEM - Journal of Information Systems and Technology Management**, Scielo, v. 9, n. 1, p. 23–38, 2012. Citation on page 43.

HALEY, C. B.; NUSEIBEH, B. The changing role of requirements and architecture in systems engineering. In: **4th International Symposium on Information Technology (ITSim 2008)**. Kuala Lumpur, Malaysia: IEEE, 2008. p. 1–8. Citation on page 90.

HALLERSTEDE, S.; HANSEN, F.; HOLT, J.; LAURITSEN, R.; LORENZEN, L.; PELESKA, J. Technical challenges of sos requirements engineering. In: **7th International Conference on System of Systems Engineering (SoSE 2012)**. Genoa, Italy: IEEE, 2012. p. 573–578. Citation on page 24.

HALLSTEINSEN, S.; GEIHS, K.; PASPALLIS, N.; ELIASSEN, F.; HORN, G.; LORENZO, J.; MAMELLI, A.; PAPADOPOULOS, G. A development framework and methodology for self-adapting applications in ubiquitous computing environments. **Journal of Systems and Software**, Elsevier, v. 85, n. 12, p. 2840–2859, 2012. Citation on page 26.

HALLSTEINSEN, S.; GEIHS, K.; PASPALLIS, N.; ELIASSEN, F.; HORN, G.; LORENZO, J.; MAMELLI, A.; PAPADOPOULOS, G. A development framework and methodology for self-adapting applications in ubiquitous computing environments. **Journal of Systems and Software**, v. 85, n. 12, p. 2840–2859, 2012. Citation on page 212.

HAMILTON, J. A.; SUMMERS, P. A.; ROSEN, J. D. **An Interoperability Road Map for C4ISR Legacy Systems**. 2002. Citation on page 38.

HAN, L.; LIU, J.; ZHOU, T.; SUN, J.; CHEN, X. Safety requirements specification and verification for railway interlocking systems. In: **40th IEEE Annual Computer Software and Applications Conference (COMPSAC 2016)**. Atlanta, USA: IEEE, 2016. p. 335–340. Citation on page 32.

HAN, S. Y.; DELAURENTIS, D. Development interdependency modeling for system-of-systems(sos) using bayesian networks: Sos management strategy planning. In: **Conference on Systems Engineering Research (CSER 2013)**. Atlanta, GA: Elsevier, 2013. p. 698–707. Citations on pages 1, 16, 17, and 31.

HAYMAN, O. K.; NORDIN, A.; ISMAIL, A. R.; SULAIMAN, S. An analysis of ambiguity detection techniques for software requirements specification (srs). **International Journal of Engineering & Technology**, IJET, v. 7, n. 2, p. 501–505, 2018. Citation on page 49.

HAZZAN, O. **Qualitative Research in Software Engineering**. 2010. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.7613. [Online, Acessed: september 25, 2019]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.7613>. Citation on page 44.

HEIMDAHL, M. P. E.; LEVESON, N. G. Completeness and consistency in hierarchical state-based requirements. **IEEE Transactions on Software Engineering**, IEEE Press, v. 22, n. 6, p. 363–377, 1996. ISSN 0098-5589. Citation on page 30.

HENSHAW, M. **The Systems of Systems Engineering Strategic Research Agenda**. 2013. Citations on pages 18 and 19.

HERSHEY, P.; TALBOT, J. **The Role of Modeling and Simulation in Supporting the Internet of Things as Systems-of-Systems**. 2019. Citation on page 39.

HOFMEISTER, C.; KRUCHTEN, P.; NORD, R. L.; OBBINK, H.; RAN, A.; AMERICA, P. A general model of software architecture design derived from five industrial approaches. **Journal of Systems and Software**, Elsevier, v. 80, n. 1, p. 106–126, 2007. Citation on page 92.

HOGIE, K.; CRISCUOLO, E.; PARISE, R. Using standard internet protocols and applications in space. **Computer Networks**, ScienceDirect, v. 47, n. 5, p. 603–650, 2005. Citation on page 41.

HOLT, J.; PERRY, S.; BROWNSWORD, M.; CANCILA, D.; HALLERSTEDE, S.; HANSEN, F. O. Model-based requirements engineering for system of systems. In: **7th International Conference on System of Systems Engineering (SoSE 2012)**. Genoa, Italy: IEEE, 2012. Citation on page 25.

HOLT, J.; PERRY, S.; PAYNE, R.; J., B.; HALLERSTEDE, S.; HANSEN, F. O. A model-based approach for requirements engineering for systems of systems. **IEEE Systems Journal**, IEEE, v. 9, n. 1, p. 252–262, 2015. Citation on page 25.

HONOUR, E. Verification and validation issues in systems-of-systems. In: **1st Workshop on Advances in Systems of Systems (AiSoS 2013)**. Roma, Italy: Larsen, Legay, Nyman (Eds.), 2013. p. 2–7. Citations on pages 26, 31, 32, and 33.

HORITA, F. E. A.; FAVA, M. C.; MEDIONDO, E. M.; ROTAVA, J.; SOUZA, V. C.; UEYAMA, J.; ALBUQUERQUE, J. P. Agora-geodash: A geosensor dashboard for real-time flood risk monitoring. In: **11th International Conference on Information Systems for Crisis Response And Management (ISCRAM 2014)**. Pennsylvania, USA: The Pennsylvania State University, 2014. p. 1–10. Citation on page 21.

HU, L.; YONGLIANG, T.; YUAN, G.; JINPENG, B.; JIANGAN, B. System of systems oriented flight vehicle conceptual design: Perspectives and progresses. **Chinese Journal of Aeronautics**, ScienceDirect, v. 28, n. 3, p. 617–635, 2015. Citation on page 212.

HULGAN, J. **Requirements Architecture Part 1: What is Requirements Architecture and why is it important?** 2012. https://seilevel.com/requirements/requirements-architecture-part-1-what-is-requirements-architecture-and-why-is-it-important. [Online, Acessed: October 13, 2018]. Available: <https://seilevel.com/requirements/requirements-architecture-part-1-what-is-requirements-architecture-and-why-is-it-important>. Citation on page 93.

IEEE 610. **IEEE standard glossary pf software engineering terminology - IEEE Std 610-12-1990**. 1990. Citation on page 194.

ISO/DIS 26262. **Road vehicles — Functional safety**. 2009. Citation on page 141.

ISO/IEC-13568. **Information technology — Z formal specification notation — Syntax, type system and semantics**. 2002. Citation on page 30.

ISO/IEC-25010. **Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models - ISO/IEC 25010:2011**. 2011. Citations on pages 11, 81, and 139.

ISO/IEC-25010. **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models - ISO/IEC 25010:2011**. 2011. Citations on pages 261 and 271.

ISO/IEC/IEEE-15288. **Systems and software engineering: Software life cycle processes - ISO/IEC/IEEE 15288:2015**. 2015. Citations on pages 21 and 25.

ISO/IEC/IEEE-21839. **Systems and software engineering– Systems of Systems (SoS) considerations in life cycle stage of a system**. 2019. Citation on page 67.

ISO/IEC/IEEE-21840. **Systems and software engineering– Guidelines for the utilization od ISO/IEC/IEEE 15288 in the context of Systems of Systems (SoS) Engineering**. 2019. Citation on page 67.

ISO/IEC/IEEE-24765. **Systems and Software Engineering Vocabulary**. 2017. Citations on pages 14, 26, 68, and 139.

ISO/IEC/IEEE-25012. **Software engineering —-Software product Quality Requirements and Evaluation (SQuaRE) —- Data quality model (ISO/IEC 25010:2019)**. 2019. Citation on page 139.

ISO/IEC/IEEE-29148. **Systems and software engineering Life cycle processes - Requirements engineering - ISO/IEC/IEEE 29148**. 2011. Citations on pages 21, 22, 67, 68, 103, and 195.

ISO/IEC/IEEE-42010. **Systems and software engineering - Architecture description - ISO/IEC/IEEE 42010**. 2011. Citations on pages 14, 35, and 36.

JACKSON, M. C.; KEY, P. Towards a system of systems methodologies. **The Journal of the Operational Research Society**, Palgrave Macmillan Journals, v. 35, n. 6, p. 473–486, 1984. Citation on page 14.

JACOB, F. **The Logic of Living Systems:History of Heredity**. : Allen Lane, 1974. Citation on page 14.

JAMSHIDI, M. System of systems engineering-new challenges for the 21st century. **IEEE Aerospace and Electronic Systems Magazine**, IEEE, v. 23, n. 5, p. 4–19, 2008. Citations on pages 2, 4, 15, 21, 22, and 156.

JAMSHIDI, M. **Systems of Systems Engineering: Principles and Applications**. : CRC Press, 2009. Citations on pages 21, 34, 36, 75, 76, and 130.

JARKE, M.; LOUCOPOULOS, P.; LYYTINEN, K.; MYLOPOULOS, J.; ROBINSON, W. The brave new world of design requirements. **Information Systems**, Elsevier Science Ltd., v. 36, n. 7, p. 992–1008, 2011. Citations on pages 24 and 27.

JAYAPUTERA, G.; LOKE, S.; ZASLAVSKY, A. Mission impossible? automatically assembling agents from high-level task descriptions. In: **IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)**. Halifax, Canada: IEEE, 2003. p. 1–7. Citations on pages 125 and 126.

JOHNSON, B.; HOLNESS, K.; PORTER, W.; HERNANDEZ, A. Complex adaptive systems of systems: A grounded theory approach. **The Grounded Theory Review**, http://groundedtheoryreview.com, v. 17, n. 1, p. 52–69, 2018. Citation on page 49.

JR., R. H. K.; TOLK, A. A systems engineering process for development of federated simulations. In: **SpringSim**. 2009. p. 1–8. Citation on page 212.

KASUNIC, M. **Measuring Systems Interoperability**. 2001. Citation on page 37.

KASUNIC, M. **Designing an Effective Survey**. 2005. Citation on page 57.

KAZMAN, R.; BASS, L.; WEBB, M.; ABOWD, G. SAAM: a method for analyzing the properties of software architectures. In: **16th international conference on Software engineering (ICSE 1994**. Sorrento, Italy: IEEE, 1994. p. 81–90. Citations on pages 34 and 36.

KAZMAN, R.; KLEIN, M.; BARBACCI, M.; LONGSTAFF, T.; LIPSON, H.; CARRIERE, J. The architecture trade-off analysis method. In: **4th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 1998)**. Monterey, California: IEEE, 1998. p. 68–78. Citation on page 36.

KAZMAN, R.; SCHMID, C.; NIELSEN, C. B.; KLEIN, J. Understanding patterns for system of systems integration. In: **8th International Conference on System of Systems Engineering (SoSE 2013)**. Maui, USA: IEEE, 2013. p. 141–146. Citations on pages 20 and 74.

KEATING, C.; ROGERS, R.; UNAL, R.; DRYER, D.; SOUSA-POZA, A.; SAFFORD, R.; PETERSON, W.; RABADI, G. System of systems engineering. **Engineering Management Journal**, Taylor & Francis, v. 15, n. 3, p. 36–45, 2003. Citation on page 199.

KEATING, C. B.; PADILHA, J. J.; ADAMS, K. System of systems engineering requirements: Challenges and guidelines. **Engineering Management Journal**, Old Dominion University, v. 20, n. 4, p. 24–31, 2015. Citations on pages 1, 16, 24, 25, 26, 90, and 199.

KHAN, H. H.; MAHRIN, M. N.; CHUPRAT, S. Situational factors affecting requirement engineering process in global software development. In: **IEEE Conference on Open Systems (ICOS 2013)**. Sarawak, Malaysia: IEEE Computer Society, 2013. p. 118–122. Citation on page 100.

KHAN, K. S.; RIET, G. T.; GLANVILLE, J.; SOWDEN, A. J.; KLEIJNEN, J. *et al.* **Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews**. : NHS Centre for Reviews and Dissemination, 2001. Citation on page 209.

KI-ARIES, D.; FAILY, S.; DOGAN, H.; WILLIANS, C. System of systems characterisation assisting security risk assessment. In: **13th Annual Conference on System of Systems Engineering (SoSE 2018)**. Paris, France: IEEE, 2018. p. 485–492. Citations on pages 75, 77, and 78.

KILICAY-ERGIN, N.; DAGLI, C. Incentive-based negotiation model for system of systems acquisition. **Systems Engineering**, v. 18, n. 3, p. 310–321, 2015. Citation on page 212.

KINGSTO, G.; FEWELL, S.; RICHER, W. An organisational interoperability agility model. In: **10th International Command and Control Research and Technology Symposium (ICCRTS 2005)**. Washington, USA: Crown Copyright, 2005. p. 158–176. Citation on page 38.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. 2007. Citations on pages 7, 46, 51, 205, 209, 223, and 224.

KITCHENHAM, B.; SJØBERG, D. I. K.; BRERETON, O. P.; BUDGEN, D.; DYBÅ, T.; HöST, M.; PFAHL, D.; RUNESON, P. Can we evaluate the quality of software engineering experiments? In: **4th International Symposium on Empirical Software Engineering and Measurement (ESEM 2010)**. Bolzano, Italy: ACM, 2010. p. 1–8. Citations on pages 27, 207, and 209.

KNODEL, J.; NAAB, M. **Pragmatic Evaluation of Software Architecture**. : Springer, 2016. Citations on pages 3, 62, and 151.

KRAMER, J.; MAGEE, J. Self-managed systems: An architectural challenge. In: **2007 Future of Software Engineering (FOSE 2007)**. Washington, USA: IEEE Computer Society, 2007. p. 259–268. Citation on page 145.

KRIESBERG, L. Special issue: Collaborative problem solving amid conflicts. In: ZHU, Y. L. C. M. G. Z. (Ed.). : Willey Online Library, 2012. chap. Mediation in Conflict Systems, p. 149–162. Citation on page 79.

KRISHNA, A.; VILKOMIR, S. A.; GHOSE, A. K. Consistency preserving co-evolution of formal specifications and agent-oriented conceptual models. **Information and Software Technology**, Butterworth-Heinemann, Newton, USA, v. 51, n. 2, p. 478–496, 2009. Citation on page 31.

KUBICEK, H.; CIMANDER, R.; SCHOLL, J. Organizational interoperability in e-government,. In: KUBICEK RALF CIMANDER, H. J. S. H. (Ed.). : Springer-Verlag Berlin Heidelberg, 2011. chap. Layers of Interoperability, p. 85–96. Citations on pages 36, 37, and 39.

KUMAR, P.; MERZOUKI R.AND BOUAMAMA, O. B. Multilevel modeling of system of systems. **IEEE Transactions on Systems, Man, and Cybernetics**, IEEE, Early Access Articles, n. 0, p. 1–12, 2017. Citations on pages 16, 22, 40, 125, 126, and 147.

LAMSWEERDE, A. v. Formal specification: A roadmap. In: **Conference on The Future of Software Engineering (ICSE 2000**. Limerick, Ireland: ACM, 2000. p. 147–159. Citations on pages 28 and 32.

LAMSWEERDE, A. V. Goal-oriented requirements engineering: A guided tour. In: **Fifth IEEE International Symposium on Requirements Engineering (RE 2001)**. Toronto, Canada: IEEE Computer Society, 2001. p. 249–262. Citation on page 124.

LANA, C. A.; GUESSI, M.; ANTONINO, P. O.; ROMBACH, D.; NAKAGAWA, E. Y. A systematic identification of formal and semi-formal languages and techniques for software-intensive systems-of-systems requirements modeling. **IEEE Systems Journal**, p. 1–12, 2018. ISSN 1932-8184. Citations on pages 27, 30, 151, and 159.

LANA, C. A.; SOUZA, N. M.; DELAMARO, M. E.; NAKAGAWA, E. Y.; OQUENDO, F.; MALDONADO, J. C. Systems-of-systems development: Initiatives, trends, and challenges. In: **XLII Latin American Computer Conference (CLEI 2016)**. Valparaiso, Chile: IEEE, 2016. p. 1–12. Citations on pages 14, 90, and 151.

LANE, A. J. Systems-of-systems capability to requirements engineering. In: **9th International Conference on Systems of Systems Engineering (SOSE)**. Adelaide, Australia: IEEE, 2017. p. 91–96. Citations on pages 132, 134, 147, and 193.

LANE, J. A. **What is a system-of-system and why should I care?** 2013. Citation on page 18.

LARSON, W. J.; WERTZ, J. R. **Space Mission Analysis and Design**. 2. ed. Hawthorne, CA: Microcosm Press, 1991. Citations on pages 23, 93, and 194.

LARSSON, J.; BORG, M.; OLSSON, T. Testing quality requirements of a system-of-systems in the public sector - challenges and potential remedies. In: **3rd International Workshop on Requirements Engineering and Testing (RET 2016)**. Gothenburg, Sweden: Cornell University Library, 2016. p. 1–15. Citation on page 90.

LAWRENCE, B. B. **Qualitative research methods for the social sciences**. 4. ed. : Pearson Education, Inc., 2000. Citation on page 50.

LEGGETT, L. **Glossary of Defense Acquisition Acronyms & Terms**. 2017. Citations on pages 1, 105, and 194.

LEITE, M. J. Interoperability assessment methodology. In: **66th Military Operations Research Society Symposium (MORSS 1998)**. Monterey, CA: Joint Theater Air and Missile Defense Organization, 1998. p. 1–14. Citation on page 38.

LEVESON, N. The drawbacks in using the term systems-of-systems. **Biomedical Instrumentation & Technology**, Advancing Safety in Health technology (AAMI), v. 47, n. 2, p. 115–118, 2013. Citations on pages 14 and 194.

LEVESON, N. G.; HEIMDAHL, M. P. E.; HILDRETH, H.; REESE, J. D. Requirements specification for process-control systems. **IEEE Transactions on Software Engineering**, IEEE Press, v. 20, n. 9, p. 684–707, 1994. Citation on page 30.

LEWIS, G.; MORRIS, E.; PLACE, P.; SIMANTA, S.; SMITH, D. Requirements engineering for systems of systems. In: **3rd Annual IEEE International Systems Conference (SysCon 2009)**. Vancouver, Canada: IEEE, 2009. p. 247–252. Citations on pages 24, 26, 33, and 34.

LI, D.; YANG, Y. Enhance value by building trustworthy software-reliant system of systems from software product lines. In: **PLEASE**. 2012. p. 13–16. Citation on page 212.

LI, F. L. **Desiree: a Refinement Calculus for Requirements Engineering**. Master's Thesis (These) — International Doctorate School in Information and Communication Technologies, University of Trento (DISI), 2016. Citation on page 34.

LINHARES, M. V.; OLIVEIRA, R. S.; FARINES, J. M.; VERNADAT, F. Introducing the modeling and verification process in sysml. In: **IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)**. Patras, Greece: IEEE, 2007. p. 344–351. Citations on pages 30, 31, and 32.

LORMANS, M.; DEURSEN, V.; HANS-GERHARD, G. An industrial case study in reconstructing requirements views. **Empirical Software Engineering**, SpringerLink, v. 13, n. 6, p. 727–760, 2008. Citation on page 79.

LOUALI, R.; BOUAZIZ, S.; ELOUARDI, A.; ABOUZAHIR, M. Platform simulation based unmanned aircraft systems design. In: **WCCS**. 2014. p. 736–742. Citation on page 212.

LOUREIRO, G.; PANADES, W. F.; SILVA, A. Lessons learned in 20 years of application of systems concurrent engineering to space products. **Acta Astronautica**, Elsevier, v. 151, p. 44–52, 2018. Citation on page 212.

LU, J.; CHEN, D.; WANG, J.; TORNGREN, M. Towards a service-oriented framework for mbse tool-chain development. In: **13th Annual Conference on System of Systems Engineering (SoSE 2018)**. Paris, France: IEEE, 2018. p. 568–575. Citation on page 212.

LUBAS, D. G.; COMMAND, U. A. F. M. Department of defense system of systems reliability challenges. In: **2017 Annual Reliability and Maintainability Symposium (RAMS 2017)**. Orlando, USA: IEEE, 2017. p. 1–6. Citation on page 16.

LUNA, S.; LOPES, A.; YAN, H.; TAO, S.; ZAPATA, F.; PINEDA, R. Integration, verification, validation, test, and evaluation framework for system-of-systems. **Procedia Computer Science**, v. 20, p. 298–305, 2013. Citations on pages 96 and 212.

LUNA, S.; LOPES, A.; YAN, H.; TAO, S.; ZAPATA, F.; PINEDA, R. Integration, verification, validation, test, and evaluation (IVVT&E) framework for system-of-systems (SoS). **Procedia Computer Science**, Science Direct, v. 20, p. 298–305, 2013. Citations on pages 3 and 33.

MAALEM, S.; ZAROUR, N. Challenge of validation in requirements engineering. **Journal of Innovation in Digital Ecosystems**, Elsevier, v. 3, n. 1, p. 15–21, 2016. Citations on pages 26 and 33.

MACHADO, C.; SILVA, E.; BATISTA, T.; LEITE, J.; NAKAGAWA, E. Y. Architectural elements of ubiquitous systems: a systematic review. In: **8rd International Conference on Software Engineering Advances (ICSEA 2013)**. Venice, Italy: IARIA, 2013. p. 208–213. Citation on page 207.

MACIEL, R. S. P.; DAVID, J. M. N.; CLARO, D. B.; BRAGA, R. Grand research challenges in is in brazil - 2016-2026. In: BOSCARIOLI, R. M. A. C.; MACIEL, R. S. P. (Ed.). : Sociedade Brasileira de Computa ç ao, 2017. chap. Full Interoperability: Challenges and Opportunities for Future Information Systems, p. 107–118. Citation on page 37.

MADEYSKI, L.; ORZESZYNA, W.; TORKAR, R.; JÓZALA, M. Overcoming the equivalent mutant problem: a systematic literature review and a comparative experiment of second order mutation. **IEEE Transactions on Software Engineering**, IEEE, v. 40, n. 1, p. 23–42, 2014. Citation on page 207.

MADNI, A.; SIEVERS, M. System of systems integration: key considerations and challenges. **Journal Systems Engineering**, Wiley Online Library, v. 17, n. 3, p. 330–347, 2014. Citation on page 74.

MADNI, A.; SIEVERS, M. System of systems integration: key considerations and challenges. **Journal Systems Engineering**, v. 17, n. 3, p. 330–347, 2014. Citation on page 212.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M. Reconciling software development models: a quasi-systematic review. **The Journal of Systems and Software**, Elsevier, v. 85, n. 2, p. 351–369, 2012. Citations on pages 53, 209, and 226.

MAIER, M. W. Architecting principles for systems-of-systems. In: **6th International Symposium of the National Council on Systems Engineering (INCOSE 1996)**. Boston, MA: Wiley Online Library, 1996. p. 1–13. Citations on pages 15, 18, 24, and 75.

MAIER, M. W. Architecting principles for systems-of-systems. **Systems Engineering**, Wiley Online Library, v. 1, n. 4, p. 267–284, 1998. Citations on pages 1, 13, 15, 16, 18, 31, 200, and 206.

MAIER, M. W. Modeling and simulation support for system of systems engineering applications. In: RAINEY, L. B.; TOLK, A. (Ed.). Hoboken, NJ: John Wiley & Sons. Inc., 2014. chap. The Role of Modeling and Simulation in System of Systems Development, p. 11–41. Citations on pages 1, 15, and 36.

MALLEK, S.; DACLIN, N.; CHAPULA, V. The application of interoperability requirement specification and verification to collaborative processes in industry. **Computers in Industry**, Elsevier, v. 63, n. 7, p. 643–658, 2012. Citations on pages 41, 67, 68, 151, and 193.

MARTIN, J. **An Information Systems Manifesto**. : Prentice Hall, 1984. Citation on page 2.

MATER, J. **A Smart Grid Interoperability Maturity Model Rating System: Predicting "Plug and Play" Integration Probability**. 2009. https://www.gridwiseac.org/pdfs/forum_papers09/mater.pdf. [Online, Acessed: April 07, 2018]. Available: <https://www.gridwiseac.org/pdfs/forum_papers09/mater.pdf>. Citation on page 39.

MAVIN, A.; WILKINSON, P.; HARWOOD, A.; NOVAK, M. Easy approach to requirements syntax (ears). In: **17th IEEE International Requirements Engineering Conference, (RE 2009)**. Washington, DC, USA: IEEE Computer Society, 2009. p. 317–322. Citation on page 14.

MAYK, I.; MADNI, A. M. **The Role of Ontology in System-of-Systems Acquisition**. 2006. Citation on page 49.

MAZZUCHI, T.; ALBAKRI, G. System of systems engineering facilitates integration of large scale complex systems. **INCOSE**, v. 21, n. 1, p. 811–855, 2011. Citation on page 212.

MEDVIDOVIC, N.; TAYLOR, R. N. A classification and comparison framework for software architecture description languages. **IEEE Transactions on Software Engineering**, IEEE Press, v. 26, n. 1, p. 70–93, 2000. Citation on page 35.

MENS, T.; MAGEE, J.; RUMPE, B. Evolving software architecture descriptions of critical systems. **Computer**, IEEE, v. 43, n. 5, p. 42–48, 2010. Citation on page 36.

MENSH, D. R.; KITE, R. S.; DARBY, P. H. The quantification of interoperability. **Naval Engineering Jornal**, Wiley Online Library, v. 101, n. 3, p. 251–259, 1989. Citation on page 37.

MENSING, B.; GOLTZ, U.; ANICULAESEI, A.; HEROLD, S.; RAUSCH, A.; GARTNER, S.; SCHNEIDER, K. Towards integrated rule-driven software development for it ecosystems. In: **DEST**. 2012. p. 1–6. Citation on page 212.

MERRIAM, S.; TISDELL, E. **Qualitative Research: A guide to design and implamentation**. : Jossey Bass, 2015. Citation on page 157.

MICHENI, E.; MUKETHA, G. M.; WAMOCHO, L. A review of agent based interoperability frameworks and interoperability assessment models. **Scholars Journal of Engineering and Technology (SJET)**, Scholar Academic and Scientific Publisher, v. 2, n. 2B, p. 291–300, 2014. Citation on page 39.

MILES, M. B.; HUBERMAN, A. M.; SALDANA, J. **Qualitative Data Analysis: a methods sourcebook**. 3. ed. : Sage Publications, Inc, 2014. Citations on pages 43, 44, 49, and 148.

MISIC, V. B.; VELASEVIC, D. M. Formal specification in software development: A overview. **Yugoslav Journal of Operations Research**, Journal of Operations Research, v. 7, n. 1, p. 79–96, 1997. Citation on page 28.

MISURACA, G.; ALFANO, G.; VISCUSI, G. Interoperability challenges for ict-enabled governance: Towards a pan-european conceptual framework. **Journal of Theoretical and Applied Electronic Commerce Research**, www.jtaer.com, v. 6, n. 1, p. 95–111, 2011. Citation on page 114.

MITTAL, S.; RAINEY, L. Harnessing emergence: The control and design of emergent behavior in system of systems engineering. In: **Conference on Summer Computer Simulation (SummerSim 2015)**. San Diego, CA, USA: Society for Computer Simulation International, 2015. p. 1–10. Citations on pages 13 and 36.

MITTAL S.AND DIALLO, S.; TOLK, A. **Emergent Behanvior in Complex Systems Engineering: A Modeling and Simulation Approach**. : John Wiley & Sons, Inc, 2018. Citations on pages 22 and 36.

MOHAPATRA, P. K. J. **Software Engineering: A lifecycle Approach**. : New Age International (P) Limited, 2010. Citation on page 33.

MOKHTARPOUR, B.; STRACENER, J. A conceptual methodology for selecting the preferred system of systems. **IEEE System Journal**, IEEE, v. 11, n. 4, p. 1928–1934, 2017. Citations on pages 2 and 40.

MOKHTARPOUR, B.; STRACENER, J. T. Mission reliability analysis of phased-mission systems-of-systems with data sharing capability. In: **Annual Reliability and Maintainability Symposium (RAMS 2015)**. Palm Harbor, USA: IEEE, 2015. p. 1–6. Citation on page 54.

MONOSTORI, l.; BAUERNHANSL, K.; KONDOH, S.; KUMARA, S.; REINHART, G.; SAUER, O.; SCHUH, G.; SIHN, W.; UEDA, K. Cyber-physical systems in manufacturing. **Manufacturing Technology**, Science Direct, v. 65, n. 2, p. 621–641, 2016. Citation on page 39.

MORELAND., J. D. Integration & interoperability mission engineering. In: DURANT, B. R.; JR., J. D. M.; BARON, N. T. (Ed.). : Leading Edge: Naval Surface Warfare Center, Dahlgren Division, 2015. chap. Mission Engineering Integration and Interoperability (I&I), p. https://www.navsea.navy.mil/Portals/103/Documents/NSWC _Dahlgren/LeadingEdge /LE_IandI_Jan2015_FINAL_web.pdf. Citation on page 67.

MORRIS, E.; LEVINE, L.; PLACE, P.; PLAKOSH, D. **System of Systems Interoperability (SOSI): Final Report**. 2004. Citation on page 38.

MOSTERMANM, P. J.; GHIDELLA, J.; FRIEDMAN, J. Model-based design for system integration. In: **CDEN**. : Open Journal Systems, 2005. p. 1–10. Citation on page 212.

MUNIR, H.; MOAYYED, M.; PETERSEN, K. Considering rigor and relevance when evaluating test driven development: a systematic review. **Information and Software Technology**, Elsevier, v. 56, n. 4, p. 375–394, 2014. Citation on page 207.

NAIK, N. Building a virtual system of systems using docker swarm in multiple clouds. In: **2016 IEEE International Symposium on Systems Engineering (ISSE 2016)**. Edinburgh, UK: IEEE, 2016. p. 1–3. Citation on page 212.

NAKAGAWA, E. Y.; BECKER, M.; MALDONADO, J. C. A knowledge-based framework for reference architectures. In: **27th Annual ACM Symposium on Applied Computing (SAC 2012)**. Trento, Italy: ACM, 2012. p. 1197–1202. Citation on page 115.

NAKAGAWA, E. Y.; GONçALVES, M. B.; GUESSI, M.; OLIVEIRA, L. B. R.; OQUENDO, F. The state of the art and future perspectives in systems of systems software architectures. In: **1th International Workshop on Software Engineering for Systems-of-Systems (SESoS 2013)**. Montpellier, France: ACM, 2013. p. 13–20. Citation on page 36.

NARCISO, E. N.; DELAMARO, M. E.; NUNES, F. L. S. Test case selection: a systematic literature review. **International Journal of Software Engineering and Knowledge Engineering**, World Scientific Publishing, v. 24, n. 4, p. 653–676, 2014. Citation on page 207.

NASA. **Formal methods specification and analysis guidebook for the verification of software and computer systems - Volume II: A practitioner's companion**. 1997. Citations on pages 28, 30, 32, and 36.

NASA. **The Systems Engineering (SE) Process**. 2018. https://www.nasa.gov/pdf/598887main_Auburn_PowerPoints_SE.pdf. [Online, Acessed: June 08, 2018]. Available: <https://www.nasa.gov/pdf/598887main_Auburn_PowerPoints_SE.pdf>. Citations on pages 105 and 194.

NATARAJAN, S.; KUMAR, A.; NORI, K. V. Architecture as a solution schema for a class of problems. In: **26th Annual INCOSE International Symposium (IS 2016)**. Scotland, UK: INCOSE, 2016. p. 1–15. Citation on page 90.

NATO. **NATO Allied Data Publication 34 (ADatP-34): NATO C3 Technical Architecture**. 2003. Citation on page 38.

NATO. **NATO Interoperability Standards and Profiles**. 2017. Citation on page 38.

NCUBE, C. On the engineering of systems of systems: Key challenges for the requirements engineering community. In: **Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RESS 2011)**. Trento, Italy: IEEE, 2011. p. 70–73. Citations on pages 13, 25, and 26.

NCUBE, C.; LIM, S. L. On systems-of-systems engineering: a requirements enginnering perspective and research agenda. In: **2018 IEEE 26th Requirements Engineering Conference (RE 2018)**. Banff, Canada: IEEE, 2018. p. 112–123. Citations on pages 2, 3, 4, 24, 72, 74, 148, and 151.

NCUBE, C.; LIM, S. L.; DOGAN, H. Identifying top challenges for international research on requirements engineering for systems of systems engineering. In: **21st IEEE International Requirements Engineering Conference (RE 2013)**. Rio de Janeiro, Brazil: PUC-RIO, 2013. p. 342–344. Citations on pages 1, 2, 17, 26, and 27.

NETO, V. V. G.; GUESSI, M.; OLIVEIRA, L. B. R.; OQUENDO, F.; NAKAGAWA, E. Y. Investigating the model-driven development for systems-of-systems. In: **European Conference on Software Architecture Workshops (ECSAW 2014)**. Vienna, Austria: ACM, 2014. p. 1–8. Citations on pages 31, 32, and 35.

NGUYEN, T. H.; VO B. Q.AND LUMPE, M.; GRUNDY, J. KBRE: a framework for knowledge-based requirements engineering. **Software Quality Journal**, SpringerLink, v. 22, n. 1, p. 87–119, 2014. Citation on page 32.

NIELSEN, C. B.; LARSEN, P. G.; FITZGERALD, J.; WOODCOCK, J.; PELESKA, J. Systems of systems engineering: basic concepts, model-based techniques, and research directions. **ACM Computing Survey**, ACM, v. 48, n. 2, p. 18:1–18:41, 2015. Citations on pages 1, 15, 16, 18, and 27.

NIELSEN, C. B.; LARSEN, P. G.; FITZGERALD, J.; WOODCOCK, J.; PELESKA, J. Systems of systems engineering: basic concepts, model-based techniques, and research directions. **ACM Computing Survey**, v. 48, n. 2, p. 18:1–18:41, 2015. Citations on pages 3 and 13.

NISAR, S.; NAWAZ, M.; SIRSHAR, M. Review analysis on requirement elicitation and its issues. **International Journal of Computer and Communication System Engineering (IJCCSE)**, IJCCSE, v. 2, n. 3, p. 547–552, 2015. Citation on page 98.

NUSEIBEH, B. Weaving together requirements and architectures. **Computer**, IEEE Computer Society Press, v. 34, n. 3, p. 115–117, 2001. Citations on pages 33, 89, and 90.

NUSEIBEH, B.; KRAMER, J.; FINKELSTEIN, A. A framework for expressing the relationships between multiple views in requirements specification. **IEEE Transactions on Software Engineering**, IEEE, v. 20, n. 10, p. 760–773, 1994. Citation on page 79.

Object Management Group. **Systems Modeling Language**. 2016. Available at <http://www.omgsysml.org/>. [Online, Acessed: December 06, 2016]. Citation on page 27.

Object Management Group. **Systems Modeling Language**. 2018. Available at <http://www.omgsysml.org/>. [Online, Acessed: August 06, 2018]. Citations on pages 34 and 35.

Object Management Group. **Unified Modeling Language: what is UML?** 2018. Available at <http://www.uml.org/>. [Online, Acessed: January 18, 2018. Citations on pages 34 and 35.

OFFICER DoD D. C. I. **Levels of Information System Interoperability (LISI)**. 1998. Citation on page 38.

OFFICER DoD D. C. I. **United States Department of Defense Architecture Framework (DoDAF)**. 2010. Citations on pages 36, 37, and 212.

PANDIT, N. R. The creation of theory: A recent application of the grounded theory method. **The Qualitative Report**, https://nsuworks.nova.edu/tqr/vol2/iss4/3/, v. 2, n. 4, p. 1–15, 1996. Citations on pages 7, 11, 46, 47, and 203.

PANETTO, H.; CECIL, J. Information systems for enterprise integration, interoperability and networking: theory and applications. **Enterprise Information Systems**, Springer, v. 7, n. 1, p. 1–6, 2013. Citation on page 3.

PARTNERS, M. **MOD Architectural Framework: Technical Handbook**. 2005. Citations on pages 36 and 212.

PEARSON, G.; KOLODNY, M. U.k. mod land open systems architecture and coalition interoperability with the u.s. In: **Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IV**. Baltimore, USA: SPIE Digital Library, 2013. p. 8742 – 8742 – 9. Citation on page 67.

PEPER, C.; GOTZHEIN, R.; KRONENBURG, M. A generic approach to the formal specification of requirements. In: **First IEEE International Conference on Formal Engineering Methods (ICFEM, 1997)**. Hiroshima, Japan: IEEE, 1997. p. 252–261. Citation on page 53.

PERISTERAS, V.; TARABANIS, K. The connection, communication, consolidation, collaboration interoperability framework (c4if) for information systems interoperability. **Interoperability in Business Information Systems (IBIS)**, http://www.ibis-journal.net, v. 1, n. 1, p. 61–72, 2006. Citation on page 38.

PERTENSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering. **Information and Software Technology**, ACM, v. 64, n. C, p. 1–18, 2015. Citations on pages 51 and 207.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering. **Information and Software Technology**, Butterworth-Heinemann, v. 64, p. 1–18, 2015. Citations on pages 7 and 27.

PICCOLO, A.; GALDI, V.; SENESI, F.; MALANGONE, R. Use of formal languages to represent the ertms/etcs system requirements specifications. In: **International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion, and, Road Vehicles (ESARS 2015)**. Aachen, Germany: IEEE, 2015. p. 1–5. Citation on page 31.

PMI. Citation on page 130.

POCHA, J. J. **An Introduction to Mission Design for Geostationary Satellites**. Hawthorne, CA: Kluwer Academic Publishers, 1987. Citation on page 2.

POLACSEK, M.; BOARDMAN, G.; MCCANN, T. **Understanding, choosing and applying grounded theory: part 2**. 2018. https://journals.rcni.com/nurse-researcher/evidence-and-practice/understanding-choosing-and-applying-grounded-theory-part-2-nr.2018.e1593/abs. [Online, Acessed: september 25, 2019]. Available: <https://journals.rcni.com/nurse-researcher/evidence-and-practice/understanding-choosing-and-applying-grounded-theory-part-2-nr.2018.e1593/abs>. Citations on pages 44 and 45.

PONSARD, C.; MASSONET, P.; RIFAUT, A.; MOLDEREZ, J. F.; LAMSWEERDE, A. van; VAN, H. T. Early verification and validation of mission critical systems. **Electronic Notes in Theoretical Computer Science (ENTCS)**, Elsevier Science Publishers B. V., Amsterdam, Netherlands, v. 133, n. 31, p. 237–254, 2005. Citations on pages 27, 28, and 31.

PORTER, M. E. **Competitive Strategy: Techniques for Analyzing Industries and Competitors**. : Division of Simon & Schuster Inc., 1998. Citation on page 101.

POULOUDI, A.; WHITLEY, E. A. Stakeholder identification in inter-organizational systems: gaining insights for drug use management systems. **European Journal of Information Systems**, Springer, v. 6, n. 1, p. 1–14, 1997. Citation on page 133.

PSAIER, H.; DUSTDAR, S. A survey on self-healing systems: Approaches and systems. **Computing**, Springer-Verlag New York, Inc., v. 91, n. 1, p. 43–73, 2011. Citation on page 76.

QURESHI, N. A.; NGUYEN, C.; PERINI, A. Analyzing interoperability requirements for adaptive services-based applications: A goal-oriented approach. In: **34th Annual IEEE Computer Software and Applications Conference Workshop (COMPSACW 2010)**. Trento, Italy: IEEE, 2010. p. 239–244. Citations on pages 41 and 66.

RAJAGOPAL, P.; LEE, R.; AHLSWEDE, T.; CHIANG, C.-C.; KAROLAK, D. A new approach to software requirements elicitation. In: **6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD-SAWN 2005)**. Washington, USA: IEEE Computer Society, 2005. p. 32–42. Citation on page 98.

RAMOS, M.; MASIERO, P.; BRAGA, R.; PENTEADO, R. Reengineering legacy systems towards system of systems development. In: **13th IEEE International Conference on Information Reuse and Integration (IRI 2012)**. Las Vegas, USA: IEEE, 2012. p. 394–401. Citation on page 207.

RAMOS, M.; MASIERO, P.; BRAGA, R.; PENTEADO, R. Reengineering legacy systems towards system of systems development. In: **IRI**. 2012. p. 394–401. Citation on page 212.

RAYMER, D. P. **Aircraft Design: A Conceptual Approach**. Wright-Patterson Air Force Base, Ohio: Education Series, 1992. Citation on page 212.

RAZ, A. K.; KENLEY, C. R.; DELAURENTIS, D. A. A system-of-systems perspective for information fusion system design and evaluation. **Information Fusion**, Elsevier, v. 35, p. 148–165, 2016. Citation on page 212.

REICHLE, R.; WAGNER, M.; KHAN, M. U.; GEIHS, K.; LORENZO, J.; VALLA, M.; FRA, C.; PASPALLIS, N.; PAPADOPOULOS, G. A. A comprehensive context modeling framework for pervasive computing systems. In: **DAIS**. 2008. p. 281–295. Citation on page 212.

REMENYI, D. **Grounded Theory - The Reader Series**. 2. ed. : ACPIL, 2014. Citations on pages 44 and 51.

RESPECT-IT. **A KAOS Tutorial**. 2007. Citations on pages 31 and 34.

REZAI, R.; CHIEW, T. K.; LEE, S. P. A review on e-business interoperability frameworks. **The Journal of Systems and Software**, Science Direct, v. 93, n. 1, p. 199–2016, 2014. Citations on pages 36 and 37.

RIBEIRO FABÍOLA, G. C.; PEREIRA, C. E.; RETTBERG, A.; SOARES, M. S. Model-based requirements specification of real-time systems with uml, sysml and marte. **Software & Systems Modeling**, Springer Berlin Heidelberg, v. 15, n. 57, p. 1–19, 2016. Citation on page 13.

RICCI, N.; FITZGERALD, E.; ROSS, M. A.; RHODES, D. H. Architecting systems of systems with ilities; an overview of the sai method. In: **12th Annual Conference on Systems Engineering Research (CSER 2014)**. Cambridge, MA: Science Direct, 2014. p. 1–10. Citations on pages 40 and 212.

RODRIGUES, L. M. G. **A Reference Architecture for Healthcare Supportive Home Systems from a Systems-of-Systems Perspective**. Master's Thesis (PhD Thesis) — Institute of Mathematics and Computational Science - ICMC - University of São Paulo (USP), São Carlos, Brazil, 2018. Citation on page 35.

ROGERS, E. M. **Diffusion of Innovations**. : The American Center Library, 1983. Citations on pages 156 and 157.

ROQUE, M.; CHAPURLAT, V. Leveraging knowledge for innovation in collaborative networks-pro-ve. In: L.M. PARASKAKIS I., A. H. C.-M. (Ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. chap. Interoperability in Collaborative Processes: Requirements Characterisation and Proof Approach, p. 555–562. Citation on page 66.

ROSSAK, W.; ZEMEL, T.; KIROVA, V.; JOLOLIAN, L. A two-level process model for integrated system development. In: **ECBS**. 1994. p. 90–96. Citation on page 212.

ROWLEY, T. J. Moving beyond dyadic ties: A network theory of stakeholder influences. **The Academy of Management Review**, JSTOR, v. 24, n. 4, p. 887–910, 1997. Citation on page 76.

ROZENBERG, G.; SALOMAA, A. **Handbook of Formal Languages**. : Springer, 2004. Citation on page 27.

SAEED, S.; YOUSAFSAI, S.; PALADINO, A.; LUCA, L. M. D. Inside-out and outside-in orientations: A meta-analysis of orientation's effects on innovation and firm performance. **Industrial Marketing Management**, ScienceDirect, v. 47, p. 121–133, 2015. Citation on page 79.

SAMMI, R.; RUBAB, I.; QURESHI, M. A. Formal specification languages for real-time systems. In: **International Symposium on Information Technology**. Kuala Lumpur, Malaysia: IEEE, 2010. p. 1642–1647. Citation on page 30.

SANDUKA, I.; OBERMAISSER, R. Model-based development of systems-of-systems with real-time requirements. In: **INDIN**. 2014. p. 188–194. Citation on page 212.

SANTAMARÍA, D.; ALARCÓN, F.; JIMÉNEZ, A.; VIGURIA, A.; BÉJAR, M.; OLLERO, A. Model-based design, development and validation for (UAS) critical software. **Journal of Intelligent & Robotic Systems**, v. 65, n. 1-4, p. 103–114, 2012. Citation on page 212.

SANTOS, D. S.; OLIVEIRA, B.; GUESSI, M.; OQUENDO, F.; DELAMARO, M.; NAKAGAWA, E. Towards the evaluation of system-of-systems software architectures. In: **8th Workshop on Distributed Software Development (WDES 2014)**. Marceio, Brazil: UFAL, 2014. p. 53–57. Citation on page 36.

SCHMIDT, R. F. Software engineering: Architecture-driven software development. In: SCHMIDT, R. F. (Ed.). Waltham, MA: Elsevier, 2013. chap. Software Requirements Analysis Practice, p. 138–157. Citation on page 33.

SCHMITT, B. Experience marketing: Concepts, frameworks and consumer insights. **Marketing**, Now The Essence Knowledge, v. 5, n. 2, p. 55–112, 2011. Citation on page 79.

SEAMAN, C. Guide to advanced empirical software engineering. In: SHULL, F.; SINGER, J.; SJØBERG, D. I. K. (Ed.). London: Springer London, 2008. chap. Qualitative Methods, p. 85–115. Citations on pages 43, 50, and 57.

SEBOK. **Guide to the Systems Engineering Body of Knowledge: Systems Engineering and Management (SEBoK)**. : BKCASE community,v 1.3.1, 2014. Citations on pages 21, 22, 54, 128, and 130.

SELBERG, S. A.; AUSTIN, M. A. Toward an evolutionary system of systems architecture. **INCOSE International Symposium**, John Wiley & Sons, Inc., v. 18, n. 1, p. 1065–1078, 2008. Citation on page 36.

SERPANOS, D. The cyber-physical systems revolution. **Computer**, IEEE Computer Society, v. 51, n. 3, p. 70–73, 2018. Citation on page 71.

SERUGENDO, D. M. G.; FITZGERALD, J.; ROMANOVSKY, A. MetaSelf: An architecture and a development method for dependable self- systems. In: **SAC**. 2010. p. 457–461. Citation on page 212.

SHAMS, F.; SHARIFLOO, A.; MIRAKHORLI, M.; EMAELI, M. A service driven development process (SDDP) model for ultra large scale systems. In: **ULSSIS**. 2008. p. 37–40. Citation on page 212.

SHANMUGAPRIYA, P.; SURESH, R. M. Software architecture evaluation methods - a survey. **International Journal of Computer Applications**, Citeseer, v. 49, n. 16, p. 19–26, 2012. Citation on page 36.

SHARPLES, R. A. Implementation of human system integration (hsi) and nonfunctional characteristics' into the systems engineering lifecycle - a practical approach at airbus defence and space.

In: **6th International Conference on Applied Human Factors and Ergonomics and the Affiliated Conferences (AHFE 2015)**. Nevada, USA: Elsevier, 2015. p. 1896–1902. Citation on page 212.

SIDEREN, M. V.; CHAPULART, V. **Enterprise Interoperability**. : Springer, 2015. Citation on page 62.

SILVA, E.; BATISTA, T.; CAVALCANTE, E. A mission-oriented tool for system-of-systems modeling. In: **IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems**. Florence, Italy: IEEE, 2015. p. 31–36. Citation on page 40.

SILVA, E.; BATISTA, T.; OQUENDO, F. A mission-oriented approach for designing system-of-systems. In: **10th System of Systems Engineering Conference (SoSE 2015)**. Sam Antonio, USA: IEEE, 2015. p. 346–351. Citations on pages 40, 125, and 126.

SILVA, F. S.; SOARES, F. S. F.; PERES, A. L.; AZEVEDO, I. M.; VASCONCELOS, A. P. L.; KAMEI, F. K.; LEMOS, M. S. R. Using cmmi together with agile software development: a systematic review. **Information and Software Technology**, Science Direct, v. 58, n. 14, p. 20–43, 2015. Citation on page 207.

SKOKOVIĆ, P.; RAKIĆ-SKOKOVIĆ, M. Requirements-based testing process in practice. **International Journal of Industrial Engineering and Management**, Faculty of Technical Sciences, v. 1, n. 4, p. 155–161, 2010. Citation on page 2.

SOMMERVILLE, I. **Software engineering**. 10. ed. : Pearson Education Limited, 2016. Citation on page 18.

SOUSA-POZA, A. Mission engineering. **International Journal of Systems of Systems Engineering**, Inderscience Enterprises Ltd., v. 6, n. 3, p. 161–185, 2015. Citations on pages 2, 4, 23, 40, 54, and 148.

SRIVAS, M. K.; MILLER, S. P. **Formal Verification of a Avionics Microprocessor**. 1995. Citations on pages 28, 30, and 32.

STAFF, J. C. **DOD Dictionary of Military and Associated Terms**. 2019. Citation on page 194.

STEFANUS, V. S.; JAMESON, M. The information systems interoperability maturity model (isimm): Towards standardizing technical interoperability and assessment within government. **International Journal of Modern Education and Computer Science (IJMECS)**, MECS Press, v. 4, n. 5, p. 36–41, 2012. Citation on page 38.

STEVENS, J. **The HOW and WHY of Open Architecture**. 2008. Citation on page 79.

STEWART, K.; CLARCK, H.; GOILLAU, P.; VERRALL, N.; WIDDOWSON, M. Non - technical interoperability in multinational forces. In: **9th International Command and Control Research and Technology Symposium (ICCRTS 2004)**. Copenhagen, Denmark: Crown Copyright, 2004. p. 1–9. Citation on page 38.

STOKES, M. D. **Mission Architecture: The Key To Successful Pre-milestone A Systems Engineering**. 2012. Citation on page 95.

STOL, K.; RALPH, P.; FITZGERALD, B. Grounded theory in software engineering research: A critical review and guidelines. In: **38th International Conference on Software Engineering (ICSE 2016)**. Austin, Texas: ACM, 2016. p. 120–131. Citation on page 44.

STRAUSS, A.; CORBIN, J. **Basics of Qualitative Research: Grounded Theory Procedures and Techniques**. : SAGE Publications, Inc., 1990. Citations on pages 7, 43, 44, 46, 48, 49, 51, and 110.

STRAUSS, A.; CORBIN, J. **Basics of Qualitative Research: Grounded Theory Procedures and Techniques**. : SAGE Publications, Inc., 1998. Citations on pages 46, 47, 49, 50, 51, and 60.

STURDIVANT, R. L.; CHONG, E. K. P. Systems engineering baseline concept of a multispectral drone detection solution for airports. **IEEE ACCESS**, IEEE, v. 5, p. 7123–7138, 2017. Citation on page 212.

TANG, W.; NING, B.; XU, T.; ZHAO, L. Scenario-based modeling and verification for ctcs-3 system requirement specification. In: **2nd International Conference on Computer Engineering and Technology (ICCET 2010)**. Chengdu, China: IEEE, 2010. v. 1, p. V1–400–V1–403. Citations on pages 30 and 31.

TANG, W.; NING, B.; XU, T.; ZHAO, L. Scenario-based modeling and verification of system requirement specification for the european train control system. **WIT Transactions on the Built Environment**, WIT Press, v. 114, p. 759 – 770, 2010. Citation on page 30.

TAWEEL, A.; GARCIA, E.; MILES, S.; LUCK, M. Agent-oriented software engineering of distributed ehealth systems. In: **OTM**. 2013. p. 332–341. Citation on page 212.

TJONG, S. F. **Avoiding Ambiguity in Requirements Specifications**. Master's Thesis (These) — Faculty of Engineering & Computer Science, University of Nottingham Malaysia Campus, 2008. Citation on page 49.

TOLK, A. Beyond technical interoperability - introducing a reference model for measures of merit for coalition interoperability. In: **8th National Defense University (CCRTS 2003)**. Washington, USA: Old Dominion University, 2003. p. 1–22. Citations on pages 38 and 148.

TOLK, A. Summers of computer simulation. In: DURAK, U.; D'AMBROGIO, A.; TOLK, A.; DIALLO, S.; ZACHAREWICZ, G.; RISCO-MARTíN, J. L.; BARHAK, J.; HUNTSINGER, R. C.; RAUNAK, M. S. (Ed.). : Springer International Publishing, 2019. chap. Limitations and Usefulness of Computer Simulations for Complex Adaptive Systems Research, p. https://www.springerprofessional.de/en/limitations–and–usefulness–of–computer–simulations–for–complex–a/16707404. Citation on page 76.

TOLK, A.; MUGUIRA, J. A. The levels of conceptual interoperability model. In: **Fall Simulation Interoperability Workshop**. Orlando, USA: Curran Associates, Inc, 2003. p. 53–66. Citations on pages 3, 38, 39, and 151.

TOLK, A.; RAINEY, L. B. Modeling and simulation support for systems of systems engineering applications. In: MITTAL, S. D. S.; (EDS.), A. T. (Ed.). : John Wiley & Sons, Inc, 2015. chap. Towards a Research Agenda for M&S Support of Systems-of-Systems Engineering, p. 583–592. Citation on page 36.

TU, Z.; ZACHAREWICZ, G.; CHEN, D. Harmonized and reversible development framework for hla based interoperable application. In: **DEVS**. 2011. p. 51–58. Citation on page 212.

UNGER, C.; CIMIANO, P. Representing and resolving ambiguities in ontology-based question answering. In: **Proceedings of the TextInfer 2011 Workshop on Textual Entailment (EMNLP 2011)**. Edinburgh, UK: Association for Computational Linguistics, 2011. p. 40–49. Citation on page 49.

URQUHART, C. Exploring analyst-client communication: Using grounded theory techniques to investigate interaction in informal requirements gathering. In: **IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research**. Philadelphia, USA: Chapman & Hall, Ltd., 1997. p. 149–181. Citations on pages 43 and 63.

URWIN, E. N.; VENTERS, C. C.; RUSSELL, D. J.; LIU, L.; LUO, Z.; WEBSTER, D. E.; HENSHAW, M.; XU, J. Scenario-based design and evaluation for capability. In: **5th International Conference on System of Systems Engineering**. Loughborough, UK: IEEE, 2010. p. 1–6. Citation on page 36.

USCHOLD, M.; GRUNINGE, M. **Ontologies Principles Methods and Application**. 1996. Citations on pages 126 and 127.

VASSALOS, V.; PAPAKONSTANTINOU, Y. Using knowledge of redundancy for query optimization in mediators. In: **AAAI Workshop on AI and Information Integration**. Madison, Wisconsin: Stanford InfoLab Publication Server, 1998. p. 1–8. Citation on page 80.

VEAN, G. E. L. Interoperability in defense communications. **IEEE Transsactions on Communications**, IEEE, v. 28, n. 9, p. 1445–1455, 1980. Citation on page 37.

VEER H, V. D.; WILES, A. **Achieving Technical Interoperability - the ETSI Approach**. 2006. Citation on page 39.

VESONDER, G.; D., V.; HUTCHISON, N.; S., L.; MILLER, W.; TAO, H. Y. S.; WADE, J. **RT-171: Mission Engineering Competencies**. 2018. Citations on pages 2, 3, 4, 23, 41, and 54.

VIERHAUSER, M.; RABISER, R.; GRUNBACHER, P.; AUMAYR, N. A requirements monitoring model for systems of systems. In: **IEEE 23rd International Requirements Engineering Conference (RE 2015)**. Ottawa, Canada: IEEE, 2016. p. 96–104. Citations on pages 125 and 126.

VILLALBA, M. T.; FERNáNDEZ-SANZ, L.; MARTíNEZ, J. J. Empirical support for the generation of domain-oriented quality models. **IET Software**, The Institution of Engineering and Technology, v. 4, n. 1, p. 1–14, 2010. Citation on page 139.

WAGNER, L. G.; COFER D. COLLINS, K. S. R.; RAPIDS, C.; IOWA. **Formal Methods Tools Qualification**. 2017. Citation on page 32.

WALDEN, D. D. The changing role of the systems engineer in a system of systems (sos) environment. In: **1st Annual IEEE International Systems Conference (SysCon 2007)**. Honolulu, Hawai: IEEE, 2007. p. 1–6. Citation on page 128.

WALDEN, D. D.; ROEDLER, G. J.; FORBERG, K. J.; HAMELIN, D. R.; SHORTELL, T. M. **INCOSE systems engineering handbook: a guide for systems life cycle processes ans activities**. 2015. Citations on pages 21, 22, and 132.

WALKER, R. G. **A method to define requirements for System-of-Systems**. Master's Thesis (PhD Thesis) — Faculty of Old Dominion University, USA, 2014. Citations on pages 24 and 25.

WANG, Q.; WANG, Z.; ZHANG, T.; ZHU, W. A quality requirements model and verification approach for system of systems based on description logic. **Frontiers of Information Technology & Electronic Engineering**, Zhejiang University and Springer-Verlag Berlin Heidelberg, p. 1–17, 2016. Citations on pages 3, 26, and 27.

WANG, Q.-L.; WANG, Z.-X.; ZHANG, T.-T.; ZHU, W.-X. A quality requirements model and verification approach for system of systems based on description logic. **Frontiers of Information Technology & Electronic Engineering**, Springer Link, v. 18, n. 3, p. 346–361, 2017. Citations on pages 1, 28, and 32.

WERMELINGER, M. A. **Specification of Software Architecture Reconfiguration**. Master's Thesis (PhD Thesis) — Science and Technology Faculty, University of Nova Lisboa, Lisboa, Portugal, 1999. Citation on page 76.

WERTZ, J. R.; EVERETT, D. F.; PUSCHELL, J. J. **Space Mission Engineering: The New Smad**. 3. ed. Hawthorne, CA: Microcosm Press, 2015. Citations on pages 23, 40, 93, and 194.

WESTHEALTH, I. **Introduction to Interoperability and Decision-Maker's Interoperability Checklist**. 2010. Citation on page 37.

WIEGERS, K.; BEATTY, J. **Software Requirements**. : Pearson Education, 2013. Citation on page 33.

WIND, R. Y. J.; MAHAJAN, V. **Convergence Marketing: Strategies for Reaching the New Hybrid Consumer**. : Financial Times Print-Hall, 2001. Citation on page 113.

WODTKE, C. **Introduction to OKRs**. : O'Reilly Media, 2016. Citation on page 106.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **18th International Conference on Evaluation and Assessment in Software Engineering (EASE 2014)**. London, UK: ACM, 2014. p. 38:1–38:10. Citations on pages 7, 51, and 208.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **EASE**. 2014. p. 38:1–38:10. Citations on pages 54 and 223.

WOHLIN, C.; RUNESON, P.; HST, M.; OHLSSON, M. C.; REGNELL, B. j.; WESSLN, A. **Experimentation in software engineering**. : Springer Publishing Company, Incorporated, 2012. Citation on page 157.

WOHLIN, C.; SMITE, D.; MOE, N. B. A general theory of software engineering. **Journal System Software**, Elsevier Science Inc., v. 109, n. C, p. 229–242, 2015. Citation on page 44.

WüRFEL, D.; LUTZ, R.; DIEHL, S. Grounded requirements engineering. **Journal System Software**, Elsevier Science Inc., v. 117, n. C, p. 645–657, 2016. Citations on pages 43 and 148.

WYATT, E. A. J. **A Reliability-based Measurement of Interoperability for Conceptual-Level Systems of Systems**. Master's Thesis (Thesis) — Georgia Institute of Technology, Atlanta, USA, 2014. Citation on page 37.

YIN, R. K. **Case Study Research and Applications: design and methods**. 6. ed. : SAGE Publications, Inc., 2018. Citation on page 44.

YU, E. **Modelling strategic relationships for process reengineering. Computer Science**. Master's Thesis (PhD Thesis) — Computer Science, University of Toronto, Toronto, Canada, 1995. Citations on pages 34 and 124.

YU, E. **iStar an agent-and goal-oriented modelling framework**. 2011. Available at <http://www.cs.toronto.edu/km/istar/>. [Online, Acessed: August 18, 2018]. Citations on pages 31 and 34.

YU, E.; GUNTER, D. A.; FRANCH, X.; CASTRO, J. Practical applications of i* in industry: the state of the art. In: **Requirements Engineering@Brazil (RE 2013)**. Rio de Janeiro, Brazil: PUC-RIO, 2013. Citation on page 34.

YU, E.; MYLOPOULOS, J. **Why Goal-Oriented Requirements Engineering**. 2017. Available at <http://www.cs.toronto.edu/pub/eric/REFSQ98.html>. [Online, Acessed: January 03, 2017]. Citation on page 13.

YUAN, L.; TANG, T.; LI, K. Modelling and verification of the system requirement specification of train control system using sdl. In: **10th International Symposium on Autonomous Decentralized Systems (ISADS 2011)**. Washington, USA: IEEE Computer Society, 2011. p. 81–85. Citations on pages 28 and 31.

ZEIGLER, B. P.; NUTARO, J. J. Towards a framework for more robust validation and verification of simulation models for systems of systems. **Journal of Defense Modeling and Simulation: Applications, Methodology, Technology**, IEEE Software, v. 13, n. 1, p. 3–16, 2015. Citation on page 32.

ZHANG, L. Aspect-oriented formal techniques of cyber physicalsystems. **Journal of Software**, Academy Publisher, v. 7, n. 4, p. 823–834, 2012. Citations on pages 31 and 54.

ZHANG, L. Modeling large scale complex cyber physical control systems based on system of systems engineering approach. In: **20th International Conference on Automation and Computing (ICAC 2014)**. Cranfield, United Kingdom: IEEE, 2014. p. 55–60. Citations on pages 27, 30, and 31.

ZHU, W.; HE, H.; WANG, Z. Ontology-based mission modeling and analysis for system of systems. In: **IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)**. Exeter, UK: IEEE, 2017. p. 538–544. Citations on pages 40, 125, 126, 135, and 136.

ZOU, L.; LV, J.; WANG, S.; ZHAN, N.; TANG, T.; YUAN, L.; LIU, Y. Verifying chinese train control system under a combined scenario by theorem proving. In: **Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2014)**. Menlo Park, USA: Springer-Verlag New York, Inc., 2014. p. 262–280. Citation on page 31.

ZOWGHI, D.; COULIN, C. Engineering and managing software requirements. In: AURUM, A.; WOHLIN, C. (Ed.). Berlin, Germany: Springer Berlin Heidelberg, 2005. chap. Requirements elicitation: a survey of techniques, approaches, and tools. Citation on page 13.

Glossary

Architecting provides the strategies and modeling approaches to ensure we spend time developing the possible "could be" states, and evaluating the best alternative given a set of desired properties and criteria for the future System.

Architectural Configuration It determines the way in which architectural elements are interconnected to form the structure of a software architecture. It can be also referred to as topology.

Artifact One of outcomes produced during the development of software. Some artifacts (e.g., use cases, class diagrams, and other UML models, requirements and design documents) help describe the function, architecture, and design of software. Other artifacts are concerned with the process of development itself - such as project plans, business cases, and risk assessments..

Capability It is the ability to achieve the desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks (T) (BALDWIN, 2008).

Capability Objectives They are sometimes general statement(s) of one capability mapped in the desired operational task or mission that evolve over time (LANE, 2017).

Coalition interoperability represent the ways in which constituents and mediators can be arranged to form the SoS.

Constituent Mission The parent mission of a constituent is a set of objectives, which are required to be achieved by the constituent systems to accomplish the requirements of its normal functioning..

Constraint It is a statement that may constrain the design of a software architecture. A constraint may be related to design decisions, stakeholder requirements, technical and/or business concerns, etc.

Information Technology refers to technologies that information processing uses hardware, software, communication infrastructure, and related services to connect application and data exchanged..

Interoperability Requirement a statement that specifies a function, ability or characteristics, related to the ability of a partners to ensure its partnership in terms of compatibility, interoperation, autonomy, and reversibility (MALLEK *et al.*, 2012).

Methodological Triangulation involves using more than one method to gather data, such as interviews, observations, questionnaires, and documents..

Mission an activity composed of tasks, together with the purpose, that clearly reveal the action to be taken and the rationale, as it is considered in the context of this thesis (LARSON; WERTZ, 1991; BEALE *et al.*, 2004; WERTZ *et al.*, 2015; LEGGETT, 2017; STAFF, 2019).

Mission Objectives they are statement(s) responsible for quantifying mission goal(s), constraint(s) and services of the SoS as measurable data or properties related to space, time or effectiveness (FAISANDIER, 2015; NASA, 2018).

Operational Technology refers to use hardware and software that detects or causes a change through the direct monitoring and/or control of physical devices, process and events in the enterprise..

Purpose expresses the relevance of the SoS in its context of use. It presents the final aim of the SoS in its environment (FAISANDIER, 2015).

Specification A document that specifies, ideally in a complete, precise and verifiable manner, the requirements, design, behavior, or other characteristics of a component or system, and, often, the procedures for determining whether these provisions have been satisfied (IEEE 610, 1990).

Strategic Goal the broad statements describing where the enterprise wants to be in the future in a qualitative way..

Strategy to defines actions to achieve the objectives, besides mobilizing technical and non-technical resources to execute these actions..

System is usually understood as a set of elements (e.g., software, hardware, processes, human, and data) that act together to generate behavior or function not available from any elements individually (LEVESON, 2013).

Systems-of-Systems is a concept which represent an organization of the heterogeneous and independent constituents, which have their own mission (i.e., individual mission) and that interoperate to carry out a greater mission, also Known as global mission..

Technical Interoperability Requirements a statement that specified a infrastructure resource, characteristics, ability or constraints of communication with data exchange regards to the ability of a collaborator to ensure its collaboration in terms of composability, compatibility, integrability, interoperation, autonomy, variability, transparency, resilience, sustainability, reversibility..

Trade-off Decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders (ISO/IEC/IEEE-29148, 2011).

APPENDIX

A

Development process of the substantive theory

Figure 43 shows the process of creation of the substantive theory developed in this research, called *ATLANTA Theoretical Framework*. It displays as each systematic mapping contributed to code and the empirical data were analyzed achieving the theoretical saturation, and the contribution of the systematic mappings, experts, and feedback for improving such theory.

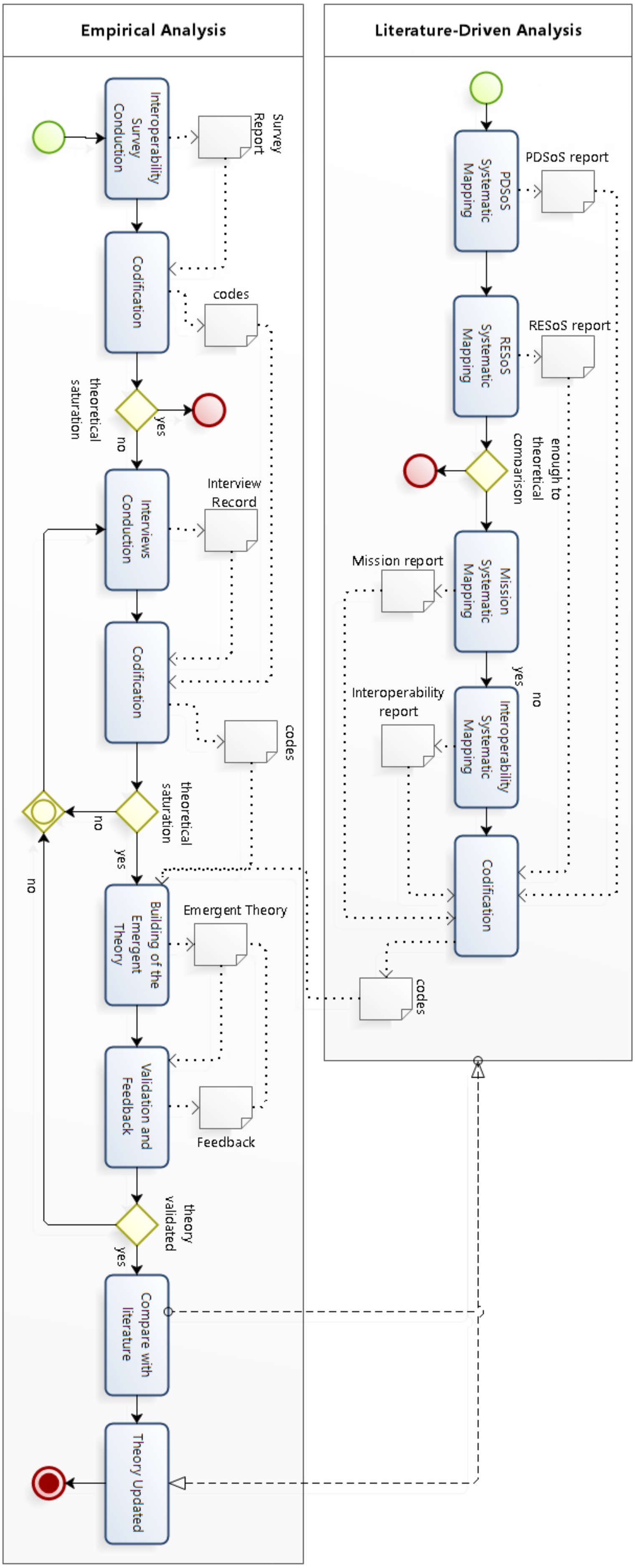


Figure 43 – Substantive theory development process

Source: Elaborated by the author.

APPENDIX

B

System Engineering and Systems-of-Systems Comparison

Table 19 suggests that SoSE addresses problem domains that are aimed at modifying our traditional way of thinking about the complex systems constructions (e.g., SoS), as well as the way of thinking about its requirements (KEATING *et al.*, 2015). Due to the increasing complexity of SoS their problems have become more emergent and within of context highly critical. Traditional systems have only one mission to achieve throughout its development lifecycle, whereas SoSE has one purpose with multiples missions and goals. Indeed, these systems have organized in collaboration to form more complex SoS and currently, the own SoS also are becoming a part constituent of others SoS, elevating still more its complexity.

Table 19 – Comparison between drivers of systems engineering and SoS engineering

Source: Adapted from Keating *et al.* (2003), Gorod *et al.* (2008).

Area	**Systems Engineering**	**SoS Engineering**
Focus	Single complex system	Multiple integrated complex systems
Nature	Technical	Socio-technical
Objective	Optimization	Satisficing
Approach	Process	Methodology
Expectation	Solution	Initial response
Problem	Well defined	Emergent
Analysis	Technical dominance	Contextual influence criticality
Goals	Unitary	Pluralistic
Boundary	Fixed and defined (i.e., static)	Fluid and ambiguous (i.e., dynamic)
Structure	Hierarchical	Network
Timeframe	System Life Cycle	Continuous
Centricity	Platform	Network
Tools	Many	Few
Management Framework	Established	Various

In other perspective, Table 20 shows the relation between the attributes of the development lifecycle of traditional systems and of an SoS. For example, the traditional systems have clear stakeholders while in SoS context their are multiples and many time unknown during part of the development of SoS. Besides that, they often change to SoS domain leading to multiples and contradictory mission objectives. However, in the traditional system, the modification of stakeholders is smaller and the objectives of systems are clearly supporting the requirements and design process. Both, systems and SoS have changes after in operation. Change management to traditional level is well-structured, in SoS this does not occur because is not easy to make change management when they occur at runtime. Monitoring and control are necessary to continue verification of failures of adaptation to first perform this changes when necessary, still there exists the external changes that are implemented at design time and other that emerge of the combination of constituents, called emergent behavior. At least, we have ownership of the systems that are clear and easy to identify to traditional whereas in SoS usually the owners are set of organization when are SoS with central control; otherwise, they do not have any ownership.

Table 20 – Comparison between characteristics of systems engineering and SoS engineering

Source: Adapted from Maier (1998).

Description	Systems tend to have	SoS tend to have
Stakeholders	A clear set of stakeholders	Multiple levels of stakeholders with mixed and possibly competing interests
Objectives	Clear objectives and purpose	Multiple and possibly contradictory objectives and purpose
Management	operational priorities, with escalation to resolve priorities	Multiple, and sometimes different, operational priorities with no clear escalation routes
Lifecycle	A single life cycle	Multiple lifecycles with elements being implemented asynchronously
Ownership	Clear ownership with the ability to move resources between elements	Multiple owners making individual resourcing decisions

While the two other tables addressed issues regards to the development of the systems, Table 21 discusses the properties of them. In general, the properties of the traditional systems are involved with their parts, i.e., components that together form the whole systems. On the other hand, SoS properties are related to constituents, which usually are the traditional system that collaborates with each other to form the whole SoS and achieve the emergence of SoS. However, the traditional systems maintain the autonomy, the ability to choose and connectivity. In fact, many differences are identified in the literature to both, traditional and SoS system and many others still are being identified and described such as adaptability, open-ended architecture, resilience, etc.

Table 21 – SoS distinguishing characteristics

Source: Adapted from Boardman and Sauser (2006).

Attribute	System	SoS
Autonomy	Autonomy is ceded by parts to grant autonomy to the system.	Autonomy is exercised by CS in order to fulfill the purpose of the SoS
Belonging	Parts are akin to family members; they did not choose themselves but came from parents. Belonging of parts is in their nature.	While some CS are directed or coerced to belong to SoS, some CS may be unaware of the SoS. Some CS choose to belong on a cost/benefits basis; also, to cause greater fulfillment of their own purposes, and because of belief in the SoS supra purpose.
Connectivity	Prescient design, along with parts, with high connectivity hidden in elements, and minimum connectivity among major subsystems.	Dynamically supplied by CS with every possibility of myriad connections between constituent systems, possibly via a net-centric architecture, to enhance SoS capability.
Diversity	Managed i.e. reduced or minimized by modular hierarchy; parts' diversity encapsulated to create a known discrete module whose nature is to project simplicity into the next level of the hierarchy.	Increased diversity in SoS capability achieved by released autonomy, committed belonging, and open connectivity.
Emergence	Foreseen, both good and bad behavior, and designed in or tested out as appropriate.	Enhanced by deliberately not being foreseen, though its crucial importance is, and by creating an emergence capability climate, that will support early detection and elimination of bad behaviors.

APPENDIX

C

Grounded Theory Process

Table 22 shows the original grounded process proposal by Pandit (1996), which adapted and extended to be applied in Chapter 3.

Table 22 – Pandit's grounded theory process

Source: Adapted from Pandit (1996).

PHASES	ACTIVITIES
PHASE 1 - RESEARCH DESIGN	
Step 1 - Review of technical literature	Definition of research question
Step 2 - Selecting cases	Theoretical, not random, sampling
PHASE 2 -DATA COLLECTION	
Step 3 - Develop rigorous data collection protocol	Employ multiple data collection methods
Step 4 - Entering the field	Overlap data collection and analysis, using flexible data collection methods
PHASE 3 - DATA ORDERING	
Step 5 - Data ordering	Arraying collected data chronologically
PHASE 4 -DATA ANALYSIS	
Step 6 - Analysing data relating to the first case	Unify coding (open, axial, and selected coding)
Step 7 - Theoretical sampling	Theoretical replication across cases (go to step 2 until theoretical saturation)
Step 8 - Reaching closure	Theoretical saturation when possible
PHASE 5 - LITERATURE COMPARISON	
Step 9 - Compare emergent theory with extant literature	Comparisons with similar and conflicting frameworks

APPENDIX

D

PDSoS - Systemic Mapping Protocols

Initial Remarks

Our mapping was conducted in the context of software development processes for SoS. This mapping was carried out from April/2015 to October/2015 and updated from September/2019 to December/2018 and involved six researchers, among them four experts. We used the guidelines established by Kitchenham and Charters (2007) that is divided in three phases: planning, conducting, and reporting In the first phase, the research objectives and the systematic review protocol are defined. In the second phase, the primary studies are identifying, selecting, and evaluating second to the inclusion and exclusion criteria defined in the planning phase. For each selected studies, data are extracted and synthesized. In the third phase, a final report is organized, presented and discussed. The next sections are presented present these three phases in details.

Phase I - Planning

In this phase, the SM protocol was established, including the research questions, search strategy, selection criteria, quality assessment criteria and data extraction and synthesis method. The SM protocol was elaborated by the author and validated by the experts.

Research Question

In order to structure and simplify the research questions development, we used the PICO model (KITCHENHAM; CHARTERS, 2007), which includes the consideration of four elements: Population, Intervention, Comparison and Outcomes. In the Population is described the population group that is observed by the intervention; In the Intervention identifies what will be investigated this SM; Comparison is identified what will be used to compare in the context of the SM; and Outcomes are the expected effects of the proposed intervention. To this SM the

Population comprises the developers and researchers of SoS and systems engineering community. *Intervention* for our SM is related to the context of SoS development processes. *Comparison* it not applicable. Lastly, in *Outcomes* characterize all primary studies identified in the literature that report SoS development processes. Furthermore, we hope to identify SoS characteristics, type of empirical evaluation, technical factors, activities and/or artifacts addressed or mentioned by the processes, factors that can influence the development of the SoS, and trends and challenges discussed by the studies. The research questions are:

RQ-01 Have SoS been constructed based on development processes? Which are those processes?
This RQ aims at identifying processes, but not excluding other initiatives (e.g., methods, frameworks, and models) proposed for building SoS.

RQ-02 Which SoS characteristics have been considered in SoS development processes?
We aim at investigate the main characteristics mentioned by the authors based on the five Maier's characteristics (MAIER, 1998).

RQ-03 Which technical factors and management activities were addressed or mentioned in the SoS development processes?
This RQ, the objective is to identify the technical factors (e.g., communication, cost dependence, risk management) that were used or influenced to build of the SoS development processes.

RQ-04 Which main challenges and trends have been discussed by the authors of the primary studies?
This RQ aims at identifying the main challenges and trends of SoS development processes discussed by the authors of the primary studies.

RQ-05 How have been established the SoS development processes?
The objective is to identify approaches used to building SoS development processes and to verify which technologies help in the development those processes (e.g, SOA, MDA, MDD).

RQ-06 What do have influenced the development of the SoS?
This RQ aims at identify the factors that influence (e.g., constituents architecture, several life cycle) the interoperability in SoS development processes.

Search Strategy

In order to determine the search strategy of selection of the primary studies we choose four search sources: publication database, related works, snowballing and expert opinion. The four search sources are presented as follows:

- **Publication database:**

  For identifying primary studies is necessary to define which publication database will be adopted and the search string that will be used on these publication databases. At these SM, we adopted the publication database ACM Digital Library, IEEE Xplore, Science Direct, Scopus, and ISI Web of Science. They were chosen to be recognized and widely used in Software Engineering, as example: (PERTENSEN *et al.*, 2015) (KITCHENHAM *et al.*, 2010) (MADEYSKI *et al.*, 2014) (MUNIR *et al.*, 2014) (SILVA *et al.*, 2015c) (MACHADO *et al.*, 2013) (NARCISO *et al.*, 2014) (GUESSI *et al.*, 2015).

  To define the search string, five steps were taken: (i) search by synonyms of "development process" in systematic literature reviews of traditional software development processes; (ii) search in dictionaries; (iii) consulting experts; (iv) pilot search; and (v) use of control studies (CALINESCU; KWIATKOWSKA, 2008a) (CARBON *et al.*, 2008a) (FARROHA; FARROHA, 2011a) (RAMOS *et al.*, 2012a).

  The pilot research showed that the term "development technique" and the plural of the synonyms of "development process" present in the search string not influenced the total number of primary studies returned by the engine search. Furthermore, it was noted that 50% of control studies were not returning at these search, which could be a failure of the search string. An analysis and modifications were done together with experts achieving a higher return than 231%[1] of primary studies when the search string returned all control studies.

  The search string was designed to cover variations and synonyms for terms related to "development process" and "systems-of-systems" and adapted for meeting particularities of each engine search. The search string used in our SM is presented in Table 23. The first column represents the research areas and the second column the corresponding terms.

Table 23 – Search string

Area	**Terms**
Development Process	("Development Method" OR "Development Approach" OR "Development Design" OR "Design Flow" OR "Development Procedure" OR "Development Life Cycle" OR "Agile Development" OR "Software Product Line")
	AND
Systems-of-Systems	("System of Systems" OR "System-of-Systems" OR "Systems-of-Systems" OR "Systems of Systems")

- **related works:**

[1] the search string of the pilot research returned 183 primary studies, after the modifications the search string returned the total of 423 ones.

It was also considered the studies cited as related works in the relevant primary studies found in the engine search;

- **snowballing:**

  To increase the accuracy of our SM was used the snowballing. The snowballing refers to using the reference list of a paper or the citations to the paper to identify additional papers (WOHLIN, 2014a). If using the reference is called of backward snowballing, on the other hand, using the citation is known as forward snowballing. The steps following in our SM for snowballing procedure is described in (WOHLIN, 2014a).

- **expert opinion:**

  Studies suggested by experts in SoS domain also was considered. Although the indication of studies by expert can be considered as bias, we have adopted this source aiming to not lose any important evidence.

Selection criteria

In order to accordingly select the studies to answer our research questions, we established the Inclusion (IC) and Exclusion Criteria (EC), which it was used in the first and second selection. In the first selection, the application of the criteria were limited to primary studies metadata (i.e., title, abstract, and keywords). At the second selection, the criteria were also considered in reading full study. In the case of any discordance in the first selection, the studies were included and analyzing in the second selection. In the second selection, in case of discordance, the study was discussed together with experts until reaching an agreement. The IC and EC are:

Inclusion Criteria

IC.1: The study discusses SoS development process.
IC.2: The study addresses SoS characteristics, problems or activities related to SoS development processes.

Exclusion criteria

EC.1: The study is not related to SoS.
EC.2: The study does not discuss any SoS development process.
EC.3: Development process is not the main focus of the study.
EC.4: The study is an editorial, keynote, opinion, tutorial, poster or panel.
EC.5: The study is duplicated or there is newer or a more complete one about the same research.
EC.6: The study compiles results of other studies.
EC.7: The study is not written in English.
EC.8: The full text of the study is not available.

Quality Assessment Criteria

In order to evaluate the primary studies were adapted the criteria described in (KHAN *et al.*, 2001) (KITCHENHAM; CHARTERS, 2007) (KITCHENHAM *et al.*, 2010). The quality criteria and the specification scores for the SM are shown in Table 24. The possible score depends of each quality criteria. When the primary study fully matches a quality criterion, it had 2 as score and 0 otherwise. In some cases, the quality criterion is partially met by the primary studies and in those case, intermediate scores were defined, which can range from 0.5 to 1.5.

The final score of the quality assessment (QA) for each primary study was calculated using the following formula:

$$QA = \frac{\sum_{8}^{i=1} CQ\_i}{\text{maximum score}} * 100$$

Table 24 – Quality assessment criteria

Source: Adapted from Kitchenham *et al.* (2010).

Quality Criteria (QC)	Score
QC1: Do the authors clearly state the aims of the research?	0 - No 1 - Partially 2 - Totally clear
QC2: Do the authors discuss the limitations of their study?	0 - No 1 - Partially 2 - Clearly discussed
QC3: Do the authors state the findings clearly ?	0 - No 1 - Clearly partially 2 - Totally clear
QC4: Is there evidence that the findings of the study can be used by other researchers/practitioners?	0 - No 1 - Partially reusable 2 - Totally reusable
QC5: Does the study body fully meet issues provided in its abstract?	0 - No 1 - Attend partially 2 - Attend completely
QC6: Do the described methods were empirically evaluated?	0 - No 1 - Partially evaluated 2 - Totally evaluated
QC7: How the empirical evaluation conducted?	0 - No 0.5 - Illustrative example 1 - Case study 1.5 - Researchers/Experts 2 - Use in industry
QC8: Are there any ideas for further investigation presented?	0 - No 2 - Yes

Data Extraction and Synthesis Method

The data extraction is designed to gather the information necessary to answer the research questions and evaluate the quality of the study (MAGDALENO *et al.*, 2012). Thus, for each primary study included, data were extracted and discussed with the experts. To extraction of the data we used the Start Tool[2] (FABBRI *et al.*, 2012), Microsoft office Excel 2013 and considered three categories of data: (i) characterization of studies (e.g., author, year, and publication vehicle); (ii) infrastructure used in development of approaches (e.g., technologies, tools, and languages); and (iii) general features of the studies (e.g., SoS characteristics, technical factors, and processes) that enable us record details of the studies, organizing the synthesis in table, graphic or graphs.

To share the activities of the research protocol and the steps followed by the reviewers, it becomes possible to audits and facilitates the process of a new conduction, addressing issues related.

[2] <http://lapes.dc.ufscar.br/tools/starttool>

APPENDIX

E

Selected Primary Studies of Systematic Mapping

Table 25 presents the 45 studies selected in this systematic mapping, their ID, author name (s), publication title as an external link to where the study was published, and publication year.

Table 25 – List of included primary studies from the SoS developmento mapping

Source: Elaborated by the author.

Study ID	Included Study	Year
S0	Raymer (RAYMER, 1992)	1992
S1	Rossak et al. (ROSSAK *et al.*, 1994)	1994
S2	Ministry of Defence (PARTNERS, 2005)	2005
S3	Mostermanm et al. (MOSTERMANM *et al.*, 2005)	2005
S4	Boehm et al.(BOEHM; LANE, 2006)	2006
S5	Biltgen (BILTGEN, 2007)	2007
S6	Shams et al. (SHAMS *et al.*, 2008)	2008
S7	Gokhale et al. (GOKHALE *et al.*, 2008b)	2008
S8	Carbon et al. (CARBON *et al.*, 2008b)	2008
S9	Butterfield et al. (BUTTERFIELD *et al.*, 2008)	2008
S10	Reichle et al. (REICHLE *et al.*, 2008)	2008
S11	Kewley et al. (JR.; TOLK, 2009)	2009
S12	Ballagny et al. (BALLAGNY *et al.*, 2009)	2009
S13	Farcas et al. (FARCAS *et al.*, 2010)	2010
S14	Decher (DECHER, 2010b)	2010
S15	Calinescu et al. (CALINESCU; KWIATKOWSKA, 2008b)	2010
S16	Serugendo et al. (SERUGENDO *et al.*, 2010)	2010
S17	Department of Defense (OFFICER, 2010)	2010
S18	Tu et al. (TU *et al.*, 2011)	2011
S19	Farroha et al. (FARROHA; FARROHA, 2011b)	2011
S20	Dahmann et al. (DAHMANN *et al.*, 2011)	2011
S21	Mazzuchi et al. (MAZZUCHI; ALBAKRI, 2011)	2011
S22	Mensing et al. (MENSING *et al.*, 2012)	2012
S23	Li and Yang (LI; YANG, 2012)	2012
S24	Hallsteinsen et al. (HALLSTEINSEN *et al.*, 2012b)	2012
S25	Santamaria et al. (SANTAMARÍA *et al.*, 2012)	2012
S26	Ramos et al. (RAMOS *et al.*, 2012b)	2012
S27	Braga et al. (BRAGA *et al.*, 2012)	2012
S28	Taweel et al.(TAWEEL *et al.*, 2013)	2013
S29	Acheson et al. (ACHESON *et al.*, 2013)	2013
S30	Luna et al. (LUNA *et al.*, 2013a)	2013
S31	Madni and Sievers (MADNI; SIEVERS, 2014b)	2014
S32	Louali et al. (LOUALI *et al.*, 2014)	2014
S33	Sanduka et al. (SANDUKA; OBERMAISSER, 2014)	2014
S34	Ricci et al. (RICCI *et al.*, 2014)	2014
S35	Ergin et al. (KILICAY-ERGIN; DAGLI, 2015)	2015
S36	Barker and Summers (BARKER; SUMMERS, 2015)	2015
S37	Hu et al. (HU *et al.*, 2015)	2015
S38	Sharples (SHARPLES, 2015)	2015
S39	Diaz et al. (DIAZ *et al.*, 2016)	2016
S40	Naik (NAIK, 2016)	2016
S41	Raz et al. (RAZ *et al.*, 2016)	2016
S42	Sturdivant and Chong (STURDIVANT; CHONG, 2017)	2017
S43	Albers et al. (ALBERS *et al.*, 2018)	2018
S44	Loureiro et al. (LOUREIRO *et al.*, 2018)	2018
S45	Lu et al. (LU *et al.*, 2018)	2018

APPENDIX

F

SMRE-SoS - Systemic Mapping Protocols

Systematic Mapping Protocol

Formal and Semi-formal Modeling Languages and Techniques

Cristiane Aparecida Lana, Milena Guessi, Pablo Oliveira Antonino, Dieter Rombach, and Elisa Yumi Nakagawa

The Systematic Mapping (SM) protocol includes the objective, research questions, search strategy, selection criteria, procedures for studies selection, and data extraction and synthesis methods. The SM protocol was elaborated by the authors of this article and validated by experts in Systems-of-Systems (SoS) and requirements engineering. To conduct this mapping, we followed the process proposed by Petersen et al. [1] and Kitchenham and Charters' guidelines [2].

1. Research question and objective

The overall research objective of this study is to identify studies that address formal and semi-formal languages and techniques for requirements modeling that have been or could be applied to SoS. In order to structure this objective, we developed our research question using the PICO model [2], which considers four elements: Population, Intervention, Comparison, and Outcomes. In the Population, it is described the population group that is observed by the intervention; in the Intervention, it is identified what will be investigated in this SM; Comparison identifies what will be used to compare in the context of the SM; and Outcomes are the expected effects of the proposed intervention.
To this SM, the **Population** comprises the systems engineering community and specifically the requirements engineering community, which has researched requirements to develop SoS. **Intervention** for our SM is related to studies that address support (e.g., languages and techniques) to formally or semi-formally model requirements of SoS or similar systems (e.g., autonomous systems, dynamic systems). **Comparison** is not applicable in our SM. Lastly, **Outcomes** describe all primary studies identified in the literature that report what is adopted to formally and semi-formally model requirements of SoS. The research question formulated to be answered by this SM is:

RQ: Which formal and semi-formal languages and techniques have been used for modeling SoS requirements?

2. Search Strategy

To define the search strategy for primary studies selection, we chose three search sources: (i) publication databases using search string; (ii) manual search; and (iii) expert opinion, which are presented as follows:

- Publication databases and search string: To identify primary studies, it is necessary to define the publication databases that will be adopted and the search string that will be used on these databases. To select the databases for our SM, we considered the criteria addressed and discussed in Dieste et al. [3] and Kitchenham and Charters [2]. Moreover, they are the most relevant database to computer science area, and widely used and recognized in Software Engineering, such as in [4] [5] [6] [7] [8]. Table 1 lists the databases adopted in this mapping.

Table 1: Search Database

Digital libraries	Web site
ACM Digital Library	http://dl.acm.org
IEEE Xplore	http://ieeexplore.ieee.org
Science Direct	http://sciencedirect.com
Scopus	http://scopus.com
ISI Web of Science	http://webofscience.com
Springer Link	https://link.springer.com

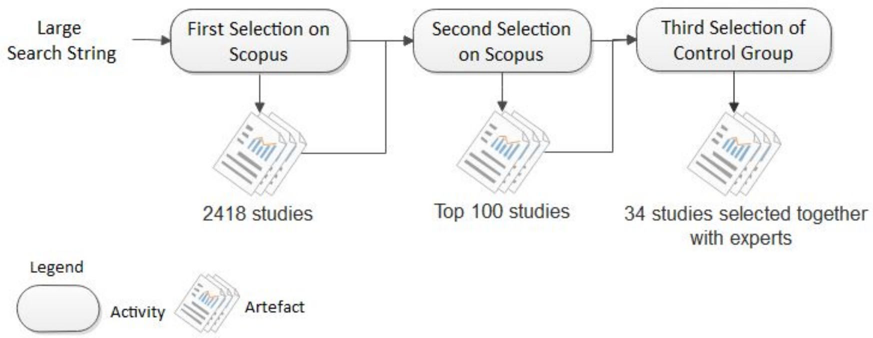
To establish the search strategy and answer our RQ, we identified two main keywords: “requirement modeling” and “formal”. We also identified terms related to these keywords that were evaluated by experts. Subsequently, we conduced iterative steps to identify other related terms, summarized as follow:

- **Search for synonyms of keywords in other SM or** Systematic Literature Review (SLR)**:** In the first step, we searched synonymous through of SM or SLR in main publication venues of the areas related with these keywords (e.g., RE conference, Journal of Computer Science (JCS), and ACM TOSEM);
- **Search for synonyms in dictionaries:** We also evaluated the synonymous of the keywords using dictionaries, such as Oxford e Cambridge as well as synonymous online bases, as Wordnet and Thesaurus;
- **Experts opinion:** The search string was evaluated three times by two group of experts, who supported us in evaluation of new terms, as “quality”' and “non functional”, besides excluding terms that did not influence the obtainment of studies from the publication databases (see, Table 2). Moreover, in the third evaluation, the experts also suggested to adopt a control group, based on number of citations of the studies that could be included in this SM. For this, we applied the process summarized in Figure 1.

Table 2: Keywords that did not influence return of primary studies

Keywords	Synonyms
Modeling	"modelling requirements" OR "modelling of requirements" OR "requirements modelling" OR "requirements molding" OR "requirements molding" OR "requirements representation" OR "requirements representation" OR requirements analyze" OR "requirements validation" OR
Validation	"requirements validating" OR "requirements valid" OR "requirement validate" OR "requirements validate" OR "requirements validation" OR "requirement validations" OR "requirement validations" OR "requirements validated" OR

Figure 1: Process to define the control group



- **Pilot search**: Initially, we used the pilot search to refine our search string and verify the influence of the keywords when applied in the publication databases. In this case, it was possible to observe that plural of the keywords and terms did not modify the final result returned from the databases. On the second moment, the pilot was used to validate our string with the control group as presented in Table 3; and
- **Manual search**: Searches in databases were complemented with manual searches in publication venues that were not indexed by none database used in this mapping. To identify the events (conferences and workshops) that were not indexed by the databases adopted by this SM, we analyzed the 1,702 events in the area of Science Computer area. 27 events were identified as potential ones to accomplish manual search. Of these events, only two (i.e., Brazilian Symposium of Software Quality (SBQS) and Engineering Requirement Workshop (WER)) are not indexed by some database and, therefore, studies published in these events were manually searched

The search string final used by this SM is:

(“requirement modeling” OR “model of requirement” OR “requirement model” OR “modeling requirement” OR “modeling of requirement” OR “requirement representation” OR “requirement analysis” OR “analysis of requirement” OR “requirement analyzing” OR “requirement design” OR “requirement verification” OR “verification of requirement” OR “evolution of requirement” OR “requirement evolution” OR “requirement validation” OR “validating requirement” OR “validation of requirement” OR “requirement specification” OR “quality requirement” OR “non functional requirement” OR “non-functional requirement” OR “nonfunctional requirement” OR “non functional property” OR “non-functional property” OR “nonfunctional property” OR “non functional characteristic” OR “non-functional characteristic” OR “nonfunctional characteristic” OR “quality attribute” OR “quality characteristic” OR “quality factor” OR “quality criterion”) AND (formal))

Table 3: Control studies

ID	Title	Year	Citation
1	Goal-directed,requirements acquisition	1993	1064
2	Handling,obstacles in goal-oriented requirements engineering	2000	345
3	Specification Matching of Software Components	1997	335
4	Automated Consistency Checking of Requirements Specifications	1996	325
5	Requirements,Specification for Process-Control Systems	1994	248
6	Using,model checking to generate tests from requirements specifications	1999	227
7	Model,checking large software specifications	1998	176
8	Reasoning,about partial goal satisfaction for requirements and design engineering	2004	140
9	On,Formalism in Specification	1985	131
10	Goal-oriented,requirements analysis and reasoning in the Tropos methodology	2005	115
11	Using,Abstraction and Model Checking to Detect Safety Violations in Requirements,Specifications	1998	114
12	Formal,approach to scenario analysis	1994	109
13	Model,checking early requirements specifications in Tropos	2001	105
14	An,Operational Approach to Requirements Specification for Embedded Systems	1982	104
15	From,system goals to software architecture	2003	101
16	SOFL: A,formal engineering methodology for industrial applications	1998	74
17	On,formal requirements modeling languages: RML revisited	1994	74
18	RELAX: A,language to address uncertainty in self-adaptive systems requirement	2010	72
19	SCR*: a,toolset for specifying and analyzing requirements	1995	65
20	Techne:,Towards a new generation of requirements modeling languages with goals,,preferences, and inconsistency handling	2010	64
21	Informal,and Formal Requirements Specification Languages: Bridging the Gap	1991	64
22	From,goals to aspects: Discovering aspects from requirements goal models	2004	61
23	A formal,approach to adaptive software: Continuous assurance of non-functional,requirements	2012	54
24	Experiences,using lightweight formal methods for requirements modeling	1998	47
25	Lightweight,validation of natural language requirements	2002	47
26	Deriving,event-based transition systems from goal-oriented requirements models	2008	41
27	Formal,approach to specifications in conceptual design	1992	40
28	Facilitating,'fuzzy to formal' requirements modelling	1994	40
29	Semantic,parameterization: A process for modeling domain descriptions	2008	37
30	A,methodology for mapping SysML activity diagram to time petri net for,requirement validation of embedded real-time systems with energy constraints	2009	37
31	Logical,framework for modeling and reasoning about the evolution of requirements	1997	34
32	Semantics,of trace relations in requirements models for consistency checking and,inferencing	2011	33
33	From,English to formal specifications	1994	32
34	Proving,the shalls: Early validation of requirements through formal methods	2006	31

3. Selected Criteria

The selection criteria (inclusion and exclusion criteria) were used to evaluate primary studies retrieved from the publication databases and were applied in the first and second selection to identify studies relevant to answer our RQ.

3.1. Inclusion criteria

- **IC1:** The study addresses formal and semi-formal modeling of requirements in SoS or similar systems.

3.2. Exclusion criteria

- **EC1:** The study is not directly related to how formally and semi-formally model or validate requirements for SoS or similar systems.
- **EC2:** The study does not discuss challenges of formal and semi-formal modeling and validation of requirements for SoS or similar systems.
- **EC3:** The study is related to modeling or validation of requirements for monolithic systems.
- **EC4:** The study discusses challenges of modeling and validation of requirements for monolithic systems.
- **EC5:** The study is an editorial, keynote, opinion, tutorial, poster or panel.
- **EC6:** The study is duplicated or there is a newer or a more complete one about the same research.
- **EC7:** The study is a gray literature.
- **EC8:** The study is not written in English.
- **EC9:** The full text of the study is not available.
- **EC10:** The is an extended abstract[1]

4. Data extraction and synthesis methods

The data extraction is designed to gather the information necessary to answer the research questions [9]. Thus, for each primary study included, data were extracted and discussed with the experts. For data management and extraction, we used the Parsif.al[2], Tableau Public[3], and MS Excel that helped us to maintain the accuracy and consistency of the extracted data.

[1] To this work, we considered extended abstract studies with until three pages
[2] https://parsif.al/
[3] https://public.tableau.com/pt-br/s/

Reference

[1] K. Pertensen, S. Vakkalanka, and L. Kuzniarz. "Guidelines for Conducting Systematic Mapping Studies in Software Engineering". In: **Information and Software Technology**. 64.C (2015), pp. 1–18.

[2] B. Kitchenham and S. Charters. **Guidelines for performing systematic literature reviews in software engineering**. Tech. rep. Keele University and Durham University Joint Report, 2007.

[3] O. Dieste, A. Grimán, and N. Juristo. "Developing Search Strategies for Detecting Relevant Experiments". In: **Empirical Software Engineering**. 14.5 (2009), pp. 513–539.

[4] B. Kitchenham et al. "**Can we evaluate the quality of software engineering experiments?**" In: 4th International Symposium on Empirical Software Engineering and Measurement (ESEM 2010). Bolzano, Italy: ACM, 2010, pp. 1–8.

[5] H. Munir, M. Moayyed, and K. Petersen. "Considering rigor and relevance when evaluating test driven development: a systematic review". In: **Information and Software Technology**. 56.4 (2014), pp. 375–394.

[6] F. S. Silva et al. "Using CMMI together with agile software development: a systematic review". In: **Information and Software Technology**. 58.14 (2015), pp. 20–43.

[7] C. Machado et al. "**Architectural elements of ubiquitous systems**: a systematic review". In: 3rd International Conference on Software Engineering Advances (ICSEA 2013). Venice, Italy: ACM, 2013, pp. 208–213.

[8] M. Guessi et al. "**A systematic literature review on the description of software architectures for systems of systems**". In: 30th Symposium on Applied Computing ACM (SAC 2015). Salamanca, Spain: ACM, 2015, pp. 1–8.

[9] A. M. Magdaleno, C. M. L. Werner, and R. M. Araujo. "Reconciling software development models: a quasi-systematic review". In: **The Journal of Systems and Software**. 85.2 (2012), pp. 351–369.

APPENDIX

G

Selected Primary Studies of Systematic Mapping of RE-SoS

Formal and Semi-Formal Languages and Techniques for Software-intensive Systems-of-Systems Requirements Modeling

Table 26 presents the 26 studies selected in this systematic mapping, their ID, author name (s), publication title as an external link to where the study was published, and publication year.

Table 26 – List of included primary studies from the SoS requirements mapping

Source: Elaborated by the author.

ID	Author	Title	Year
S1	Atkinson, W. and Cunningham, J.	Proving properties of a safety-critical systems	1991
S2	Leveson, N.G. and Reese, J.D.	Requirements specification for process-control systems	1994
S3	Heimdahl, M.P.E. and Leveson, N.G.	Completeness and consistency in hierarchical state-based requirements	1996
S4	Crow, J. and Di Vito, B.	Formalizing Space Shuttle software requirements: four case studies	1998
S5	Jong, E. et al.	Refinement in requirements specification and analysis: a case study	2000
S6	Sánchez-Alonso, M. and Murillo, J. M.	Specifying cooperation environment requirements using formal and graphical techniques	2002
S7	Ponsard, C. et al.	Early verification and validation of mission critical systems	2005
S8	Linhares, M. V. et al.	Introducing the modeling and verification process in SysML	2007
S9	Ghazel, M. and El Koursi, E.M.	Automatic level crossings: from informal functional requirements' specifications to the control model design	2007
S10	Jamal, M. and Zafar, N.A.	Requirements analysis of air traffic control system using formal methods	2007
S11	Goldsby, H. J. et al.	Goal-based modeling of dynamically adaptive system requirements	2008
S12	Krishna, A. et al.	Consistency preserving co-evolution of formal specifications and agent-oriented conceptual models	2009
S13	Sun, H. et al.	Automata-based verification of security requirements of composite Web Services	2010
S14	Tang, W. et al.	Scenario-based modeling and verification for CTCS-3 system requirement specification	2010
S15	Tang, W. et al.	Scenario-based modeling and verification of system requirement specification for the European Train Control System	2010
S16	Whittle, J. et al.	RELAX: A language to address uncertainty in self-adaptive systems requirement	2010
S17	Yuan, L. et al.	Modelling and verification of the system requirement specification of train control system using SDL	2011
S18	Cimatti, A. et al.	Formalizing requirements with object models and temporal constraints	2011
S19	Zhang, L.	Aspect-oriented formal techniques of cyber physical systems	2012
S20	Deb, N. and Chaki, N.	Verification of i* models for existential compliance rules in remote healthcare systems	2014
S21	Zhang, L.	Modeling large scale complex cyber physical control systems based on system of systems engineering approach	2014
S22	Zou, L. et al.	Verifying Chinese train control system under a combined scenario by theorem proving	2014
S23	Chen, Z. et al.	Exploring a timed-automata fuzzy cognitive maps based approach for modeling systems of systems	2015
S24	Piccolo, A. et al.	Use of formal languages to represent the ERTMS/ETCS system requirements specifications	2015
S25	Han, L. et al.	Safety requirements specification and verification for railway interlocking systems	2016
S26	Wang, Q.-L. et al.	A quality requirements model and verification approach for system of systems based on description logic	2017

APPENDIX

H

Snowballing - Systemic Mapping Protocols

Unified Systematic Mapping Protocol to apply Snowballing

The Systematic Mapping (SM) protocol includes the objective, research questions, search strategy, selection criteria, procedures for studies selection, and data extraction and synthesis methods. The SM protocol was elaborated by the author of this article and validated by experts. To conduct mission and interoperability mappings, we followed the process proposed by Wohlin (2014b) guidelines.

Research question and objective to SoS Mission

The overall research objective of this study is to identify studies that specify and design SoS mission and mission engineering from at design time to at runtime. In order to structure this objective, we developed our research question using the PICO model (KITCHENHAM; CHARTERS, 2007), which considers four elements: Population, Intervention, Comparison, and Outcomes. In the Population, it is described the population group that is observed by the intervention; in the Intervention, it is identified what will be investigated in this SM; Comparison identifies what will be used to compare in the context of the SM; and Outcomes are the expected effects of the proposed intervention. To this SM, the Population comprises the systems engineering community and specifically the requirements engineering community, which has researched Mission and architecture to develop SoS. Intervention for our SM is related to studies that address support the design from SoS. Comparison is not applicable in our SM. Lastly, Outcomes describe all primary studies identified in the literature that report what is adopted to specification and design of SoS mission and mission engineering. The research question formulated to be answered by this SM is:

RQ: Which approaches (i.e., techniques, languages, requirements, and others) have been used for specify and design SoS mission from design time to at runtime?

Selected Criteria

The selection criteria (inclusion and exclusion criteria) were used to evaluate primary studies retrieved from the Scopus and were applied in the first and second selection to identify studies relevant.

Inclusion criteria

IC1: The study addresses specification and design of SoS mission or mission engineering.

Exclusion criteria

- EC1: The study is not directly related to how specification and design of SoS mission and mission engineering of SoS.
- EC2: The study does not discuss challenges of SoS mission.
- EC3: The study is related to specification and design of mission for monolithic systems.
- EC4: The study discusses challenges of mission for monolithic systems.
- EC5: The study is an editorial, keynote, opinion, tutorial, poster or panel.
- EC6: The study is duplicated or there is a newer or a more complete one about the same research.
- EC7: The study is a gray literature.
- EC8: The study is not written in English.
- EC9: The full text of the study is not available.
- EC10: The is an extended abstract

Research question and objective to SoS Interoperability

The overall research objective of this study is to identify studies that technical interoperability requirements analysis of SoS. In order to structure this objective, we developed our research question using the PICO model (KITCHENHAM; CHARTERS, 2007), which considers four elements: Population, Intervention, Comparison, and Outcomes. In the Population, it is described the population group that is observed by the intervention; in the Intervention, it is identified what will be investigated in this SM; Comparison identifies what will be used to compare in the context of the SM; and Outcomes are the expected effects of the proposed intervention. To this SM, the Population comprises the systems engineering community and specifically the requirements engineering community, which has researched SoS and similar systems. Intervention for our SM is related to studies that address support the analysis of technical interoperability requirements. Comparison is not applicable in our SM. Lastly, Outcomes describe all primary studies identified in the literature that report what is adopted to approaches and methodologies to analyze technical interoperability requirements at design time and operational environment. The research question formulated to be answered by this SM is:

RQ: How have technical interoperability requirements been analyzed in SoS domain?

Selected Criteria

The selection criteria (inclusion and exclusion criteria) were used to evaluate primary studies retrieved from the Scopus and were applied in the first and second selection to identify studies relevant.

Inclusion criteria

IC1: The study addresses analysis of technical interoperability requirements (ATIR)

Exclusion criteria

- EC1: The study is not directly related to ATIR in SoS domain or similar domain..
- EC2: The study does not discuss challenges of ATIR.
- EC3: The study is related to ATIR for monolithic systems.
- EC4: The study discusses challenges of ATIR for monolithic systems.
- EC5: The study is an editorial, keynote, opinion, tutorial, poster or panel.
- EC6: The study is duplicated or there is a newer or a more complete one about the same research.
- EC7: The study is a gray literature.
- EC8: The study is not written in English.
- EC9: The full text of the study is not available.
- EC10: The is an extended abstract

Data extraction and synthesis methods

The data extraction is designed to gather the information necessary to answer the research questions (MAGDALENO *et al.*, 2012). Thus, for each primary study included, data were extracted and discussed with the experts. For data management and extraction, we used the Parsif.al , Tableau Public , and MS Excel that helped us to maintain the accuracy and consistency of the extracted data.

APPENDIX I

Selected Primary Studies of Systematic Mapping of Mission-SoS

Table 27 presents the 20 studies selected in this systematic mapping, their ID, author name (s), publication title as an external link to where the study was published, and publication year.

Table 27 – List of included primary studies from the mission mapping

Source: Elaborated by the author.

ID	Author	Title	Year
ID1	Dash et al.	A Practical Approach for Phased Mission Analysis	1989
ID2	Silva et al.	A Mission-Oriented Approach for Designing System-of-Systems	2015
ID3	Vensoder et al.	RT-171: Mission Engineering Competencies	2018
ID4	Mokhtarpour et al.	A Conceptual Methodology for Selecting the Preferred System of Systems	2017
ID5	Neto et al.	A Study on Goals Specification for Systems-of-Information Systems: Design Principles and a Conceptual Model	2018
ID6	Hernandez, A.	Mission Engineering and Analysis: Innovations in the Military Decision Making Process	2017
ID7	Ayla, G.	Supervision for System of Systems Engineering Based on Fuzzy Logic Approach	2015
ID8	Etzien, et al.	Designing for adaptability and evolution in system of systems engineering	2014
ID9	Kumar et al.	Multilevel Modeling of System of Systems	2017
ID10	Easy et al.	Reliability Analysis of Phased Mission	1975
ID11	Bindra et al.	DESCENT: Mission Architecture and Design Overview	2017
ID12	Jayaputera et al.	Mission Impossible? Automatically Assembling Agents from High-Level Task Descriptions	2003
ID13	Anđelković, B.	Mission Level Modeling and Simulation Environment for Hardware/Software Systems	2006
ID14	Sousa-Poza, A.	Mission Engineering	2015
ID15	Deitz et al.	The Missions & MMatching Military Asets to Mission Objectives	2016
ID16	Bhasin et al.	Lunar Communication Terminals for NASA Exploration Missions: Needs, Operations Concepts and Architectures	2008
ID17	Mokhtarpour et al.	Mission Reliability Analysis of Phased-Mission Systems-of-Systems with Data Sharing Capability	
ID18	Ricci et al.	Architecting Systems of Systems with Ilities: an Overview of the SAI Method	2014
ID19	Garces et al.	A Process to Establish, Model and Validate Missions of Systems-of-Systems in Reference Architectures	2017
ID20	Zhu et al.	Ontology-Based Mission Modeling and Analysis for System of Systems	2017

APPENDIX

J

Selected Primary Studies of Systematic Mapping of Interoperability-SoS

Table 28 presents the 36 studies selected in this systematic mapping, their ID, author name (s), publication title as an external link to where the study was published, and publication year.

Table 28 – List of included primary studies from the interoperability mapping

Source: Elaborated by the author.

ID	Author	Title	Year
ID1	Ceci et al.	Facilitating Data Interoperability in Science and Technology –,A Case Study and a Technical Solution	2017
ID2	Figay et al.	Interoperability framework for dynamic manufacturing networks	2012
ID3	Daclin et al.	Writing and verifying interoperability requirements: Application to collaborative processes	2016
ID4	Agostinho et al.	Sustaining interoperability of networked liquid-sensing enterprises:A complex systems perspective	2015
ID5	Hogie et al.	Using standard Internet Protocols and applications in space	2005
ID6	Chapurlat et al.	System interoperability: definition and propositionof interface model in MBSE Context	2012
ID7	Gries et al.	Modeling Semantic Dependencies to allow FlowMonitoring in Networks with Black-box Nodes	2019
ID8	Chen et al.	Framework for Enterprise Interoperability	2006
ID9	Karagoz et al.	Developing Conceptual Models of the Mission Space (CMMS) – AMetamodel Based Approach	2006
ID10	Malik et al.	Towards verifying service interoperability requirements for pervasivecomputing environments	2012
ID11	Billaud et al.	Interoperability as a Key Concept for the Control and Evolution of the System of Systems (SoS)	2017
ID12	Pridmore et al.	Interoperability - How do we know when we have achieve it?	1989
ID13	Chapurlat et al.	System of Systems Architecting: A Behavioural and Properties Based Approach for SoS "–ilities" Modelling and Analysis	2015
ID14	Mandutianu, S.	Space Networking- Interacting in Space Communication Networks	2000
ID15	Wilson, W.	Applying Layering Principles to Legacy Systems: Link 16 as a Case Study	2001
ID16	Dias et al.	Neptus - A Framework to Support Multiple Vehicle Operation	2005
ID17	Miao et al.	Comparing Smart Grid Technology Standards Roadmap of The IEC, NIST and SGCC	2012
ID18	Cayir et al.	Information Technology Interoperability Awareness: A Taxonomy Model Based on Information Requirements and Business Needs	2008
ID19	Wang et al.	A Survey of Technical Requirements and Consumer Application Standards for IP-based Smart Grid AMI Network	2010
ID20	Abdullah et al.	Metadata Interoperability Requirements for Aggregrating Islamic Manuscript Bibliographic Records	2014
ID21	Manley, D.	The Nato Drive to Mission Modularity	2018
ID22	Gracanin et al.	A Service-Centric Model for Wireless Sensor Networks	2005
ID23	Monarch at al.	Autonomy and Interoperability in System of Systems Requirements Development	2005
ID24	Chapulart et al.		2010
ID25	Pullen et al.	Coalition Command and Control – Simulation Interoperation as a System of Systems	2016
ID26	Bruzzone et al.	Interoperability Requirements for Developing Simulation Solutions for Innovative Integrated Systems	2014
ID27	White, B. E.	Layered Communications Architecture for the Global Grid	2001
ID28	Petersen et al.	Interoperability Requirements Elicitation Validation and Solutions Modeling	2005
ID29	Chen et al.	Architectures for enterprise integration and interoperability: Past, present and future	2008
ID30	Garces et al.	Towards a Taxonomy of Software Mediators for Systems-of-Systems	2018
ID31	Fotouhi et al.	Interoperability in heterogeneous low-power wireless networks for health monitoring systems	2016
ID32	Roque et al.	Interoperability in Collaborative Processes: Requirements Characterisation and Proof Approach	2009
ID33	Tibaut et al.	Interoperability requirements for automated manufacturing systems in construction	2016
ID34	Vachtsevanos et al.	Intelligent control of Unmanned Aerial Vehicles for Improved Autonomy and Autonomy and Reliability	2004
ID35	Mallek et al.	The application of interoperability requirement specification and verification to collaborative processes in industry	2012
ID36	Qureshi et al.	Analyzing Interoperability Requirements for Adaptive Service-based Applications: A Goal-oriented Approach	2010

APPENDIX

K

Survey Protocol and Questionnaire

This appendix present the questionnaire applied to the industry practitioners and researchers to collect real data and analysis the current state of the practice problems regarding SoS interoperability and task of interoperability requirements analysis in development of the systems and similar complex system. Figure 44 shows the welcome page the of Survey Anyplace[1] with the partner Fraunhofer IESE, which we used to conducted the online survey.

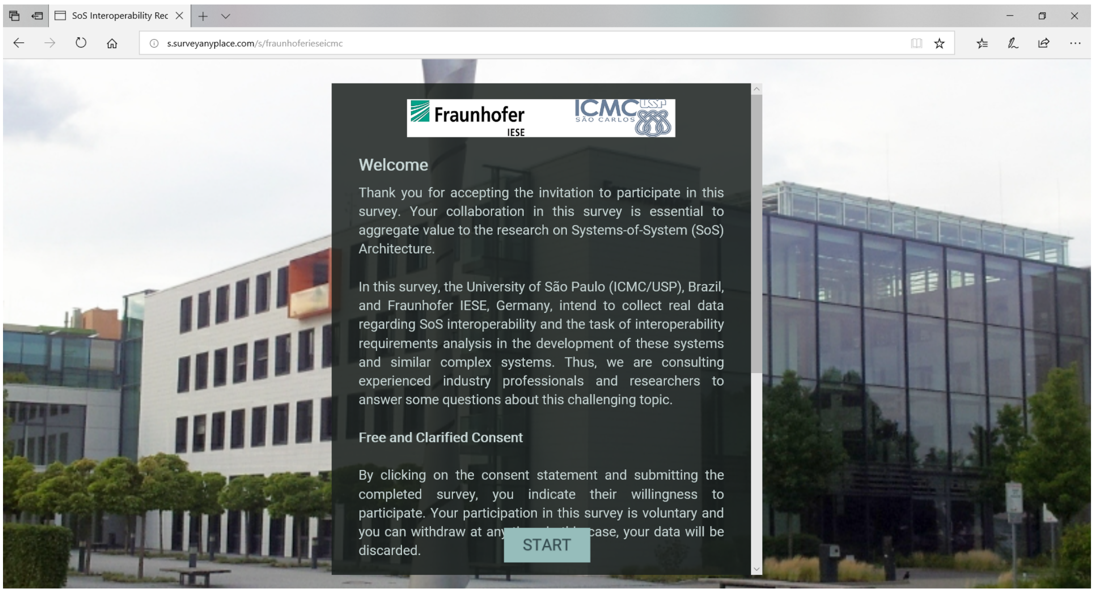


Figure 44 – Excerpt of the Welcome in the system Survey Anyplace

Source: Elaborated by the author.

SoS Interoperability Requirements Analysis

Welcome

[1] <https://surveyanyplace.com/>

Thank you for accepting the invitation to participate in this survey. Your collaboration in this survey is essential to aggregate value to the research on Systems-of-System (SoS) Architecture.

In this survey, the University of São Paulo (ICMC/USP), Brazil, and Fraunhofer IESE, Germany, intend to collect real data regarding SoS interoperability and the task of interoperability requirements analysis in the development of these systems and similar complex systems. Thus, we are consulting experienced industry professionals and researchers to answer some questions about this challenging topic.

Free and Clarified Consent

By clicking on the consent statement and submitting the completed survey, you indicate their willingness to participate. Your participation in this survey is voluntary and you can withdraw at any time; in this case, your data will be discarded.

Survey Data: 1st reminder: July 09, 2018; 2st reminder: July 21, 2018; and **Deadline** August 10, 2018.

Kind regards,

PhD Student Cristiane Lana (University of São Paulo, Brazil; University of Kaiserslautern and Fraunhofer IESE, Germany)
Prof. Dr. Elisa Yumi Nakagawa (University of São Paulo, Brazil)
Dr. Pablo Oliveira Antonino (Fraunhofer IESE, Germany)
Prof. Dr. Milena Guessi (University of São Paulo, Brazil)
Prof. Dr. Dieter Rombach (Fraunhofer IESE and University of Kaiserslautern, Germany)

Architectural Analysis Practices of SoS Interoperability Requirements

The following definitions apply to the terms used in this survey:

Systems-of-Systems are used to describe large, complex, and software-intensive systems composed of separately developed systems with managerial and operational independence (i.e., constituent systems) that work collaboratively towards the achievement of a global mission.

Interoperability is a requirement that provides dynamic interactive information and the possibility of data exchange among constituent systems to work together to enable new functionalities.

Emergent Behavior is the combination of constituent systems interacting with each other and with the environment over time to create new functionalities.

System Requirements Analysis is a structural process aimed at translating user capability into requirements.

1 - Have you (or your organization) ever performed interoperability requirements analysis for SoS development?*

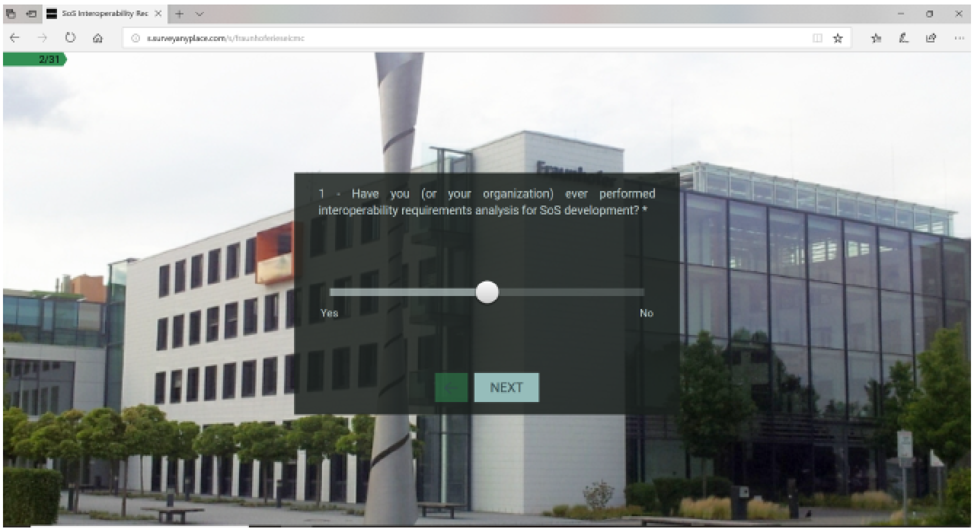


Figure 45 – Research question one - Architectural Analysis Practices

Source: Elaborated by the author.

2 - In your opinion, which is(are) the reason(s) why interoperability requirements analysis is not considered in SoS development? Please choose all that apply.

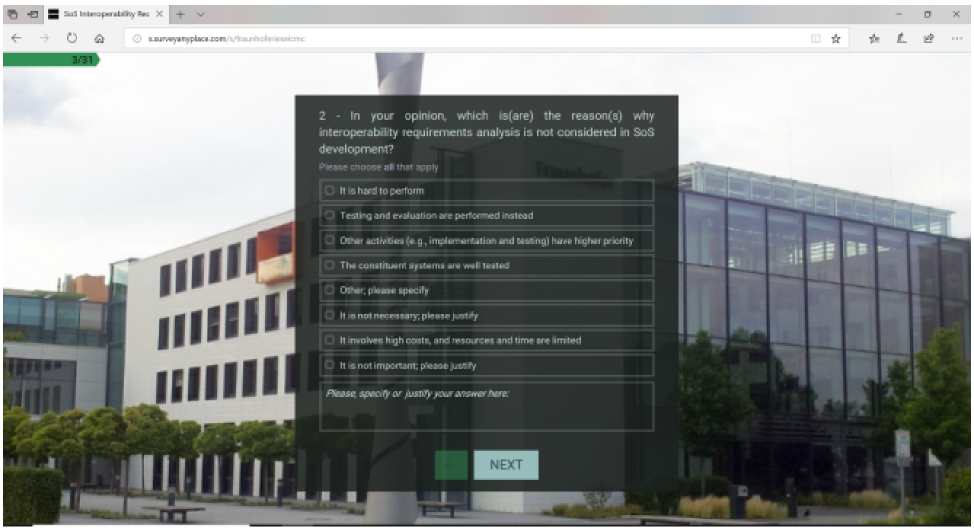


Figure 46 – Research question two - Architectural Analysis Practices

Source: Elaborated by the author.

3 - In your opinion, who should perform the task of interoperability requirements analysis?* Please chose all that apply.

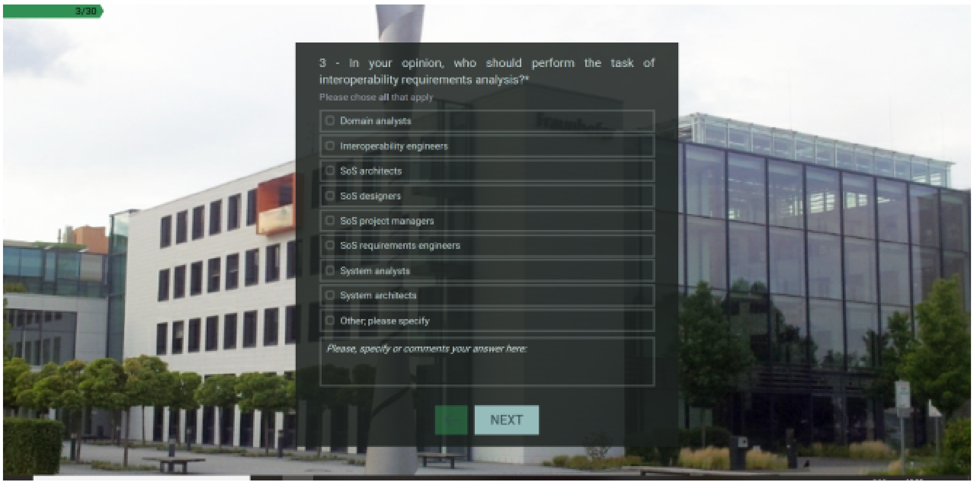


Figure 47 – Research question three - Architectural Analysis Practices

Source: Elaborated by the author.

4 - In your experience, which are the main tasks that deal with interoperability requirement analysis?* Please all that apply.

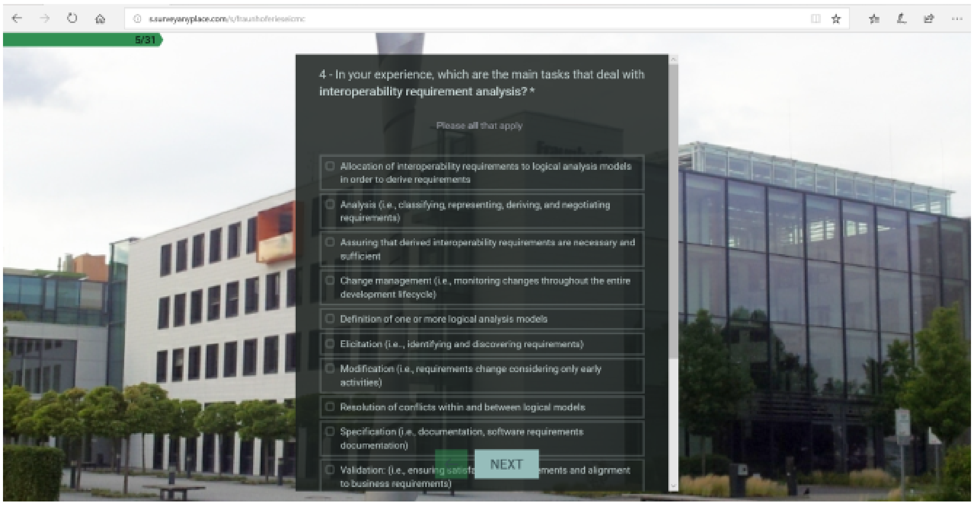


Figure 48 – Research question four - Architectural Analysis Practices

Source: Elaborated by the author.

5 - Which formal level has been used to document interoperability requirements of SoS in projects (or organizations) in which you have been involved?* Please all that apply.

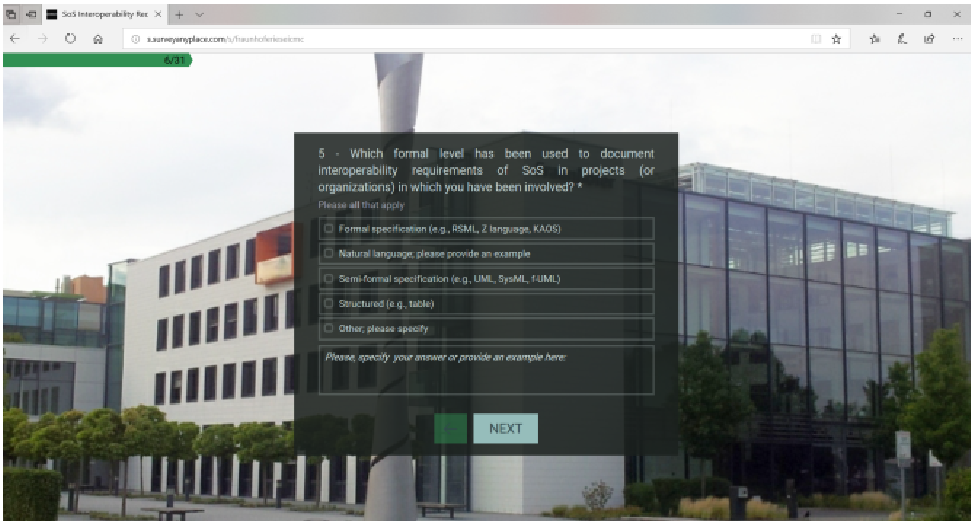


Figure 49 – Research question five - Architectural Analysis Practices

Source: Elaborated by the author.

6 - In your experience, which of the following languages support the specification of SoS interoperability requirements?* Please choose all that apply.

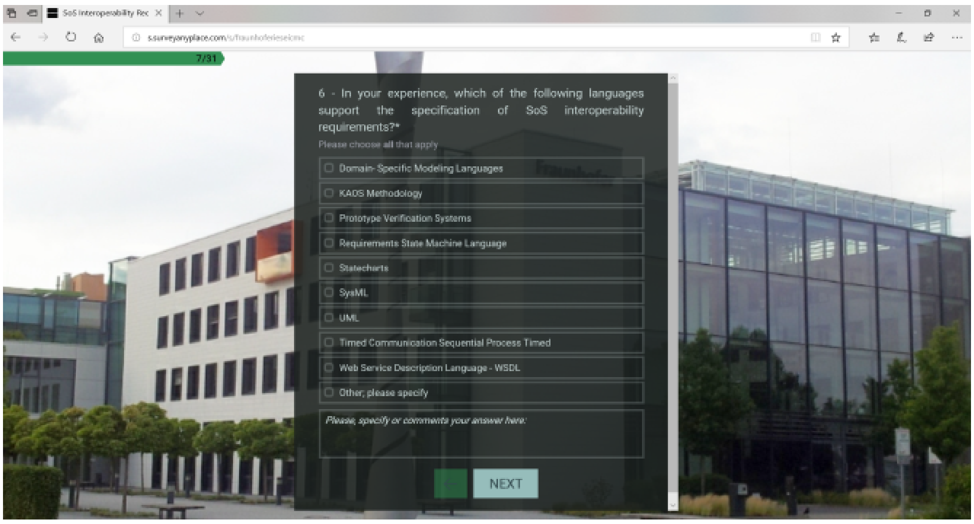


Figure 50 – Research question six - Architectural Analysis Practices

Source: Elaborated by the author.

7 - Considering the SoS development lifecycle, in which activities have you elicited interoperability requirements in projects (or organizations) in which you have been involved?* Please all that apply

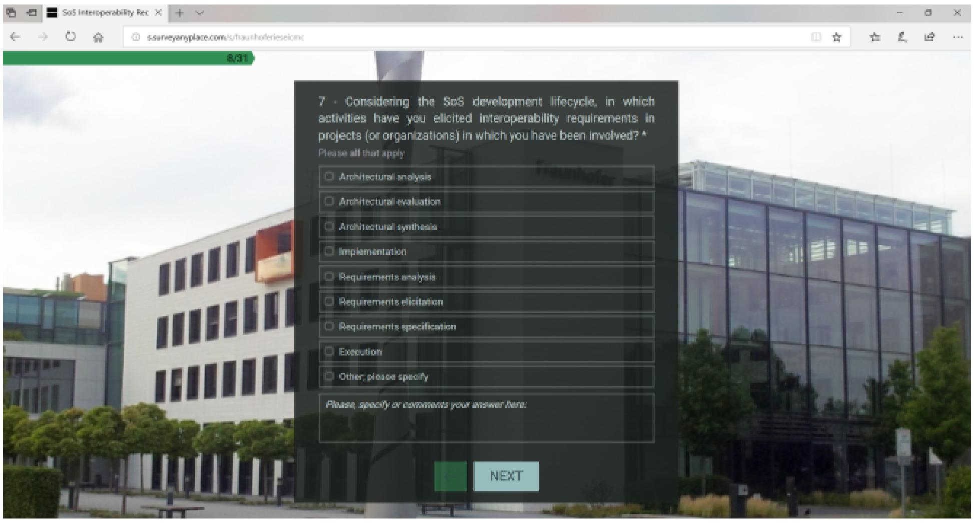


Figure 51 – Research question seven - Architectural Analysis Practices

Source: Elaborated by the author.

8 - In your experience, how much effort (%) has been spent in the SoS Requirements Engineering phase on eliciting interoperability requirements?* Please choose only one of the following.

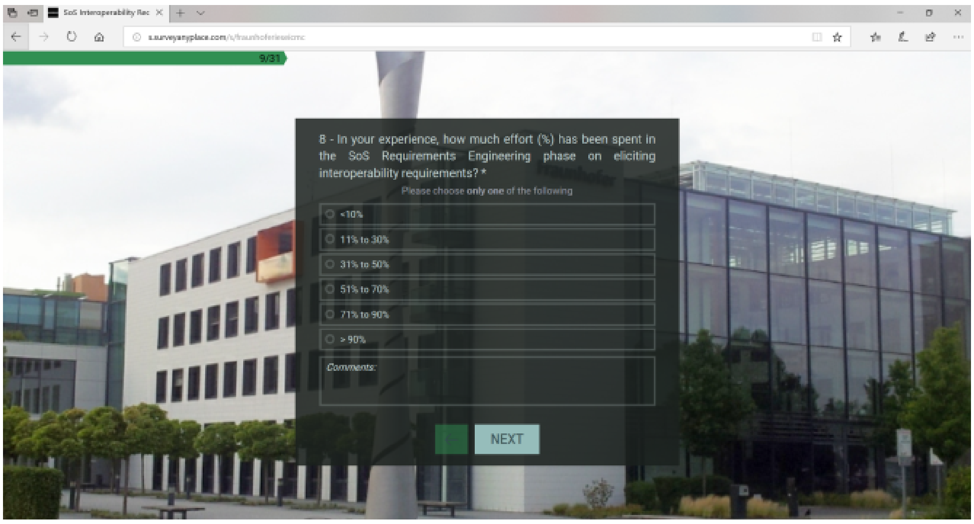


Figure 52 – Research question eight - Architectural Analysis Practices

Source: Elaborated by the author.

9 - In your experience, what has been the abstraction level of the interoperability requirements elicited in the SoS Requirements Engineering phase?*

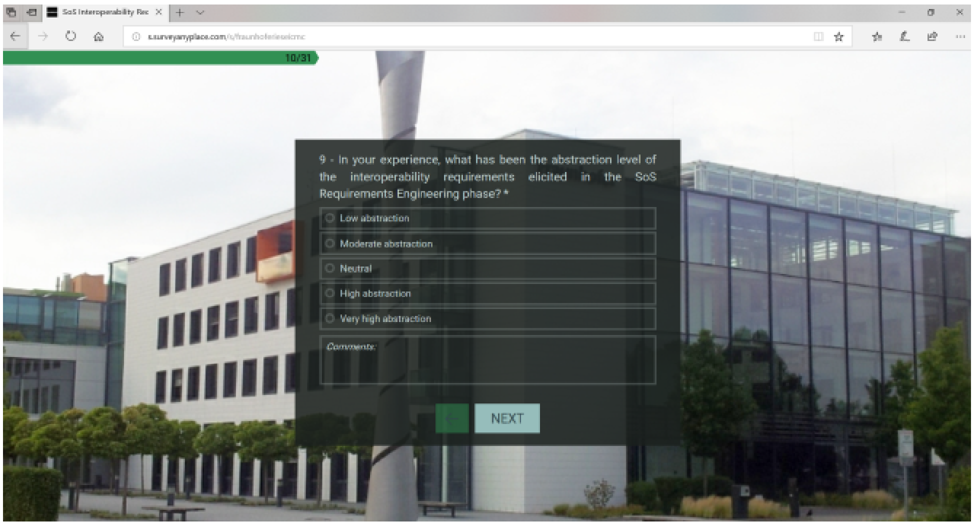


Figure 53 – Research question nine - Architectural Analysis Practices

Source: Elaborated by the author.

10 - In your experience regarding SoS architectural analysis, how much effort (%) has been spent on eliciting interoperability requirements?* Please choose only one of the following.

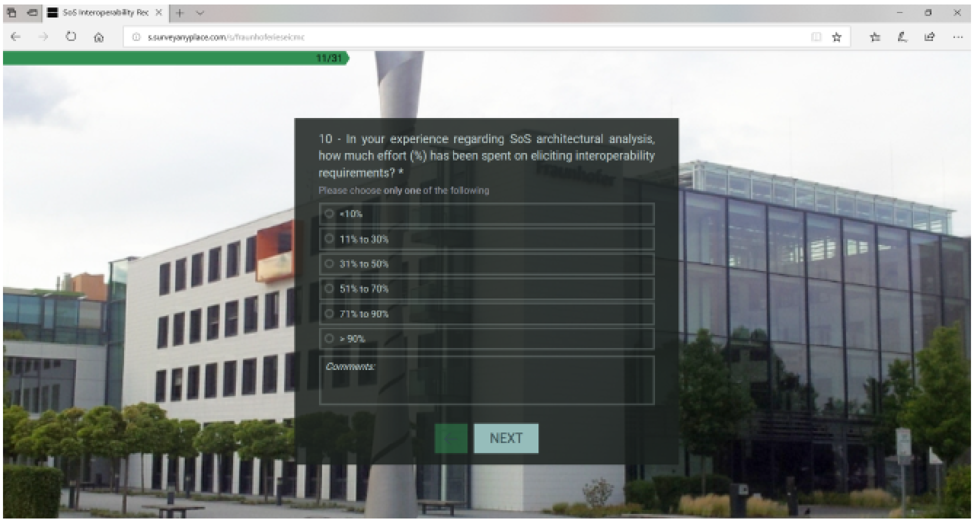


Figure 54 – Research question ten - Architectural Analysis Practices

Source: Elaborated by the author.

11 - In your experience, what has been the abstraction level of the interoperability requirements elicited in the SoS Architectural Analysis phase?*

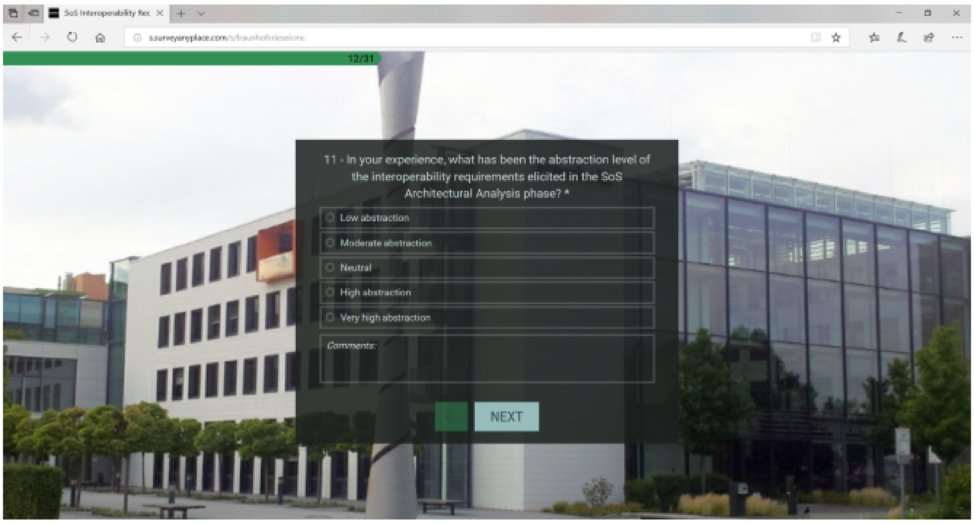


Figure 55 – Research question 11 - Architectural Analysis Practices

Source: Elaborated by the author.

12 - In your opinion, which SoS characteristics impact SoS interoperability requirements?*

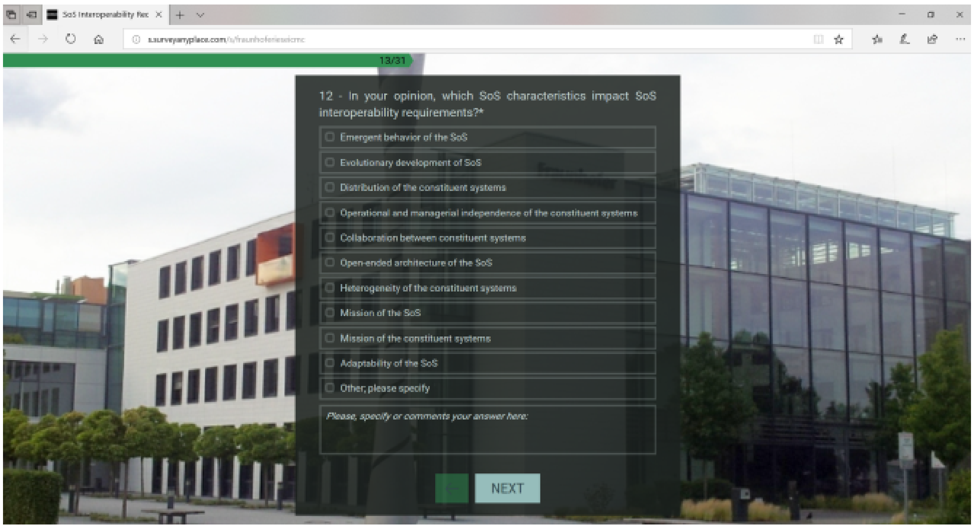


Figure 56 – Research question 12 - Architectural Analysis Practices

Source: Elaborated by the author.

13 - In your opinion, which are the main challenges/difficulties regarding the analysis of interoperability requirements?* Please choose all that apply.

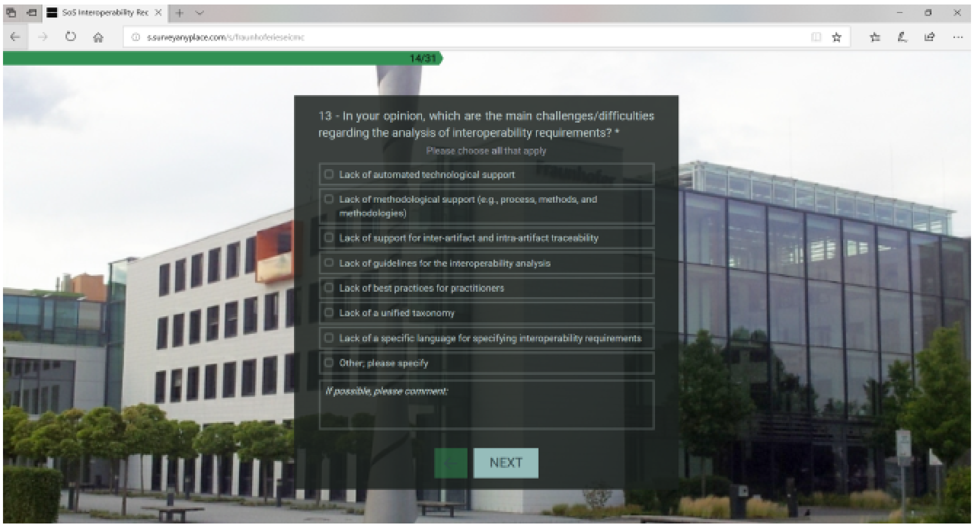


Figure 57 – Research question 13 i- Architectural Analysis Practices

Source: Elaborated by the author.

14 - Have interoperability requirements been represented in the architectural views of the SoS?* Please all that apply.

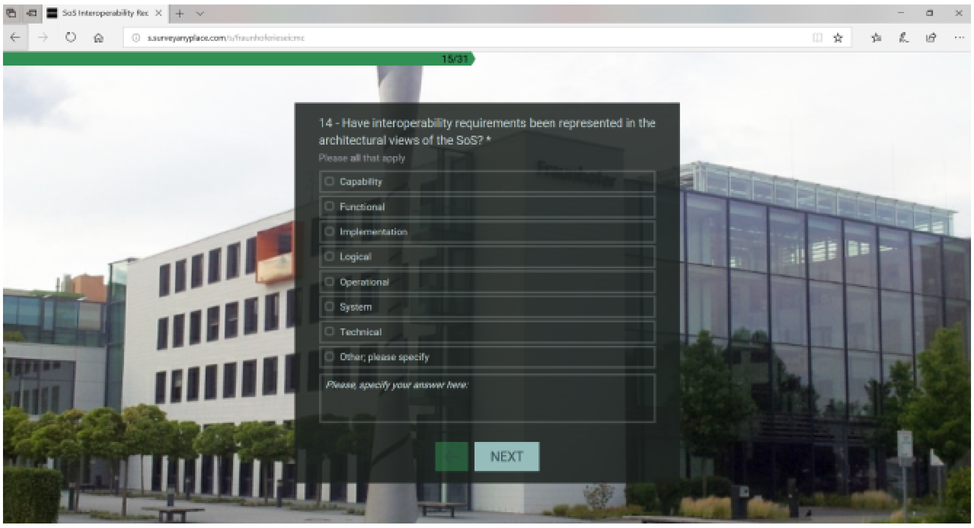


Figure 58 – Research question 14 - Architectural Analysis Practices

Source: Elaborated by the author.

15 - How did the architect/engineer solve issues related to uncovered interoperability requirements?*

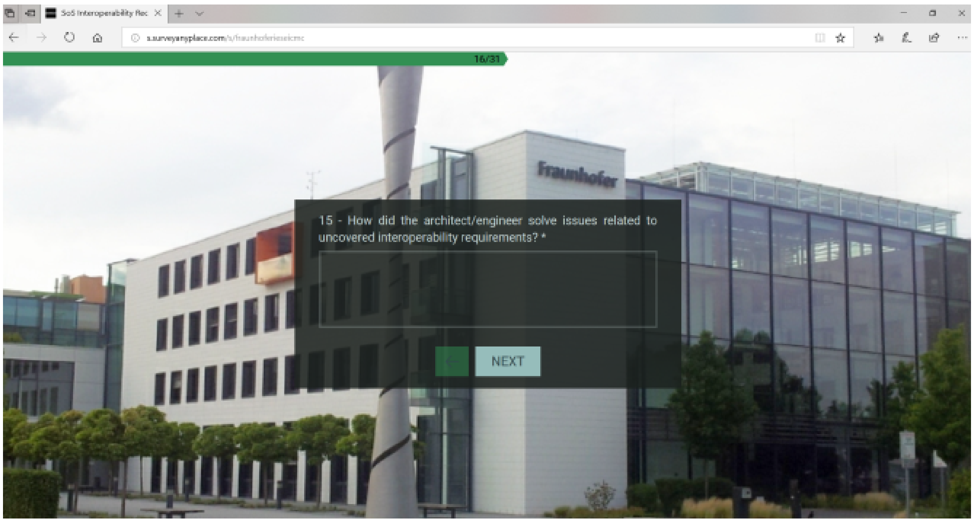


Figure 59 – Research question 15 - Architectural Analysis Practices

Source: Elaborated by the author.

16 - In your experience, do you consider interoperability as a . . . *

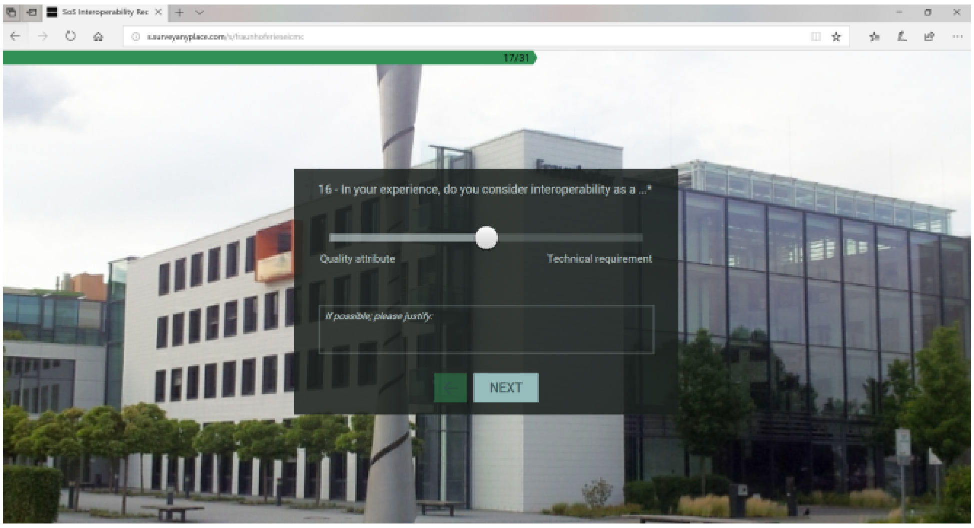


Figure 60 – Research question 16 - Architectural Analysis Practices

Source: Elaborated by the author.

Landscape regarding SoS interoperability

It would be amazing for us to continue hearing your opinions and learning from your experience! We are almost done!

16 - Which development methodologies have been applied in SoS projects in which you have been involved?* Please choose all that apply.

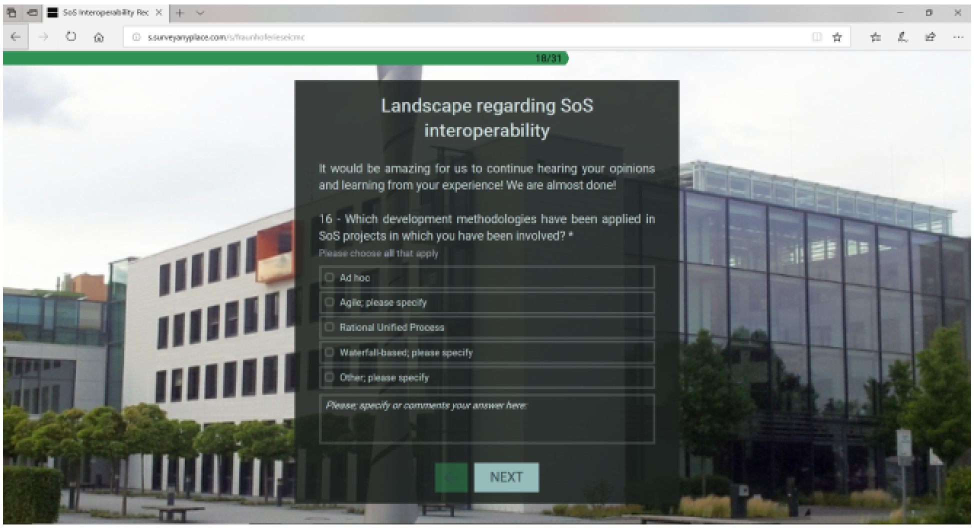


Figure 61 – Research question 16 - SoS interoperability

Source: Elaborated by the author.

17 - Have you been using any framework to support the development of SoS?* Please choose all that apply.

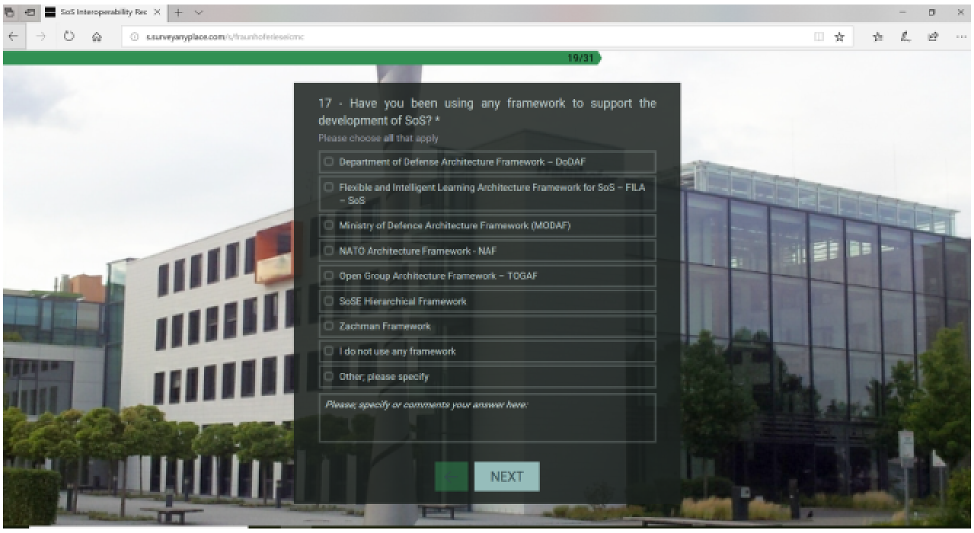


Figure 62 – Research question 17 - SoS interoperability

Source: Elaborated by the author.

18 - Do you have experience with SoS interoperability in any of the following application domains?* Please choose all that apply.

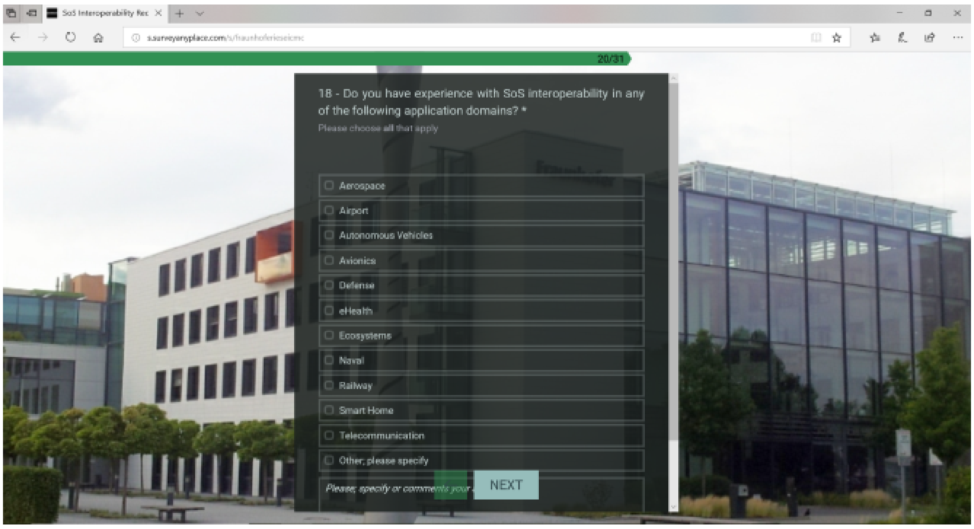


Figure 63 – Research question 18 - SoS interoperability

Source: Elaborated by the author.

19 - Which type of SoS interoperability has been used in projects (or organizations) in which you have been involved?* Please choose all that apply

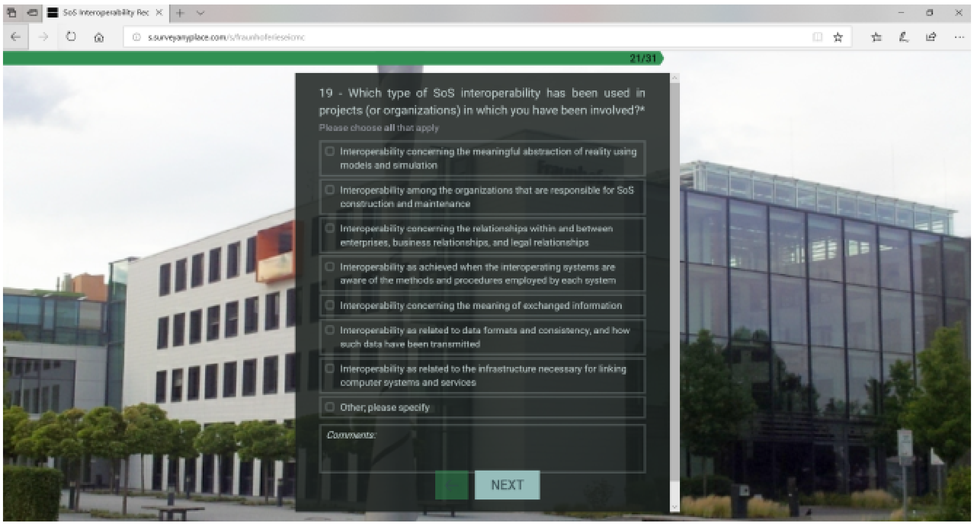


Figure 64 – Research question 19 - SoS interoperability

Source: Elaborated by the author.

20 - In particular, based on your experience, which type of interoperability is/should be considered most relevant during the architectural analysis of SoS?* Please choose all that apply.

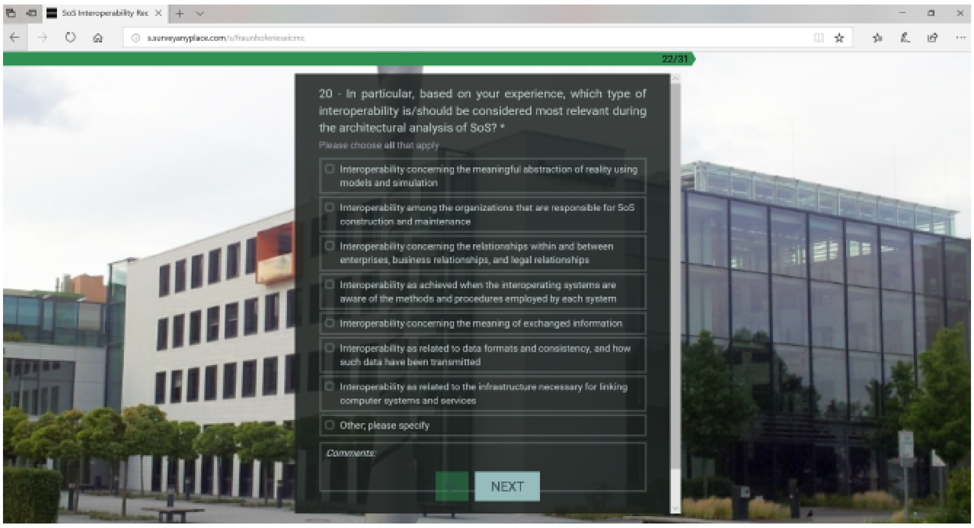


Figure 65 – Research question 20 - SoS interoperability

Source: Elaborated by the author.

21 - Which model have you been applying to evaluate SoS interoperability?* Please choose all that apply.

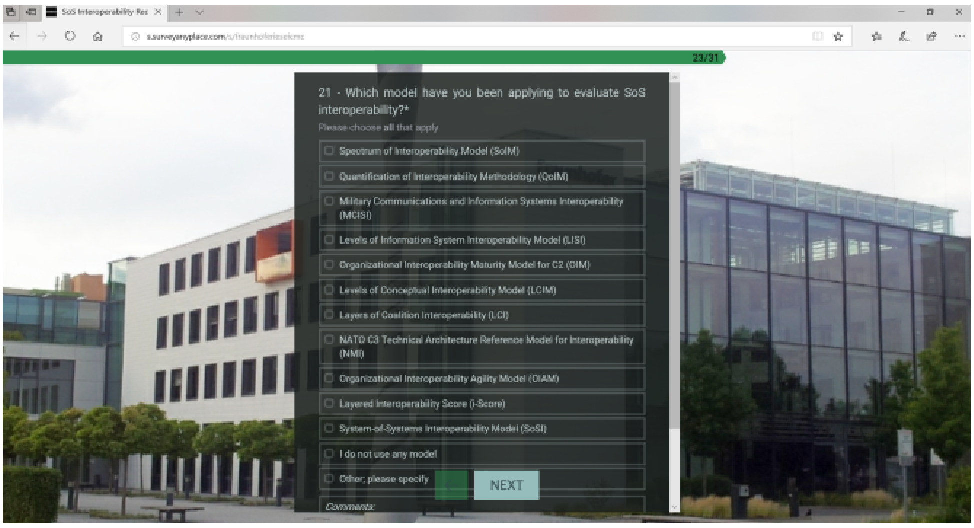


Figure 66 – Research question 21 - SoS interoperability

Source: Elaborated by the author.

Please feel free to provide any information or send by email any file that you consider useful for this survey.

Demographic data

1- What is your country of residence?*

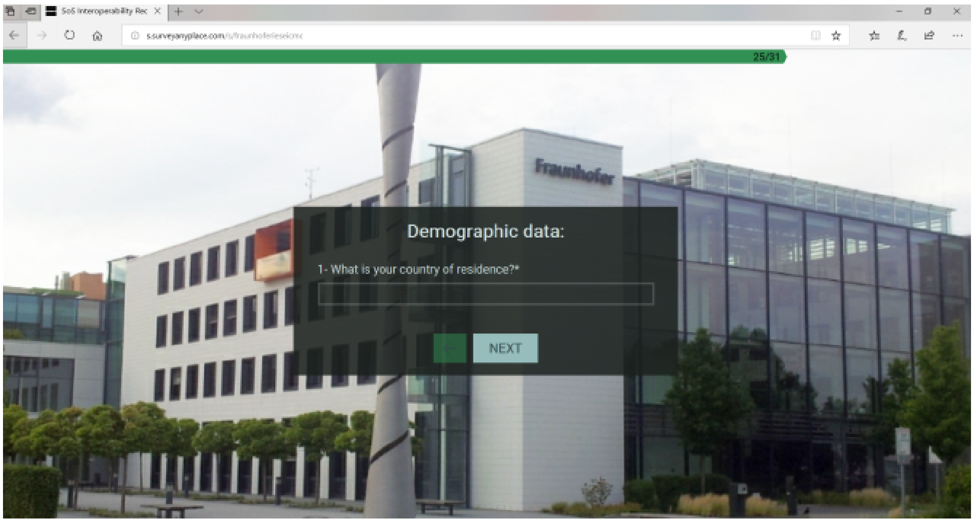


Figure 67 – Demographic research question one

Source: Elaborated by the author.

2 - What is your current occupation/career?* Please choose only one of the following.

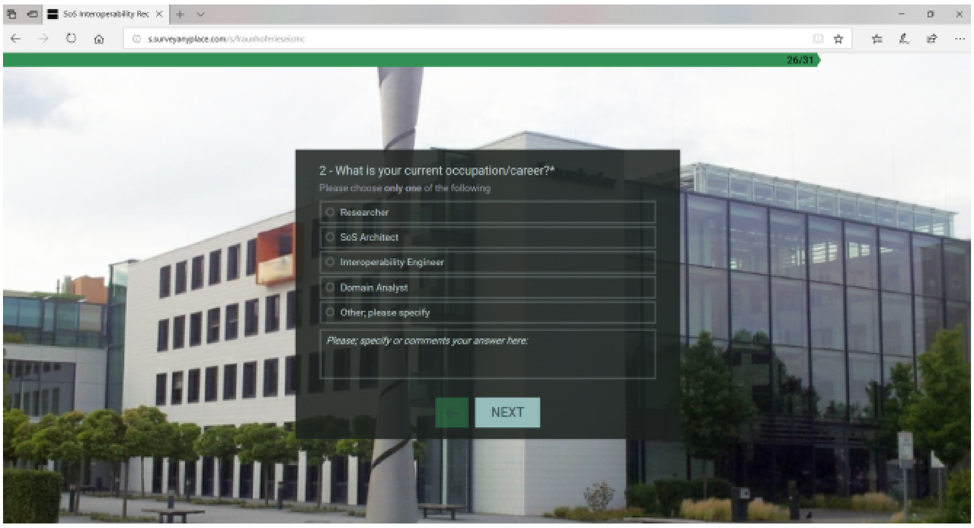


Figure 68 – Demographic research question two

Source: Elaborated by the author.

3 - How many years have you been working in this position?* Please specify.

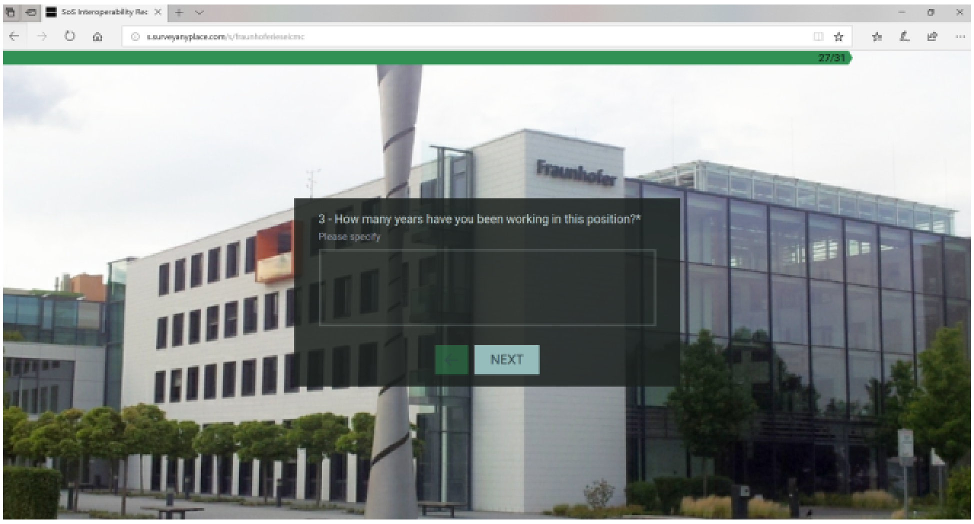


Figure 69 – Demographic research question three

Source: Elaborated by the author.

4 - What is your educational degree?* Please choose only one of the following.

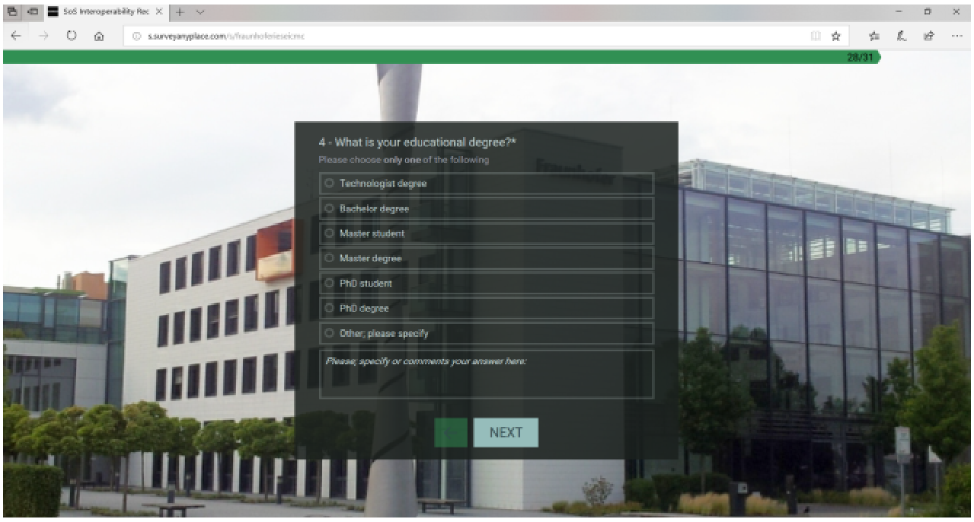


Figure 70 – Demographic research question four

Source: Elaborated by the author.

5 - What is your educational background?* Please choose only one of the following.

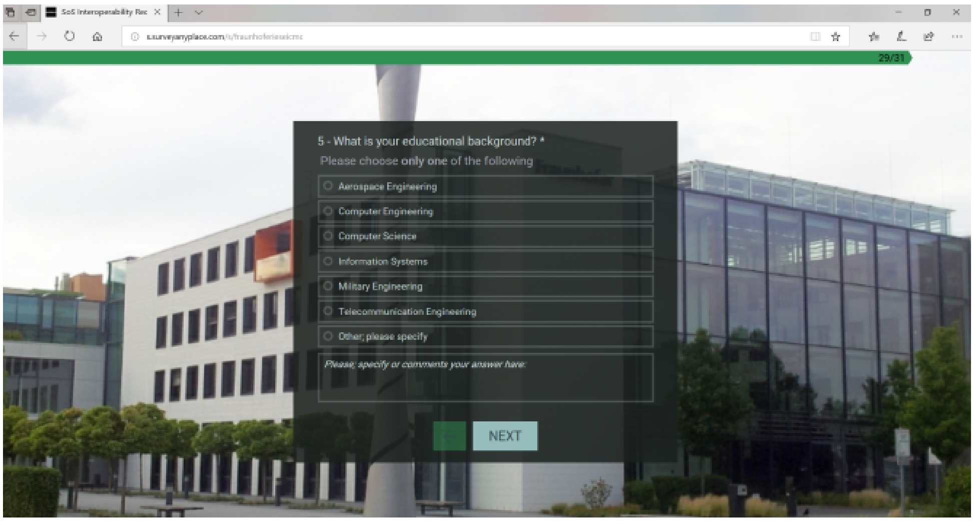


Figure 71 – Demographic research question five

Source: Elaborated by the author.

6 - In which company do you work? If possible, please specify.

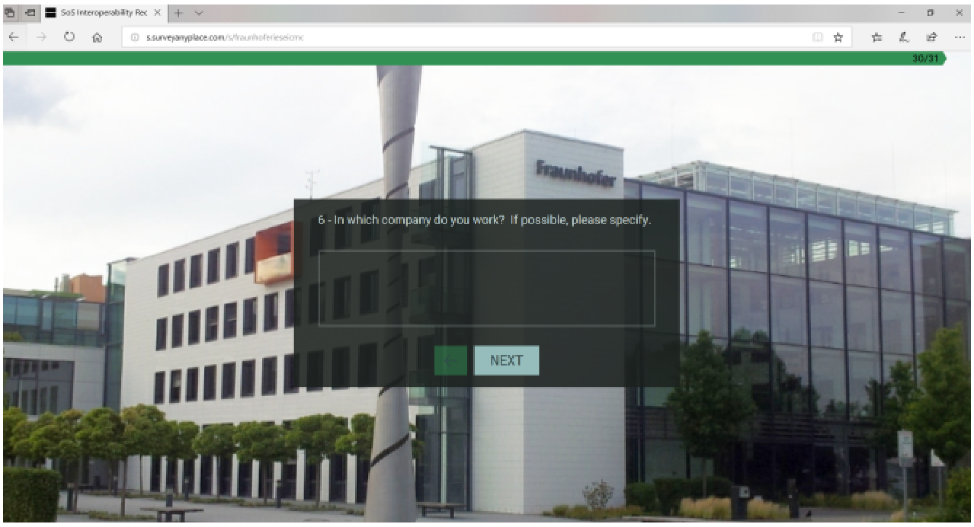


Figure 72 – Demographic research question six

Source: Elaborated by the author.

Thank you for your willingness to help us! If you are interested in the results of this survey, just write your email address in this field. We will then contact you as soon as we finish this survey.

Sincerely,

PhD Student Cristiane Lana (University of São Paulo, Brazil; University of Kaiserslautern and Fraunhofer IESE, Germany)
Prof. Dr. Elisa Yumi Nakagawa (University of São Paulo, Brazil)
Dr. Pablo Oliveira Antonino (Fraunhofer IESE, Germany)
Prof. Dr. Milena Guessi (University of São Paulo, Brazil)
Prof. Dr. Dieter Rombach (Fraunhofer IESE and University of Kaiserslautern, Germany)

APPENDIX

L

Interviews Script with Key Questions

This appendix presents the key questions used to guide the interview with the 27 participants.

- What are the business stakeholders impact the strategical analysis?
- How have standards and regulations interfering with the organizational activity?
- What are the main technologies there at each organizational level?
- How have mission influenced the strategical decision to activities in operational environmental?
- How are the capabilities of operational environmental identified at the tactical level?
- How have interoperability requirements been identified and specified?
- How have Interoperability requirements been meet to the technical level?
- How has value been considered to improve the quality of the services or software?
- what is the vision of service development based on value and user experience, what technical requirements would be needed to meet them?
- How have the stakeholders and end-user influenced the decision of implantation of a new technology?
- What factors were considered to implement a new system for the operating environment?
- How is operational technology aligned with business strategies?
- What benefits do these technologies bring to your processes [insert the process of each company or that each individual works]?

- What is the importance of relating organizational mission, system mission and operational activity?
- Which interoperability requirements are considered essential for each organizational level: strategic, tactical, operational, and technological?
- In interoperability, a threat is everything that impacts an already established interoperation. What do you consider to be the main technical interoperability requirements threats?
- A barrier is what prevents interoperability from being established, considering this context, what do you consider to be the main technical interoperability requirements barriers currently for systems interoperability?
- Do you do interoperability requirements analysis, if so, how is the technical interoperability requirement analyzed?
- What analysis criteria and/or metrics are considered in the assessment?
- What are the technical and metrics attributes considered essential for the interoperation of systems, without missing the mission?
- What are the input and output types exchanged between these systems?
- What are the quality attributes and metrics considered essential for technical interoperability requirements for hardware, software, network infrastructure (communication and data) standards, protocols, interfaces, others?
- What types of systems exist in the industry, how is the acquisition made and how is it managed?
- The amount of data has grown exponentially and not all data needs to be available all the time. How do you prioritize data to be available in real time during the operation process for decision making or to be only online?

APPENDIX

M

List of Category, Sub-category, and Codes

This appendix displays the category, sub-categories, and codes identified in the conduction of grounded theory described in Chapter 4.

Table 29 – Data/communication channel

Source: Elaborated by the author.

Category: Data/Communication Channel	
Sub-category	**Network**
Codes	Application
	Command (executable instruction)
	Data format
	Direction of Information flow
	Distribution Network
	Interface description
	Meaning
	Network path
	Networks connectivity
	Provide data at runtime
	Provide data on line
	Shared applications
	Standards
	Timelines or data latency
	Transmission protocols
	Transmission rates
	Transmitter/Receiver
Sub-category	**Security**
Codes	Common messages format
	Confirmation (receipts or execution)
	Consequences (results)
	Cryptograph
	Governing implementation criteria
	Message standard
	Real Time Monitoring
	Reporting Responsibility (R2)
	Security policy
	Threaten and Attack
	Trusted information exchange
Sub-category	**Access control**
Codes	Access audit
	Access security
	Authentication
	Confidentiality
	Security management services

Table 30 – Technical interoperability requirements quality

Source: Elaborated by the author.

Category: Technical Interoperability Requirements Quality	
Sub-category	**Technical Infrastructure**
Codes	Accessibility
	Adaptability
	Affordability
	Autonomy
	Availability
	awareness
	Compatibility
	Composability
	Confidentiality
	Configurability
	Connectivity
	Cooperation
	Dependability
	Distributivity
	Economic risk mitigation
	Environmental risk mitigation
	Evolvability
	Extensibility
	Extensibility
	Flexibility
	Generality
	Health and Safety risk mitigation
	Heterogeneity
	Integration
	integrity
	Maintainability
	Performance
	Preparedness
	Privacy
	Redundancy
	Reliability
	Resilience
	Response time
	Reuse of resource
	Safety
	Security
	Simplicity
	Sustainable
	Testability
	Traceability
	Transparency
	Trust
	Upgradeability
	Usefulness
Sub-category	**Software**
Codes	Adaptability
	Execution efficiency
	Fault Tolerance
	Installability
	Loose event coupling
	Portability
	Recoverability
	Scalability
Sub-category	**Hardware**
Codes	High performance
	Robustness
	Storage efficiency
	Temperature

Table 31 – Sustainable technical interoperability

Source: Elaborated by the author.

Category:	Sustainable Technical Interoperability
Sub-category	**Dilemma Openness versus Standardization**
Codes	Extensibility
	Generality
	Interconnection services
	Open interface
	Open standards
	Open technology
	Openness versus Standardization
	Standardization or Common protocol
Sub-category	**Social Environment**
Codes	Collaboration between partner organizations
	Manager stakeholders needs
	Resources security
	Stakeholder engagement
	User experience analysis
	User-facing capability
Sub-category	**Economic**
Codes	Cost Reduction and better strategies
	Costs of interoperation
	Equipment maintenance cost
	Prices of Resources
	Reduce the loss of transmission
	Reuse of resources
	Risk analysis
	Share resources
	Transactions cost
Sub-category	**Technical**
Codes	Application
	Communication
	Concorrent users
	Contractual agreements
	Data accessibility
	Data durability and Reliability
	Data exchanging
	Heterogeneity of constituents
	Information security
	Information usefulness
	Infrastructure
	Multi-functional devices
	Platform evolution
Sub-category	**Environmental**
Codes	Awareness
	Resource integrity
Sub-category	**Individual**
Codes	Human health
	Mitigating the health and safety risk
	Reducing waste and cost

Table 32 – Strategical decision

Source: Elaborated by the author.

Category: Strategical Decisions	
Sub-category	Business Alignment and Misalignment
Codes	Business goal versus systems goal
	Business objectives analysis
	Business procedures improve
	Changing the Environment
	Collaboration among organization
	Comply with policy, regulation, and laws
	Define product and value
	Domain
	External manageability
	Find problem target
	Information for multiple organizational levels
	Measure performance for multilevel
	Monitoring tasks operational
	Set of capability
	Simulation environments
	Statement of needs
Sub-category	Architectural decisions
Codes	Adopt directory services
	Architectural alternatives
	Architectural pattern
	Behavioral view
	data presentation exchange
	Develop layered system
	Interoperate with legacy system
	level of performance attendance
	Use of mediators

Table 33 – Problems and consequences

Source: Elaborated by the author.

Category: Problems and Consequences	
Sub-category	Technical Problem
Codes	Lack of Clarity in Use of Protocols
	Diversified Standards to Multiples Systems
	Closed Standards
	Systems are much specific
	Protocols is not updated
	Difficulty of Define Threats
	Lack of network coverage
	Conceptual inconsistent solutions
Sub-category	Technical Consequences
codes	Difficulty in changing technology
	Interoperation is most costly
	Systems is more difficult to interoperate with new technology
	Threats not mitigate on time
	Lack of cooperation skills
	Difficulty interoperating everywhere
	To wrong implications and decision

Table 34 – Aware communication

Source: Elaborated by the author.

Category: Aware Communication	
Sub-category	**Factors that Influence Quality of Requirements**
Codes	Adaptability
	Concepts understanding
	Cultural conflicts
	Foreseen versus unforeseen
	People distributed logistic with production continue
	Register of rationale
	Requirements volatility
	Understand the stakeholder need to define the requirement

Table 35 – Technical interoperability requirements challenges

Source: Elaborated by the author.

Category: Technical Interoperability Requirements Challenges	
Sub-category	**Barriers**
Codes	Culture and language:
	Economic - cost of development, high cost of interoperation
	Infrastructure
	Legal barriers: legal and business…
	Technical - lack of consistent data standards - System incompatibility
	Usability - data and interface with Legacy Systems and interface specification
Sub-category	**Threats**
Codes	Data backup and storage
	Distributed logistics
	Energy failure
	Evolve of individual systems
	Failure of assumption
	Failure of omission
	Individual systems failure
	Morphoclimatic aspects
	Network and communication failure
Sub-category	**Properties of SoS**
Codes	Collaboration between constituent systems
	Distribution of the constituent systems
	Emergent behavior of the SoS
	Evolutionary development of SoS
	Heterogeneity of the constituent systems
	Independence of the constituent systems
	Mission of the SoS

Table 36 – Singularity mission strategy

Source: Elaborated by the author.

Category: Mission Strategy Singularity	
Sub-category	**Nature of the Mission**
Codes	Alignment between enterprise mission, mission systems, and business process
	Business context
	Business scalability
	Continuous evolution of business process
	Flexibility to achieve the outcomes
	Goal management
	Market competitiveness
	Operational performance indicators
	Organization and competences
	Stakeholders roles and responsibilities
	Strategical loss and market-time
	Understanding of the mission strategy

Table 37 – Duality between interoperability requirements analysis

Source: Elaborated by the author.

Category: Duality about Interoperability Requirements Analysis	
Sub-category:	**Degree of Difficulty**
Codes	Analysis involves high costs, and resources
	It is hard to perform
	Resources and time are limited
	Testing higher priority
	Unknown of the consequences
Sub-category:	**Degree of Benefits**
Codes	Clear statement the interoperability requirements
	Enable initial identification of individual systems
	Estimate effort in initial implantation of SoS
	Gathering insight of business needs or expectations
	Know advantage about technical constraints
	Opportunities for new requirements or resources
	Prepare resources
	Prioritize capability and relate to constituents
Sub-category:	**Challenges**
Codes	Lack of a unified taxonomy
	Lack of automated technological support
	Lack of best practices for practitioners
	Lack of guidelines for the interoperability analysis
	Lack of methodological support
	Lack of support for inter-artifact and intra-artifact traceability
Sub-category	**Coalition Interoperability**
Codes	Decision-making
	Design Decision
	Interoperability barriers
	Interoperability Threats
	Network Path
	Supplementary requirements

Table 38 – Convergence between operational technologies and informational technologies

Source: Elaborated by the author.

Category: Convergence between Operational Technologies (OT) to Information Technologies	
Sub-category	**Degree of Infrastructure Control and Management**
Codes	Analysis of Physical Process
	Dependability
	Isolated technology
	IT Network
	Legacy systems integration with new systems of both
	OT Network
	Redundancies constants
	Reliability
	Robust Hardware
	Safety
	Security risks
	Share Resources
Sub-category:	**Degree of interoperability and monitoring**
Codes	Behavior unforeseen at design time
	Decision at runtime
	Evolution sometimes are negotiated
	New behavior that emerge at runtime = Emergent Behavior
	Organizational to technological
	Real time monitoring
	Systems are operational and managerial independent
	Systems are operational independent, but not managerial

Table 39 – Technical interoperability awareness

Source: Elaborated by the author.

Central Category: Technical Interoperability Awareness	
Codes	Adjust tasks with a focus on outcomes
	Business scalability
	Cost Awareness
	Decisions are made at different levels
	Demand for improvement
	Domain terminology
	Identify quite quality attribute
	Improves decision-making process
	Interoperability is indispensable in the 4.0 technology environment
	It is necessary to understand business goals
	Knowing how to use data for decision making
	Lack of investments
	Lack of quite infrastructure
	Lack of regulation

APPENDIX

N

Interviewees List

In this Appendix, we present the list of 27 interviewed of the Chapter 3. To each one, we described the ID used in theory, the subject type, the experience domain, and the number of years of experience.

Table 40 – List of interviewed

Source: Elaborated by the author.

ID	Subject Type	Experience Domain	Number of Year
IT_1	Industrial practitioners	Software Architecture and Safety Engineering	> 11
IT_2	Experts working with industrial projects	Software Arctecture, Reference Architecture, and SoS	> 11
IT_3	Experts working with industrial projects	Safety-critical System and SoS	> 11
IT_4	Industrial practitioners	Software Systems Engineening	> 7
IT_5	Industrial practitioners	Program Management	> 9
IT_6	Industrial practitioners	Senior Eletrial Engineening	> 5
IT_7	Experts working with industrial projects	Software Testing and Agile Development	> 11
IT_8	Experts working with industrial projects	Requirements Engineering	> 40
IT_9	Industrial practitioners	Software Engineening, Project Management and research at Mobile Telecommunication	> 15
IT_10	Industrial practitioners	Software Engineement	> 13
IT_11	Experts working with industrial projects	Software Testing and Software Quality Management	>40
IT_12	Experts working with industrial projects	Software Architecture	> 12
IT_13	Industrial practitioners	Genetical enhancement	> 13
IT_14	Industrial practitioners	Agricultural robotics and Research and Development	>30
IT_15	Experts working with industrial projects	Institutional economics	> 29
IT_16	Industrial practitioners	Modeling and Simulation	>10
IT_17	Industrial practitioners	Agricola Control Analysis	>24
IT_18	Industrial practitioners	Agricultural Technical	>37
IT_19	Experts working with industrial projects	Software Quality Management and Software Engineering	>40
IT_20	Industrial practitioners	Company Management	>12
IT_21	Industrial practitioners	Innovation Management Engineening	>6
IT_22	Industrial practitioners	Information Technology Consultant	>36
IT_23	Industrial practitioners	Software Architecture	>14
IT_24	Experts working with industrial projects	Self-adaptive System	> 8
IT_25	Experts working with industrial projects	Software Product Line and Software Product Family	>30
IT_26	Experts working with industrial projects	Software architecture	>8
IT_27	Experts working with industrial projects	Requirements Engineering	>15

APPENDIX

O

Definitions of Technical Interoperability Quality Attributes for $QM2_{TI}$

This Appendix provides the definitions of quality attributes for technical interoperability. The core of such definition was extracted from (ISO/IEC-25010, 2011) and shown in Table 45. They were adapted to meet the necessities of the ATLANTA Theoretical Framework. Table 41 shows the quality attributes related to an SoS design environment whereas Table 42 depicts the quality attributes related to an operational environment.

Table 41 – Quality of design environment for the $QM2_{TI}$ technical interoperability

Source: Elaborated by the author.

Functional Suitability	**Degree at which a service or system provides functions that meet stated and implied needs when used under specified conditions**
Functional Completeness	Degree at which the set of technical functions supports all specified tasks and user's objectives
Functional Correcteness	Degree at which an infrastructure or system provides the correct results at the needed degree of precision required
Functional Appropriateness	Degree at which technologies or functions facilitate the accomplishment of specified tasks and objectives
Traceability	Degree at which data, infrastructural elements or systems has attributes that provide an audit trail of access to them and changes in the data can be identified in a specific context of use
Compliance	Degree at which the infrastructure or systems voluntarily adhere to laws, conventions, regulations and standards to maintain the systems working and available in a specific context of use
Conformance	Ability of the system and physical and communication infrastructure mandatorily to comply with standards, conventions or regulations foreseen in laws and prescriptions related to SoS infrastructure needs
Performance Efficiency	**Degree at which a system or component accomplishes its designated functions related to the amount of resources used under stated conditions**
Resource utilization	Degree at which the amounts and types of resources used by systems infrastructures meet capabilities and requirements when performing its functions
Security	Degree at which a system protects information and data, so that the degree of data access of individuals or other systems is appropriate to their types and levels of authorization
Reliability	Degree at which technical elements, a system, or a component perform specified functions under specified conditions for a specified period of time
Scalability	Degree at which an infrastractute and communication network to handle increased workloads by repeatedly applying a cost-effective strategy for extending their growth of a system horizontally and vertically
Interoperability	**Degree at which two or more systems or components exchange information and use it**
Variability	Degree at which an infrastructure, communication network, systems, or data can be changeable over time assuming a given value or function or set of values at runtime
Integrability	Ability to an architecture to combine systems, technical infrastructural elements, or both into an overall system
Cooperation	Degree at which na architecture can organize systems, or technical elements exchange requests among them for achieving a goal.
Usability	**Degree at which specifics users use an infrastructure and communication network to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use**
Learnability	Degree at which specified users use a technical environment to achieve specified learning goals on the use of a system with effectiveness, efficiency, freedom from risk, and satisfaction in a specified context of use
Operability	Degree at which a system has attributes that simplifies its operation and control
User interface aesthetics	Degree at which a user interface provides a pleasing and satisfying interaction for the user
Accessibility	Degree at which system can be used with the widest range of characteristics and capabilities towards the achieviment of a specified goal in a specified context of use

Table 41 – Quality of design environment for the $QM2_{TI}$ technical interoperability (Continuation)

Source: Elaborated by the author.

Reliability	**Degree to which technical elements, a system, or a component perform specified functions under specified conditions for a specified period of time**
Maturity	Degree at which technical elements, a system, a component meet reliability needs under normal operation
Availability	Degree at which technical elements are operational and accessible when necessary for use, so that the a system, or component can be function normally, enabling data to be retrieved by authorized users and/or applications in a specific context of use.
Fault Tolerance	Degree at which a system or component operates as intended, despite faults in hardware, software or network
Recoverability	Degree at which a system can recover data by an interruption or failure and re-establish the desired state of the system
Correctness	Degree at which a network infrastructure, hardware, system or component is free from faults in their specification, design, and implementation
Robustness	Degree at which technical elements, systems or components function correctly under stressful environmental conditions
Security	**Degree at which a product or system protects information and data, so that the persons, products or systems have the degree of data access appropriate to their types and levels of authorization**
Confidentiality	Degree at which technical elements, or system ensure data are accessible only to those authorized
Integrity	Degree at which a communication network, system or component prevents unauthorized access to, or modification of, computer programs or data
Non-repudiation	Degree at which actions or events prove to have taken place, and cannot be repudiated later
Authenticity	Degree at which the identity of a subject or resource proves to be the one claimed
Assurance	Degree at confidence that the security features, practices, procedures, and architecture of a system accurately mediate and enforce security policy.
Traceability	Degree at which data, infrastructure elements or systems has attributes provide an audit trail of access to them and changes in the data can be identified in a specific context of use
Maintainability	**Degree at effectiveness and efficiency at which infrastructural elements or system can be modified by intended maintainers**
Reusability	Degree at which an asset can be used in more than one system, or in the construction of other assets
Analyzability	Degree at effectiveness and efficiency with which it is possible to assess the impact on a system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified
Modifiability	Degree at which a system can be effectively and efficiently modified without introducing defects or degrading existing product quality
Testability	Degree at effectiveness and efficiency with which test criteria can be established for a system, or component, and tests can be performed to determine whether those criteria have been met
Portability	**Degree at effectiveness and efficiency at which a system, or component can be transferred from some hardware, software or other operational or usage environment to another**
Adaptability	Degree at which a system can effectively and efficiently be adapted to different or evolving hardware, software or other operational or usage environments
Installability	Degree at effectiveness and efficiency at which a system can be successfully installed and/or uninstalled in a specified environment
Replaceability	Degree at which a technical infrastructure element or software systems can replace another specified for the same purpose in the same environment

Table 41 – Quality of design environment for the $QM2_{TI}$ technical interoperability (Continuation)

Source: Elaborated by the author.

Understandability	**Ease with which a system can be comprehended and technical resources are suitable at both system-organizational and detailed-statement levels, enabling data to be read and interpreted by users, and expressed in appropriate languages, symbols and units in a specific context of use.**
Readability	Degree at which the information provided by systems can be easily understood by the end-user and other systems
Expressiveness	Ability of an architecture to support technical and data concepts modeling for describing configurable elements and constraints
Clarity	Ability of systems to clearly understand data and information shared, ensuring quality of coherence and communication intelligibility.
Coherency	Degree at which readers of a data return always their most recently written value of that data with expressiveness and clarity
Consistency	Degree at which data and information exchanged between systems must have uniformity, standardization, and no contradiction and be coherent with other data or parts of a system or components in a specific context of use
Sustainability	**Degree at which systems or technical infrastructure meet technological needs without compromising the future capabilities**
Evolvability	Capability of technical elements and software-systems products to evolve and continue serving their customers in a cost-effective way
Scalability	Degree at which an infrastractute and communication network to handle increased workloads by repeatedly applying a cost-effective strategy for extending their growth of a system horizontally and vertically
Loose Events Coupling	Capacity of relationship among systems in which events in various systems can occur independently
Simplicity	Degree at which the design and implementation of technical infrastructure, a system or component are straightforward and easy to understand
Generality	Degree at which a system or component performs a broad range of functions to achieve a specific mission.
Extensibility	Degree at which a software system is extended with new features and components without loss of functionality or qualities specified as requirements, increasing its storage or functional capacity
Interoperability	Degree at which two or more systems, or components can exchange information and use it over time without restricting its future use.
Maintainability	Degree at effectiveness and efficiency at which a infrastructural elements or system can be modified by intended maintainers without compromising their current and future capabilities

Table 42 – Quality of operational environment for the $QM2_{TI}$ Technical Interoperability

Source: Elaborated by the author.

Functional Suitability	**Degree at which a services or system provides functions that meet stated and implied needs when used under specified conditions**
Accuracy	Degree at which technical infrastructure, systems, or data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.
Security	Degree at which a system protects information and data, so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization
Freedom for Risk	**Degree at which a technical resource, or system mitigates the potential risk to economic status, human life, health, or the environment**
Economic Risk Mitigation	Degree at which a technical resource, or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use
Health and Safety Risk Mitigation	Degree at which a technical resource, or system mitigates the potential risk to people in the intended contexts of use
Environmental Risk Mitigation	Degree at which a technical resource, or system mitigates the potential risk to property or the environment in the intended contexts of use
Safety	Ability of systems to avoid states that cause or lead to damage, injury, or loss of life to actors in a software's environment, and to recover and limit the damage when it does enter in bad states.
Performance Efficiency	**Degree at which a system or component accomplishes its designated functions relatated to the amount of resources used under stated conditions**
Time behaviour	Degree at which the response and processing times and throughput rates of data elements and systems, performing their functions, meet requirements
Storage Efficiency	Ability of technical resources, or systems to store and manage data that consume the least amount of space with minimal or no impact on performance, thus resulting lower total operational cost.
Execution Efficiency	Degree at which the performance of the technical infrastructure, or system is monitored in their speed of processing, irrespective of resources used by technical infrastructure, or system.
Interoperability	**Degree at which two or more systems, or components can exchange information and use it**
Compatibility	Degree at which a system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.
Interoperation	Degree at which a system can organize with other systems of multiples way to having several candidates architectures
Autonomy	Degree at which a system or resources can regardless to make their own operational and managerialdecisions
Transparency	Degree at which the technical elements are clearly designed and managed with different resources, meeting the individuals needs of each system for data exchange and communication without violate the privacy of the informations
Resilience	Ability to technical elements, or systems to provide required capability in case of adversity
Sustainability	Ability to systems to interoperate using technological needs without compromising the future capabilities
Reversibility	Ability of technical elements, or systems to reverse into change or problem phase, in which technical elements, or systems undergoing the process can return to their original state
Reachability	Ability of a device or systems to know the destination of other devices or systems and communicate with them

Table 42 – Quality of operational environment for the $QM2_{TI}$ Technical Interoperability (Continuation)

Source: Elaborated by the author.

Maintainability	**Degree of effectiveness and efficiency at which a product or system can be modified by intended maintainers**
Conciseness	Degree at which technical elements can be designed with a minimum amount of resources
Awareness	Ability at a system to monitor software application used in the network for providing better functionality and more efficient use
Upgradeability	Degree at which the technical infrastructure of a system can effectively and efficiently be upgraded keeping the quality of services and their availability
Configurability	Degree at which technical elements can effectively and efficiently be configured and customized, enabling rearrangements or adjustments for meeting the needs of the system
Transparency	**Degree at which the technical elements are clearly designed and managed with different resources, meeting the individuals needs of each system for data exchange and communication without violate the privacy of the informations**
Auditability	Ability of a system to determine whethe the technical elements are suitable for the purpose, safeguarding assets, maintaining data transparency and integrity, and operating effectively to achieve the business' goals and objectives
Usability	Degree at which an infrastructure and communication network can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use
Accessibility	Degree at which users can use a system with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use
Privacy	Ability of an individual, group, or organization to determine for themselves when, what, and how the data and information collected can be communicated to others.
Reliability	Degree at which technical elements, a system, or component perform specified functions under specified conditions for a specified period of time for maintaining the individuality of data and information for exchange and communication
Composability	**Degree at which a system can be efficiently and effectively selected and architected at runtime to satisfy high-level capabilities and achieve the specific mission.**
Complexity	Degree at which technical elements, a system or component has a design or implementation that is difficult to understand, updated, and verify
Dependability	**Ability of a system or technical infrastructure to deliver services justifiably trusted, avoiding failures that are more frequent and more severe than is acceptable**
Safety	Ability of systems to avoid states that cause or lead to damage, injury, or loss of life to actors in the software's environment, and to recover and limit the damage when it does enter in bad states.
Availability	Ability of a component or service to perform its required functions at an instant or over a stated period of time
Reliability	Degree at which technical elements, a system, or component performs specified functions under specified conditions for a specified period of time with no failure
Integrity	Degree at which a communication network, system or component prevents unauthorized access to, or modification of computer programs or data
Maintainability	Degree at effectiveness and efficiency at which a infrastructural elements or systems can be modified by intended maintainers without a new failure
Confidentiality	Degree at which technical elements, or systems ensure data are accessible only to those authorized to have access

APPENDIX

P

Selection of the Technical Interoperability Quality Attributes and Definitions the $QM2_{TI}$

This Appendix provides the source of each quality attribute of technical interoperability. Table 41 and Table 42 in Appendix O show as they were adapted and their current definition.

Table 45 – Selection of quality attributes and definition and/or adaptation of their description

Source: Elaborated by the author.

Characteristics	ISO/IEC25010	ISO/IEC25012	ISO/IEC/IEEE24765	Empirical	SM
Functional Suitability	⦿				֍
Functional Completeness	⦿				֍
Functional Correcteness	⦿				֍
Functional Appropriateness	⦿				֍
Traceability		◉		√	֍
Compliance		◉		√	֍
Conformance				√	֍
Accuracy		◉		√	
Security	⦿			√	
Performance Efficiency	⦿		©	√	֍
Resource utilization	⦿			√	֍
Security	⦿			√	֍
Reliability	⦿				
Time behaviour	⦿			√	֍
Storage Efficiency				√	֍
Execution Efficiency				√	֍
Interoprability	⦿		©	√	֍
Variability				√	֍
Integrability			©	√	
Cooperation				√	
Compatibility	⦿			√	֍
Interoperation				√	
Autonomy				√	֍
Transparency				√	֍
Resilience				√	֍
Sustainability				√	֍
Reversibility				√	֍
Reachability				√	֍
Usability	⦿			√	
Learnability	⦿				֍
Operability	⦿			√	
User interface aesthetics	⦿				֍
Accessibility	⦿	◉		√	
Reliability	⦿			√	
Maturity	⦿				֍
Availability	⦿	◉		√	
Fault Tolerance	⦿			√	
Recoverability	⦿	◉		√	
Correcteness			©	√	֍
Robustness				√	֍
Security	⦿				
Confidentiality	⦿	◉		√	
Integrity	⦿			√	
Non-repudiation	⦿				֍
Authenticity	⦿				֍
Assurance				√	֍
Maintainability	⦿			√	
Reusability	⦿			√	
Analysability	⦿				֍
Modifiability	⦿			√	
Testability	⦿				
Conciseness				√	֍
Awareness				√	֍
Upgradeability				√	֍
Configurability				√	֍

Legend:

⦿	There is in 25010 and was adapted when necessary
◉	There is in 25012 and was adapted when necessary
©	The difinition cames of 24765
√	The qualities attributes are additionals gathered in interviews
֍	The quality attributes were extraded o the SMs.

Table 45 – Selection of quality attributes and definition and/or adaptation of their description (Continuation)

Source: Elaborated by the author.

Characteristics	ISO/IEC25010	ISO/IEC25012	ISO/IEC/IEEE24765	Empirical	SM
Portability	⊙	◉		√	
Adaptability	⊙			√	
Installability	⊙				🌀
Replaceability	⊙				🌀
Understandability		◉	©	√	🌀
Readability				√	🌀
Expressiveness				√	🌀
Clarity				√	🌀
Coherency				√	🌀
Consistency		◉		√	🌀
Sustainability				√	🌀
Evolvability				√	🌀
Scalability				√	🌀
Loose Coupling				√	🌀
Simplicity				√	🌀
Generality				√	🌀
Extensablity				√	🌀
Interopoperability				√	🌀
Maintanainability				√	🌀
Freedom for Risk	⊙				
Economic Risk Mitigation	⊙			√	
Health and Safety Risk Mitigation	⊙			√	
Environmental Risk Mitigation	⊙			√	
Safety				√	
Transparency				√	🌀
Auditability				√	🌀
Usability	⊙				🌀
Accessibility	⊙				🌀
Privacy				√	🌀
Reliability	⊙			√	
Composability			©	√	🌀
Complexity			©	√	🌀
Dependability				√	🌀
Safety	⊙			√	🌀
Availability	⊙			√	🌀
Reliability	⊙			√	🌀
Integrity	⊙			√	🌀
Maintainability	⊙			√	🌀
Confidentiality	⊙			√	🌀
Efficiency	⊙				
Efficiency	⊙	◉			
Effectiveness	⊙				
Effectiveness	⊙				
Satisfaction	⊙				
Usefulness	⊙			√	🌀
Trust	⊙			√	
Context Coverage	⊙				🌀
Context completeness	⊙				
Flexibility	⊙				

Legend:	
⊙	There is in 25010 and was adapted when necessary
◉	There is in 25012 and was adapted when necessary
©	The difinition cames of 24765
√	The qualities attributes are additionals gathered in interviews
🌀	The quality attributes were extraded o the SMs.

APPENDIX

Q

Quality Attributes of ISO/IEC 25010:2011 not inserted in $QM2_{TI}$

This Appendix provides the quality attributes of ISO/IEC-25010 (2011) not considered in $QM2_{TI}$ and their respective descriptions.

Table 44 – Quality attributes not considered in $QM2_{TI}$

Source: Elaborated by the author.

Characteristics	Description	ISO/IEC25010
Performance Efficiency	**Degree at which a system or component accomplishes its designated functions relatated to the amount of resources used under stated conditions**	
Capacity	Degree at which the maximum limits of a product or system parameter meet requirements	X
Compatibility	**Degree at which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment**	
Co-existence	Degree at which a product can perform its required functions efficiently while sharing a common environment and resources with other products, with no detrimental impact on any other product	X
Usability	**Degree at which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use**	
Appropriateness recognisability	Degree at which users can recognize whether a product or system is appropriate for their needs	X
User error protection	Degree at which a system protects users against errors	X
Security	**Degree at which a product or system protects information and data, so that the degree of data access of individuals or other products or systems is appropriate to their types and levels of authorization**	
Accountability	Degree at which the actions of an entity can be traced uniquely to the entity	X
Maintainability	**Degree at effectiveness and efficiency at which a product or system can be modified by the intended maintainers**	
Modularity	Degree at which a system or computer program is composed of discrete components, such that a change to in one component exerts a minimal impact on other components	X
Satisfaction	**Degree at which user needs are satisfied when a product or system is used in a specified context of use**	
Pleasure	Degree at which a user is pleased by having fulfilled their personal needs	X
Comfort	Degree at which the a user is satisfied with physical comfort	X

APPENDIX

R

Declaration of Original Authorship and List of Publications

Publications Resulting From this Thesis

- **LANA, Cristiane A.** and GUESSI, M. and ANTONINO, P.O. and ROMBACH, D. and NAKAGAWA, E.Y.: A Systematic Identification of Formal and Semi-Formal Languages and Techniques for Software-Intensive Systems-of-Systems Requirements Modeling.

  Journal: IEEE System Journal, v. 13, n.3 pp. 2201-2212, 2019, Qualis: A2. **DOI:**10.1109/JSYST.2018.2874061; ISSN:1932-8184, 2019

  Level of Contribution: High- the PhD candidate is the main investigator

- **LANA, Cristiane A.**; SOUZA, N. M.; DELAMARO, E. M.; NAKAGAWA, E. Y.; OQUENDO, F.; MALDONADO, J. C. Systems-of-systems development: Initiatives, trends, and challenges.

  Event: Latino American Symposium on Software Engineering (SLISW 2016), Valparaiso, Chile: IEEE 2016, p. 1–12. Qualis: B3. **DOI:** 10.1109/CLEI.2016.7833329

  Level of Contribution: High - the PhD candidate is the main investigator

Other Related Publications

- GRACIANO NETO, V. V. and GARCÉS, L. and **LANA, Cristiane A.** and BOSCARIOLI, C. ; FORTES, R. P. M. and NAKAGAWA, E. Y.: Interação humano-computador em um sistema-de-sistemas para tratamento doméstico de idosos com Parkinson. 2017. Relatório Técnico.

Technical Report: Available online: <<http://repositorio.icmc.usp.br//handle/RIICMC/6652>> and <<https://www.researchgate.net/publication/321749357_Interacao_humano-\computador_em_um_sistema-de-sistemas_para_tratamento_domestico_de_idosos_com_Parkinson>>

Level of Contribution: Low – the PhD candidate participated in the conduction

- SOUZA, N. M.; SIMÕES, D. D.; OLIVEIRA, L. B. R.; **LANA, Cristiane A.**; NAKAGAWA, E. Y.; MALDONADO, J. C. Exploring together software architecture and software testing: A systematic mapping.

  Event: 35th International Conference of the Chilean Computer Science Society (SCCC 2016), Valparaiso, Chile:IEEE 2016, p. 1–12. Qualis: B3. **DOI:** 10.1109/SCCC.2016.7836025.

  Level of Contribution: Medium - the PhD candidate participated in the research conducted, paper writing, and made the presentation.

- NAKAGAWA, E. Y and ALLIAN, A. and OLIVEIRA, B. R. N. and SENA, B. and PAES, C. E. B and **LANA, Cristiane A.** and FEITOSA, D. and SANTOS, D. S. and ZANIRO, D. and DIAS, D. and HORITA, F. E. A. and AFFONSO, J. F. and ABDALLA, G. and VICENTE, I. and DUARTE, L. S. and FELIZARDO, K. R. and RODRIGUEZ, L. M. G. and OLIVEIRA, L. B. R. O. and GONÇALVES, M. B. and MORAIS, M. G. and GUESSI, M. and SILVA, N. and BIANCHI, T. and VOLPATO, T. and GRACIANO NETO, V. V. and ZANI, V. A. T. and MANZANO, W. Software architecture and reference architecture of software-intensive systems and systems-of-systems: contributions to the state of the art.

  Event: 11th European Conference on Software Architecture (ECSA 2017), Canterbury, United Kingdom: ACM 2017, p. 4-11. Qualis: B1. **DOI:** org/10.1145/3129790.3129822.

  Level of Contribution: Low - the PhD candidate support the writing of part of her PhD Project

Submitted Publications

- GUESSI, M.; GARCÉS, L.; ALLIAN, A.; **LANA, Cristiane A.**; NAVARRO, E.; FELIZARDO, K.; NAKAGAWA, E. Y. Women Representativeness in Software Architecture: a Systematic Mapping.

  Journal: Journal of Systems and Software, p. 1—29, 2020. QUALIS. A2.

  Level of Contribution: Medium - the PhD candidate participated in the research conducted and paper writing.

- PASSINI, W. F.; PFEIFER, V.; LANA, Cristiane, A.; AFONSO, F. J. Systematic Mapping on Frameworks for Self-adaptive Service-oriented Applications.

Journal: The Institution of Engineering and Technology (IET), p. 1-18, 2020. Qualis B1.

Level of Contribution: Medium - the PhD candidate participated of the planning and paper writing

ICMC USP
SÃO CARLOS