



UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO



FELIPE LIMA DUARTE

UMA ABORDAGEM ORIENTADA A MODELOS
PARA ENGENHARIA DE REQUISITOS DE
SISTEMAS DE SISTEMAS

Mossoró-RN

2018

FELIPE LIMA DUARTE

**UMA ABORDAGEM ORIENTADA A MODELOS
PARA ENGENHARIA DE REQUISITOS DE
SISTEMAS DE SISTEMAS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof^o Dr. Paulo Gabriel Gadelha Queiroz

Coorientador: Prof^o Dra. Angélica Félix de Castro

Mossoró-RN

2018

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

D812a Duarte, Felipe Lima .
UMA ABORDAGEM ORIENTADA A MODELOS PARA
ENGENHARIA DE REQUISITOS DE SISTEMAS DE SISTEMAS
/ Felipe Lima Duarte. - 2018.
120 f. : il.

Orientador: Paulo Gabriel Gadelha Queiroz.
Coorientadora: Angélica Félix de Castro.
Dissertação (Mestrado) - Universidade Federal
Rural do Semi-árido, Programa de Pós-graduação em
--Selecione um Curso ou Programa--, 2018.

1. Engenharia de Requisitos. 2. Sistema de
Sistemas. 3. Engenharia Orientada a Modelos. 4.
Design Science. 5. Pesquisa-ação. I. Queiroz,
Paulo Gabriel Gadelha , orient. II. Castro,
Angélica Félix de , co-orient. III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

FELIPE LIMA DUARTE

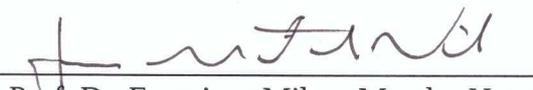
UMA ABORDAGEM DE ENGENHARIA DE REQUISITOS ORIENTADA A MODELOS PARA
SISTEMA DE SISTEMAS

Dissertação apresentada ao Programa de Pós-Graduação
em Ciência da Computação para a obtenção do título de
Mestre em Ciência da Computação.

APROVADA EM: 05,06,2018


Prof. Dr. Paulo Gabriel Gadelha Queiroz
Orientador e Presidente da Banca


Profa. Dra. Angélica Félix de Castro
Examinadora Interna - UFERSA


Prof. Dr. Francisco Milton Mendes Neto
Examinador Interno - UFERSA


Prof. Dr. Uirá Kulesza
Examinador Externo - UFRN

Agradecimentos

A Deus, por sempre ter me amparado nos momentos difíceis e por ter me dado força e paciência para concluir meu mestrado de maneira satisfatória.

Aos meus pais, Maria Ivone de Almeida Lima Duarte e José Augusto Barroso Duarte, minha irmã, Karulyne Lima Duarte, e minha esposa, Maria Nazaré dos Santos, por me apoiarem nessa jornada e em todos os aspectos da minha vida.

Aos meus colegas de mestrado: Salatiel, Weliton, Morais, Locatelli, Etienne, Paul, Jeferson, dentre outros, que sofreram comigo e me ajudaram a vencer as disciplinas do curso, principalmente PAA.

A todos os meus amigos de trabalho: André, Jonathan, Victor, João Phellipe, Israel, Alexandre, Luiz e Tullyo, que além de me apoiarem a entrar nessa jornada, promoveram um ambiente de trabalho propício para que eu conseguisse realizar o sonho do mestrado.

Ao meu orientador, Paulo Gabriel Gadelha Queiroz, que vem me ajudando, guiando e aturando desde a bolsa de monitoria, passando pelo TCC da graduação e agora no mestrado. Se hoje estou satisfeito com o trabalho realizado, foi graças as suas orientações.

À minha coorientadora, Angélica Félix de Castro, por ter me guiado durante a elaboração e execução deste trabalho e ter proporcionado um ambiente divertido nas reuniões.

A todos os professores do mestrado que contribuíram significativamente na ampliação dos meus conhecimentos e formação.

As demais pessoas que contribuíram de forma direta ou indireta para minha formação.

”O sucesso é uma consequência e não um objetivo”

Gustave Flaubert

Resumo

Um sistema de sistemas pode ser visto como um conjunto de sistemas constituintes independentes que interagem entre si com o objetivo de realizar uma missão comum, além de entregar funcionalidades complexas para o usuário final. A engenharia de requisitos é uma abordagem sistemática e disciplinada para especificação e gerenciamento dos requisitos de um *software* ou sistema. Ela tem como objetivos elicitar, modelar e gerenciar as principais funcionalidades e restrições do sistema, além de estabelecer um consenso entre os *stakeholders*. A engenharia orientada a modelos é uma abordagem de desenvolvimento de *software* que tem nos modelos as principais entidades do desenvolvimento. Os modelos podem descrever diversos aspectos do *software*, como as funcionalidades, restrições, manutenibilidade, dentre outros. Nesse contexto, o principal objetivo deste trabalho é apresentar uma abordagem de engenharia de requisitos para sistemas de sistemas que se beneficie da orientação a modelos e possa ser utilizada para definir as missões desses sistemas, além de derivar e modelar os requisitos dos seus sistemas constituintes. A condução deste trabalho foi orientada pelo paradigma *Design Science* e pela metodologia pesquisa-ação. Para evidenciar a utilização dessa última, duas versões da abordagem são apresentadas, a versão inicial, chamada de REAp-SoS; e a versão final, chamada de MORE-SoS. Para validar as abordagens, elas foram utilizadas para especificar e modelar os requisitos de dois sistemas de sistemas distintos: o primeiro é um sistema de controle e monitoramento de tráfego urbano, e o segundo é um sistema de controle e monitoramento de produção animal. As principais contribuições deste trabalho consistem na apresentação dos conceitos, fases, atividades, elementos e diretrizes da abordagem MORE-SoS, e das especificações dos dois sistemas presentes nos estudos de caso.

Palavras-chave: Engenharia de Requisitos, Sistema de Sistemas, Engenharia Orientada a Modelos, *Design Science*, Pesquisa-ação.

Abstract

A system of systems can be seen as a set of independent constituent systems that interact with each other to accomplish a common mission, as well as deliver complex functionality to the end user. Requirements engineering is a systematic and disciplined approach to specification and management of the requirements of a software or system. It aims to elicit, model and manage the main functionalities and constraints of the system, in addition to establishing a consensus among stakeholders. Model-driven engineering is a software development approach that has in the models the key entities of development. The models can describe various aspects of the software, such as features, restrictions, maintenance, among others. In this context, the main objective of this work is to present a requirements engineering approach for systems systems that benefits from model orientation and can be used to define the missions of these systems, as well as derive and model the requirements of their constituent systems. The conduction of this work was guided by the paradigm Design Science and the research-action methodology. To demonstrate the use of the latter, two versions of the approach are presented, the initial version, called REAp-SoS; and the final version, called MORE-SoS. To validate the approaches, they were used to specify and model the requirements of two distinct systems of systems: the first is an urban traffic control and monitoring system, and the second is an animal production monitoring and control system. The main contributions of this work are the presentation of the concepts, phases, activities, elements and guidelines of the MORE-SoS approach, and the specifications of the two systems present in the case studies.

Keywords: Requirements Engineering, System of Systems, Model-Driven Engineering, Design Science, Action Research.

Lista de ilustrações

Figura 1 – Metodologia de pesquisa proposta para o trabalho	22
Figura 2 – SoS de monitoramento de inundação	26
Figura 3 – Sistema integrado de defesa aérea	27
Figura 4 – Elementos principais da engenharia de sistemas para SoS	28
Figura 5 – Processo genérico de engenharia de requisitos	30
Figura 6 – Diagramas SysML	35
Figura 7 – Processo genérico da MDE	37
Figura 8 – Trabalhos que apresentam alguma abordagem de ER para SoS	44
Figura 9 – Trabalhos e suas técnicas para modelagem dos requisitos	45
Figura 10 – Trabalhos e suas respectivas fases da ER	46
Figura 11 – Processo da abordagem SoS-ACRE	51
Figura 12 – Etapas da abordagem baseada em SysML	53
Figura 13 – Abordagem REAp-SoS representada em BPMN	58
Figura 14 – Atividades da fase de concepção e modelagem da abordagem REAp-SoS	60
Figura 15 – Estrutura do SoS de controle e monitoramento de tráfego	68
Figura 16 – Missões do SoS de controle e monitoramento de tráfego urbano	69
Figura 17 – Requisitos da missão capturar e enviar informações do ambiente (parcial)	70
Figura 18 – Atividades do sistema de controle de congestionamento	71
Figura 19 – Estados do semáforo inteligente	72
Figura 20 – Abordagem MORE-SoS representada em BPMN	76
Figura 21 – Atividades da fase de concepção e modelagem	78
Figura 22 – Transformação do modelo de missões no modelo estrutural	80
Figura 23 – Transformação do modelo estrutural no modelo de requisitos	81
Figura 24 – Transformação dos modelos estrutural e de requisitos no modelo de atividades	82
Figura 25 – Influência x impacto para entidade produtores rurais	88
Figura 26 – Missões do sistema SisBovi	90
Figura 27 – Estrutura do Sistema SisBovi	91
Figura 28 – Requisitos da missão calcular anomalia (parcial)	92
Figura 29 – Requisitos da missão identificar saída do perímetro (parcial)	93
Figura 30 – Atividades do fluxo calcular anomalia	94
Figura 31 – Perfil profissional	96
Figura 32 – Viabilidade da pesquisa	97
Figura 33 – Objetivos do sistema	98
Figura 34 – Requisitos da missão capturar e enviar informações do ambiente (completo)	117
Figura 35 – Requisitos da missão calcular anomalia (completo)	118

Figura 36 – Requisitos da missão identificar saída do perímetro (completo)	119
Figura 37 – Atividades do fluxo detectar saída do perímetro	120

Lista de tabelas

Tabela 1 – Comparativo entre SoS e sistemas constituintes	24
Tabela 2 – Trabalhos encontrados na RSL	42
Tabela 3 – Trabalhos encontrados na RSL (conclusão)	43
Tabela 4 – Trabalhos e seus estudos de caso	47
Tabela 5 – Trabalhos e os desafios abordados	48
Tabela 6 – Metodologia aplicada ao estudo de caso da abordagem REAp-SoS . . .	64
Tabela 7 – Níveis de influência das entidades	67
Tabela 8 – Metodologia aplicada ao estudo de caso da abordagem MORE-SoS . .	84
Tabela 9 – Metodologia aplicada ao estudo de caso da abordagem MORE-SoS (conclusão)	85
Tabela 10 – Comparativo entre a MORE-SoS e os trabalhos relacionados	100

Lista de abreviaturas e siglas

ACRE	<i>Context-Based Requirements Engineering</i>
BPMN	<i>Business Process Model Notation</i>
CML	<i>COMPASS Modelling Language</i>
DS	<i>Design Science</i>
ER	Engenharia de Requisitos
ES	Engenharia de <i>Software</i>
ESSoS	Engenharia de <i>Software</i> aplicada a SoS
GORE	<i>Goal-Oriented Requirements Engineering</i>
GRL	<i>Goal-oriented Requirement Language</i>
Idef	<i>The Integration Definition</i>
ITS	<i>Intelligent Transport Systems</i>
IoT	<i>Internet of Things</i>
KAOS	<i>Keep All Objectives Satisfied</i>
MDE	<i>Model-Driven Engineering</i>
MG	Missões Globais
MI	Missões Individuais
MORE-SoS	<i>Model Oriented to Requirements Engineering - SoS</i>
QEP	Questão Específica de Pesquisa
QGP	Questão Geral de Pesquisa
QPP	Questão Primária de Pesquisa
REAp-SoS	<i>Requerements Engineering Approach for SoS</i>
RSL	Revisão Sistemática da Literatura
SC	Sistema Constituinte

SoS	<i>System of Systems</i>
SysML	<i>Systems Modeling Language</i>
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>

Sumário

1	INTRODUÇÃO	18
1.1	Contextualização	18
1.2	Motivação	19
1.3	Objetivo	19
1.4	Problema de Pesquisa	20
1.5	Metodologia	21
1.6	Organização do Trabalho	22
2	REFERENCIAL TEÓRICO	23
2.1	Sistema de Sistemas	23
2.1.1	Definição e Características	23
2.1.2	Tipos de SoS	25
2.1.3	Exemplos de SoS	25
2.1.4	Engenharia de Sistemas para SoS	27
2.2	Engenharia de Requisitos	29
2.2.1	Processo de Engenharia de Requisitos	29
2.3	Modelagem de Requisitos	30
2.3.1	GORE	31
2.3.1.1	Abordagens Orientadas a Objetivos	32
2.3.2	CML	33
2.3.3	Linguagem de Modelagem de Sistemas	33
2.3.3.1	Diagramas SysML	34
2.4	Engenharia Orientada a Modelos	35
2.4.1	Modelos e Metamodelos	36
2.4.2	Objetivos da Engenharia Orientada a Modelos	36
2.4.3	Processo Genérico da Engenharia Orientada a Modelos	37
2.5	Considerações Finais	37
3	REVISÃO SISTEMÁTICA E TRABALHOS RELACIONADOS	38
3.1	Engenharia de Requisitos Aplicada a SoS	38
3.1.1	Revisão Sistemática sobre ER e SoS	38
3.1.1.1	Objetivos da Revisão Sistemática	39
3.1.1.2	Questões de Pesquisa	39
3.1.1.3	Bases de Dados e <i>String</i> de Busca	40
3.1.1.4	Crítérios de inclusão e exclusão	40
3.1.1.5	Metodologia de Seleção dos Trabalhos e Extração dos Dados	41

3.1.1.6	Resultados da Revisão Sistemática	42
3.1.1.6.1	Quais abordagens, processos ou métodos de ER têm sido aplicados ao desenvolvimento de SoS?	43
3.1.1.6.2	Quais modelos de requisitos são utilizados pelas abordagens encontradas?	44
3.1.1.6.3	Quais etapas são englobadas pelos trabalhos encontrados?	45
3.1.1.7	Quais metodologias ou técnicas são utilizadas para validar essas abordagens?	46
3.1.1.7.1	Quais os principais desafios e lacunas de pesquisa da ER no contexto de SoS?	47
3.2	Trabalhos Relacionados	48
3.2.1	Abordagem SoS-ACRE	48
3.2.1.1	Ontologia	49
3.2.1.2	<i>Framework</i>	49
3.2.1.3	Processo	50
3.2.2	Abordagem Baseada em SySML para Especificação de Sistemas Complexos	52
3.2.2.1	Contexto do Sistema	52
3.2.2.2	Comportamento de Caixa Preta	54
3.2.2.3	Projeto de Arquitetura	54
3.3	Considerações Finais	55
4	REAP-SOS: UMA ABORDAGEM DE ENGENHARIA DE REQUISITOS PARA SISTEMA DE SISTEMAS	56
4.1	REAp-SoS: <i>Requirement Engineering Approach for SoS</i>	56
4.1.1	Fase de Definição do Contexto	57
4.1.2	Fase de Concepção e Modelagem	59
4.1.2.1	Identificação e Modelagem da Estrutura do SoS	59
4.1.2.2	Identificação e Modelagem das Missões do SoS	60
4.1.2.3	Identificar, Modelar e Especificar os Requisitos	61
4.1.2.4	Modelagem das Atividades do SoS	62
4.1.2.5	Modelagem dos Estados dos Sistemas Constituintes	62
4.1.3	<i>Framework</i>	62
4.2	Estudo de Caso	63
4.2.1	Metodologia Aplicada ao estudo de Caso	63
4.2.2	Contexto do Sistema de Controle e Monitoramento de Tráfego Urbano	64
4.2.3	Estrutura do Sistema de Controle e Monitoramento de Tráfego Urbano	68
4.2.4	Missões do Sistema de Controle e Monitoramento de Tráfego Urbano	68
4.2.5	Requisitos da Missão Capturar e Enviar Informações	69
4.2.6	Atividades do SoS de Controle e Monitoramento de Tráfego	70
4.2.7	Estados do Sistema Semáforo Inteligente	70
4.3	Análise da Abordagem REAp-SoS	71
4.4	Considerações Finais	73

5	MORE-SOS: UMA ABORDAGEM ORIENTADA A MODELOS PARA ENGENHARIA DE REQUISITOS DE SISTEMAS DE SISTEMAS.	74
5.1	Abordagem Orientada a Modelos para Engenharia de Requisitos de SoS	74
5.1.1	Fase de Definição do Contexto	75
5.1.1.1	Atividade de Confirmação da Existência do SoS	75
5.1.1.2	Identificação do Domínio	76
5.1.1.3	Estudo de Viabilidade	77
5.1.2	Fase de Concepção e Modelagem	78
5.1.2.1	Identificação e Modelagem das Missões do SoS	78
5.1.2.2	Identificação e Modelagem da Estrutura do SoS	79
5.1.2.3	Identificação, Modelagem e Especificação dos Requisitos	80
5.1.2.4	Modelagem das Atividades do SoS	81
5.1.2.5	Modelagem dos Estados dos Sistemas Constituintes	82
5.1.3	<i>Framework</i>	83
5.2	Estudo de Caso	83
5.2.1	Metodologia Aplicada ao estudo de Caso	84
5.2.2	Contexto do Sistema SisBovi	85
5.2.3	Concepção e Modelagem do Sistema SisBovi	89
5.2.3.1	Missões do Sistema SisBovi	89
5.2.3.2	Estrutura do Sistema SisBovi	90
5.2.3.3	Requisitos dos Sistema Constituintes	91
5.2.3.4	Atividades do Sistema SisBovi	92
5.2.3.5	Estados dos Sistema Constituintes	93
5.3	Análise do Estudo de Caso	93
5.3.0.1	Formulação do Questionário	94
5.3.0.2	Condução do questionário	95
5.3.0.3	Análise dos Resultados	95
5.4	Ameaças a Validade	99
5.5	Comparativo entre a MORE-SoS e os Trabalhos Relacionados	100
5.6	Considerações Finais	101
6	CONCLUSÃO	102
6.1	Contribuições da Pesquisa	103
6.2	Limitações da Abordagem	104
6.3	Trabalhos Futuros	104
6.4	Produções Científicas	105
	REFERÊNCIAS	106

APÊNDICES	113
APÊNDICE A – QUESTIONÁRIO APLICADO	114
APÊNDICE B – MODELOS DO SISTEMA SISBOVI	117

1 Introdução

1.1 Contextualização

Um Sistema de Sistemas (do inglês, *System of Systems* - SoS) (MAIER, 1998) pode ser definido como o resultado de um conjunto ou arranjo de sistemas independentes, chamados de Sistemas Constituintes (SC), que colaboram entre si para entregar funcionalidades e serviços não alcançáveis por um SC isoladamente (ZOPPI; CECCARELLI; BONDAVALLI, 2017). Cada SC possui um conjunto de objetivos (missões individuais) que unidos realizam os objetivos gerais (missões globais) do SoS (SILVA; BATISTA; OQUENDO, 2015).

Segundo Vargas, Gottardi e Braga (2016), os SoSs distinguem-se de outros sistemas complexos por meio de um conjunto de características, que são: (i) independência operacional e gerencial dos SCs, que podem ser gerenciados de forma independente e entregar suas funcionalidades mesmo quando não trabalham com outros sistemas; (ii) desenvolvimento evolutivo, funções e objetivos podem ser adicionados, removidos ou modificados de acordo com as necessidades dos *stakeholders*; (iii) comportamento emergente, resultante da colaboração dos SCs para entregar uma funcionalidade única e complexa; (iv) distribuição geográfica, que representa o desacoplamento físico dos SCs.

Em um processo de desenvolvimento de *software*, uma das etapas principais é a definição das funcionalidades e restrições do sistema (BENITTI, 2017). Essas definições são feitas na etapa de Engenharia de Requisitos (ER). De acordo com Paldes *et al.* (2016), é na ER que os requisitos são obtidos e refinados, com o objetivo de gerar requisitos completos, consistentes, relevantes e que atendam as necessidades do cliente.

Observa-se no desenvolvimento de um SoS, que características como independência operacional e comportamento emergente dificultam a aplicabilidade da engenharia de requisitos tradicional. Assim, para se adequar ao desenvolvimento desse tipo de sistema, as abordagens precisam passar por adaptações ou modificações.

Em relação as abordagens para o desenvolvimento de sistemas complexos, como é o caso dos SoSs, uma técnica bastante utilizada e que este trabalho pretende explorar é a Engenharia Orientada a Modelos (do inglês, *Model-Driven Engineering* - MDE) (KENT, 2002). Ela se baseia na utilização de modelos que representam formalmente aspectos do *software*, como estrutura, comportamento e necessidades dos usuários (BABUR, 2016) e (ESPERANÇA; LUCRÉDIO, 2017).

1.2 Motivação

O desenvolvimento de sistemas complexos, como os SoSs, não se trata de uma tarefa trivial, além disso, apresenta uma série de desafios, como: um número maior de etapas e atividades; envolvem uma quantidade maior de pessoas; apresentam riscos maiores, principalmente financeiros; além de envolver grandes impactos em toda a organização (ZINK, 2015). No tocante a aplicação da ER para SoS, também são encontradas algumas dificuldades, como: natureza dos requisitos, que são fragmentados, conflitantes, instáveis e em constante modificação; além da grande quantidade de *stakeholders* envolvidos (NCUBE, 2011), (SAFWAT; SENOUSY, 2015).

Com a crescente evolução tecnológica e conectividade entre diversos aparelhos (*smartphones*, computadores, fornos elétricos, televisões) por meio da *internet*, muitos SoSs estão surgindo ou já existem, mesmo que ainda não sejam reconhecidos como tal. Diante disso, observa-se que essa área da ciência da computação tende a ganhar cada vez mais relevância. Assim, se faz necessário estudar e propor novas estratégias de desenvolvimento, integração e manutenção.

Com o objetivo de identificar abordagens de ER aplicadas aos SoSs e lacunas de pesquisa, em 2016, foi executada uma Revisão Sistemática da Literatura (RSL) (BIOLCHINI *et al.*, 2005). As seguintes lacunas foram observadas: a maioria das abordagens não trabalha com todas as etapas da ER; e existe uma falta de abordagens com objetivo de derivar e especificar os requisitos dos SCs, pois grande parte delas apresentam maior preocupação com a identificação e modelagem das missões do SoS.

Portanto, os motivos da escolha da ER e dos SoSs como alvos deste trabalho são:

- A importância da ER para o processo de desenvolvimento de *software*, sendo essa a responsável pela definição das funcionalidades e do planejamento inicial do sistema. Sabe-se que uma ER bem executada evita erros e diminui custos em outras etapas do desenvolvimento (POHL; RUPP, 2011);
- A relevância e crescimento dos SoSs evidenciados por meio da RSL;
- As lacunas de pesquisa em relação a ER para SoS observadas pela RSL;
- A necessidade de técnicas, abordagens e ferramentas de ER específicas para SoS.

1.3 Objetivo

Diante do contexto e das motivações apresentadas nas seções anteriores, pode-se definir o objetivo geral deste trabalho como a realização de um estudo sobre ER e SoS, para propor uma abordagem de ER específica para SoS que seja capaz de especificar e

modelar as missões e requisitos dos SoSs e dos seus SCs. Para alcançar o objetivo geral deste trabalho, alguns objetivos específicos foram definidos:

- Identificar o estado da arte e as lacunas de pesquisa sobre ER e SoS;
- Definir uma abordagem de ER para SoS que seja capaz de especificar e modelar as missões e requisitos dos SoSs e dos seus SCs;
- Validar a abordagem por meio de um estudo de caso.

1.4 Problema de Pesquisa

Este trabalho é orientado pelo paradigma *Design Science* (DS), proposto por Hevner *et al.* (2004), segundo o qual o problema de pesquisa pode ser decomposto em problemas práticos e de conhecimento. Neste trabalho, eles estão representados como questões de pesquisa, definidas a partir dos objetivos. A Questão Geral da Pesquisa (QGP) a ser tratada é:

QGP: como executar a engenharia de requisitos para SoS?

Questões de pesquisa podem ainda ser decompostas em Questões primárias de Pesquisa (QPP) e Questões Específicas de Pesquisa (QEP), além disso, elas também podem ser classificadas como conceituais e práticas (WIERINGA, 2014). A seguir, são apresentadas as questões resultantes da decomposição da QGP descrita anteriormente:

- QPP1 (Questão Conceitual): qual o estado da arte e as lacunas de pesquisa sobre ER para SoS?
 - QEP1.1 (Questão Conceitual): quais abordagens ou métodos podem ser usados na ER para SoS?
 - QEP1.2 (Questão Conceitual): as abordagens compreendem todo o processo de ER para SoS?
 - QEP1.3 (Questão Conceitual): quais são os principais desafios da ER para SoS?
- QPP2 (Questão Prática): como realizar a ER de um SoS?
 - QEP2.1 (Questão Prática): como identificar e definir o ambiente ao qual o SoS está inserido?
 - QEP2.2 (Questão Prática): como identificar e modelar as missões de um SoS?
 - QEP2.3 (Questão Prática): como identificar e modelar os SCs de um SoS?

- QEP2.4 (Questão Prática): como derivar e modelar os requisitos dos SCs de um SoS?
- QPP3 (Questão Prática): como validar e avaliar uma abordagem de ER para SoS?

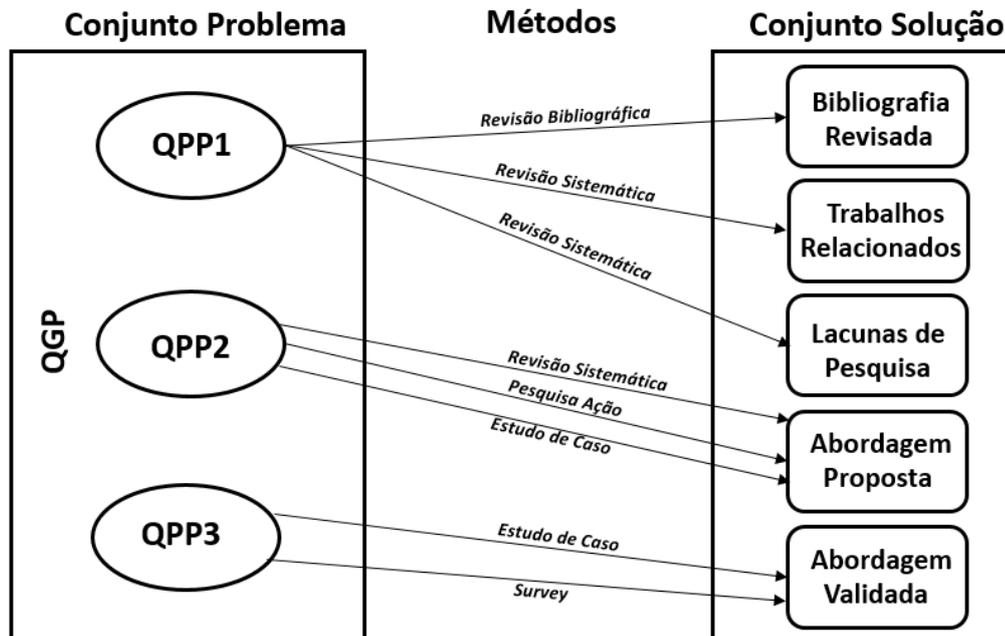
1.5 Metodologia

Como apresentado anteriormente, este trabalho é orientado pelo paradigma *Design Science*. Ele é fundamentalmente uma técnica de resolução de problemas que procura criar inovações que definam ideias, práticas e produtos por meio dos quais a análise, o projeto e a implementação podem ser efetivamente e eficientemente realizados (HEVNER *et al.*, 2004). Além disso, nessa técnica, métodos específicos da pesquisa devem conduzir o conjunto problema ao conjunto solução (WIERINGA, 2014). Na Figura 1, é apresentada a metodologia utilizada neste trabalho, assim como os métodos escolhidos para partir do conjunto problema e alcançar o conjunto solução.

Os métodos utilizados para alcançar os objetivos deste trabalho são:

- Revisão bibliográfica: realização de pesquisas por artigos e relatórios técnicos, em bases como *ACM Digital Library* e *IEEE Xplore Digital Library*, com o objetivo de obter as principais informações que norteiam o estudo dos SoSs, como por exemplo: conceitos, tipos de SoSs e características;
- Revisão Sistemática da Literatura: pesquisa executada por meio de um processo sistemático e formal. O objetivo deste método é encontrar abordagens de ER específicas para SoS e identificar as principais lacunas de pesquisa da área;
- Pesquisa-Ação: método responsável pela concepção da abordagem. Com base nos trabalhos selecionados pela RSL, uma versão inicial da abordagem deve ser concebida. Após essa concepção inicial, inicia-se um processo iterativo e incremental, a partir da qual a abordagem deve ser refinada por meio de um estudo de caso;
- Estudo de Caso: método responsável por investigar um fenômeno por meio de um caso específico. Neste trabalho, os requisitos de um SoS deverão ser especificados com a utilização da abordagem proposta;
- *Survey*: método utilizado para avaliar a abordagem proposta. As respostas obtidas são comparadas às funcionalidades modeladas para verificar se os objetivos identificados realmente satisfazem às necessidades dos *stakeholders*.

Figura 1 – Metodologia de pesquisa proposta para o trabalho



Fonte: Adaptado de Wieringa (2014)

1.6 Organização do Trabalho

O restante deste trabalho está organizado conforme apresenta-se a seguir. No capítulo 2, são apresentados conceitos sobre SoS, engenharia de requisitos, técnicas de modelagem e MDE. No capítulo 3, são apresentados o protocolo e os resultados da revisão sistemática, cujo objetivo foi identificar as abordagens de ER para SoS e as lacunas de pesquisa da área, além dos trabalhos relacionados. No capítulo 4, é apresentada a versão inicial da abordagem proposta neste trabalho (REAp-SoS), além de um estudo de caso sobre um sistema de controle e monitoramento de tráfego urbano. No capítulo 5, são apresentados os conceitos da abordagem MORE-SoS, versão final e melhorada da abordagem REAp-SoS; um estudo de caso sobre um sistema de controle e monitoramento da produção animal e a análise do *survey* realizado com o objetivo de avaliar a abordagem MORE-SoS. No capítulo 6, são apresentadas as conclusões e considerações finais deste trabalho.

2 Referencial Teórico

Neste capítulo, são apresentados os conceitos teóricos relevantes para o desenvolvimento deste trabalho, tais como: sistema de sistemas, engenharia de requisitos, modelagem de requisitos e engenharia orientada a modelos.

O capítulo está organizado da seguinte forma: na seção 2.1, são apresentados temas relevantes sobre sistema de sistemas, tais como: definição, características, exemplos e tipos; na seção 2.2, são apresentados conceitos sobre engenharia de requisitos; na seção 2.3, são apresentados conceitos e técnicas para a modelagem dos requisitos de sistemas complexos; por fim, na seção 2.4, são apresentados conceitos sobre engenharia orientada a modelos.

2.1 Sistema de Sistemas

Com o avanço da tecnologia, os sistemas computacionais estão realizando tarefas cada vez mais complexas, como por exemplo: locomoção autônoma de veículos, monitoramento de desastres naturais e concepção e manutenção de satélites. Para a realização dessas funcionalidades, diversos sistemas devem colaborar entre si, o que dá origem ao termo sistema de sistemas.

2.1.1 Definição e Características

Um SoS pode ser visto como uma classe de sistemas composta por um conjunto de sistemas constituintes independentes, que interagem entre si com o objetivo de realizar uma missão comum (ABDALLA *et al.*, 2015).

Outra definição é apresentada por Maia *et al.* (2014), que relaciona o conceito de SoS ao de Internet das Coisas (do inglês, *Internet of Things* - IoT). Essa última pode ser vista como um cenário em que inúmeros objetos (*smartphones*, *tablets*, televisores, eletrodomésticos) podem ser controlados, monitorados e gerenciados por meio da *internet*, que é a responsável pela interconexão entre eles. Juntos, esses elementos fornecem diversas funcionalidades e benefícios para o usuário. O SoS consiste na realização desse cenário, ou seja, seria a integração e operação de um conjunto de sistemas independentes que têm como objetivo entregar uma funcionalidade complexa para o usuário final (MAIA *et al.*, 2014).

Como mencionado no capítulo 1, cinco características distinguem um SoS de um sistema complexo tradicional, são elas: independência operacional e gerencial dos SCs, distribuição geográfica, desenvolvimento evolutivo e comportamento emergente. Além dessas, outra característica que está presente nos SoSs é a dinâmica inerente. Ela preconiza

que o SoS pode ser submetido a mudanças em tempo de execução, ou seja, o SoS pode ser operado e reconfigurado sem a necessidade de interrupção do seu funcionamento (CAVALCANTE *et al.*, 2016).

Além das características citadas acima, Firesmith (2010), ao analisar diversas definições de SoS, resumiu os conceitos em comum citados por diversos autores, dividindo-os em características do SoS e dos SCs. Para ele, os atributos dos SoSs como um todo são: comportamento emergente, complexos, flexíveis, evoluídos dinamicamente, dentre outros. Os atributos dos SCs são: independência operacional e gerencial, orientados a tarefas, dedicados, autônomos, pertencentes a múltiplos domínios e heterogêneos em termos de tecnologia.

Também no intuito de melhor compreender o que é um SoS, um comparativo expondo algumas diferenças entre os SoSs e seus SCs foi apresentado por Department of Defense (2008). Na Tabela 1, apresenta-se as principais características listadas pelo comparativo.

Tabela 1 – Comparativo entre SoS e sistemas constituintes

Aspectos	Sistema Constituinte	System of System
<i>Stakeholders</i>	Conjunto claro de <i>stakeholders</i> .	Os <i>stakeholders</i> de todos os sistemas têm interesses e prioridades concorrentes, além disso, em alguns casos eles podem não ter interesse no SoS.
Foco Operacional	Projetado e desenvolvido para atender aos objetivos operacionais.	Atende a um conjunto de objetivos operacionais por meio dos SCs, porém, eles podem não ter seus objetivos alinhados com os do SoS.
Teste e Evolução	Testes e avaliações do sistema são geralmente possíveis.	Os testes são mais difíceis devido à dificuldade de sincronizar os diversos ciclos de vida dos SCs. Devido a sua complexidade, dificuldades são mais propensas de acontecer.
Limites e Interface	Concentra-se nos limites e interface de um único sistema.	Foco na identificação dos sistemas que contribuem para os objetivos do SoS, permitindo o fluxo de dados entre eles, enquanto balanceia suas necessidades.
Desempenho e Comportamento.	O desempenho do sistema visa atingir objetivos específicos.	O desempenho precisa atender às necessidades dos usuários do SoS enquanto equilibra o comportamento e o desempenho dos sistemas constituintes.

2.1.2 Tipos de SoS

Dependendo do contexto ao qual está inserido, os SoSs podem assumir diferentes formas. De acordo com Maier (1998 apud VARGAS; GOTTARDI; BRAGA, 2016), os SoSs podem ser classificados em:

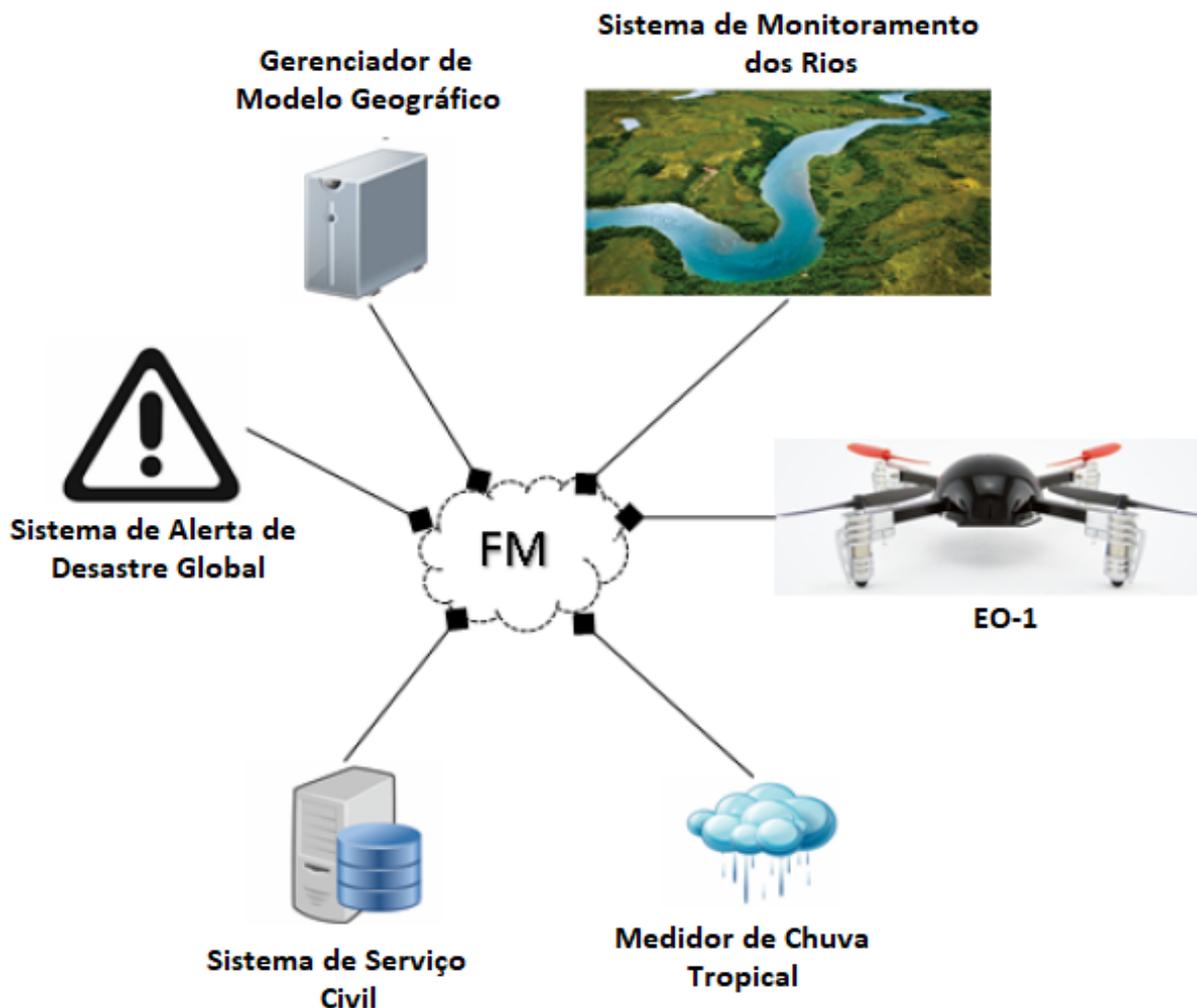
- Virtual: são SoSs que carecem de uma autoridade gerenciadora central. Os sistemas que compõem o SoS são distribuídos e geridos de forma independente;
- Colaborativo: nesse arranjo, os SCs colaboram de forma voluntária para atingir um objetivo em comum. Eles são colaboradores e decidem coletivamente como fornecer ou negar um serviço específico;
- Reconhecido: os SCs, mesmo independentes, reconhecem os objetivos e metas do SoS, assim, eles destinam recursos específicos para alcançar esses objetivos;
- Dirigido: este tipo de SoS é composto de sistemas criados para uma finalidade específica. Os SCs são geridos de forma centralizada e, quando integrados com outros sistemas, são subordinados a uma autoridade central.

2.1.3 Exemplos de SoS

Um exemplo de SoS é apresentado por Silva, Batista e Oquendo (2015), que trata de um sistema para o monitoramento e controle de inundações. O principal objetivo desse SoS é prever e alertar as autoridades quando uma inundação é detectada. Na Figura 2, apresenta-se a organização desse SoS, *Flood Monitor* (FM), com seus respectivos SCs, que estão descritos a seguir:

- Medidor de chuva tropical: produz modelos de chuvas com base em imagens feitas por satélites;
- Sistema de captura de imagens avançadas (EO-1): responsável por obter mais imagens da área, além de refinar as imagens obtidas pelos satélites;
- Sistema de alerta de desastre global: sistema responsável por alertar as autoridades quando detectada uma catástrofe natural iminente;
- Sistema de monitoramento dos rios: sistema responsável por monitorar os níveis dos rios;
- Sistema de serviço civil: fornece informações sobre edifícios e casas de uma determinada região;
- Gerenciador de modelo geográfico: produz e gerencia modelos sobre solo e terreno da região que se deseja monitorar.

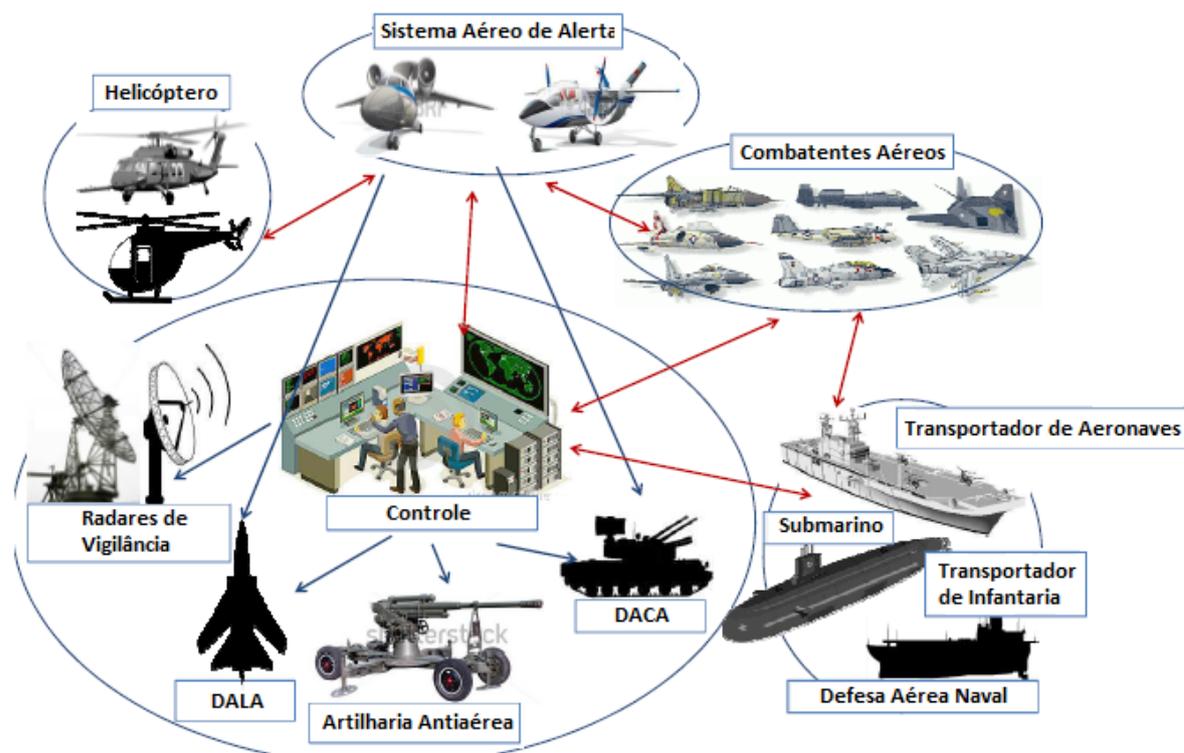
Figura 2 – SoS de monitoramento de inundação



Fonte: Adaptado de Silva, Batista e Oquendo (2015)

Outro exemplo de SoS é o sistema integrado de defesa aérea apresentado por Khlif *et al.* (2014), que o define como um SoS dirigido com SCs altamente independentes. Conforme ilustra-se na Figura 3, ele é formado por três sistemas, que são: força terrestre, composto por radares de vigilância, sistema de controle, artilharia antiaérea, Defesa Aérea de Curto Alcance (DACA) e Defesa Aérea de Longo Alcance (DALA); força marítima, composto por submarinos, transportador de infantaria e transportador de aeronaves; por fim, a força aérea, composto por helicópteros, combatentes aéreos e sistema aéreo de alerta.

Figura 3 – Sistema integrado de defesa aérea



Fonte: Adaptado de Khlif *et al.* (2014)

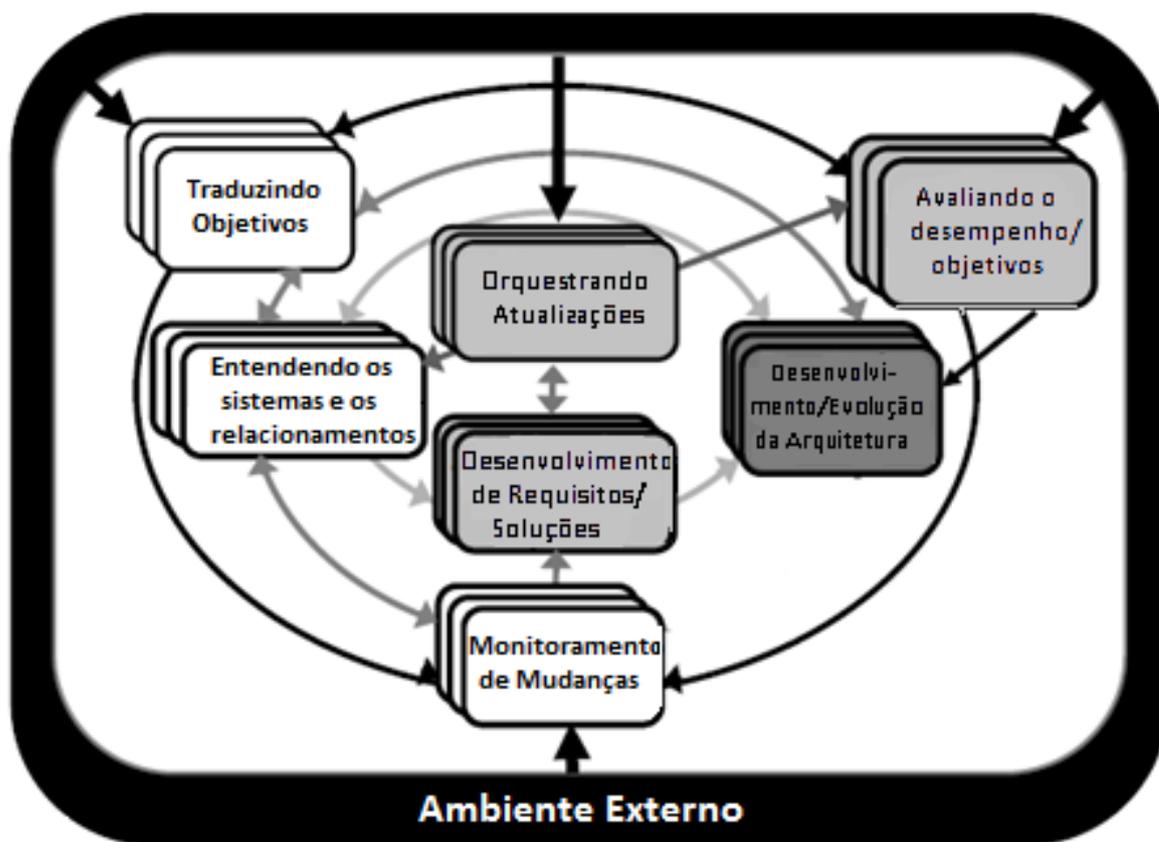
2.1.4 Engenharia de Sistemas para SoS

Nem sempre os SoS são concebidos como tal, na maioria das vezes eles tendem a sobrepor um conjunto de sistemas existentes, aperfeiçoando-os para melhor atender às necessidades dos usuários.

Uma Engenharia de *Software* aplicada a SoS (ESSoS) normalmente se concentra na evolução dos sistemas ao longo do tempo, tendo como esforços iniciais o melhoramento da forma como os sistemas trabalham juntos, antecipando mudanças, efeitos externos, internos e, eventualmente, adicionando novas funcionalidades. Em alguns casos, trabalha-se com a eliminação de sistemas, ou execução de um processo de reengenharia para melhorar a performance, capacidade e eficiência do SoS. Porém, fornecer funcionalidades mais eficientes pode ser problemático, pois recursos redundantes em vários sistemas podem ser necessários para o seu funcionamento (Department of Defense, 2008).

Dado esse contexto, há um conjunto de elementos centrais da ESSoS que descrevem as áreas chaves dessa engenharia, como podem ser vistos na Figura 4. Elas refletem as áreas nas quais os engenheiros de sistemas irão concentrar suas atenções, acompanhando como os sistemas existentes e os novos trabalharão juntos para alcançar os objetivos do usuário (DAHMANN *et al.*, 2008).

Figura 4 – Elementos principais da engenharia de sistemas para SoS



Fonte: Adaptado de Department of Defense (2008)

De acordo com Department of Defense (2008), a ESSoS apresenta sete elementos principais, que são:

- Traduzir objetivos: traduzir os objetivos de alto nível do SoS nos seus requisitos;
- Entender os sistemas constituintes e seus relacionamentos: desenvolvimento da compreensão dos sistemas envolvidos no fornecimento das capacidades do SoS, seus relacionamentos com os outros sistemas e também os seus limites;
- Avaliação do desempenho em relação aos objetivos: necessidade de definir métricas e métodos para avaliar até que ponto o desempenho do SoS cumpre seus objetivos;
- Desenvolvimento, evolução e manutenção da arquitetura: concepção e monitoramento da arquitetura do SoS, mantendo a integridade dos SCs;
- Monitoramento de mudanças: monitoramento e avaliação dos impactos potenciais das mudanças efetuadas nos SCs do SoS;

- Desenvolvimento de novos requisitos e soluções: assim como os sistemas tradicionais, um processo é necessário para recolher, avaliar e priorizar as necessidades dos usuários e, em seguida, avaliar soluções para responder a essas necessidades;
- Orquestrar atualizações no SoS: planejar melhorias para o SoS, monitorando e integrando as alterações feitas nos SCs a fim de melhorar o desempenho do SoS.

No processo geral apresentado anteriormente, é importante salientar que o escopo deste trabalho compreende a ER, assim, ele está situado e atua principalmente nas etapas de: traduzir objetivos, entender os sistemas constituintes e seus relacionamentos e no desenvolvimento de novos requisitos e soluções.

2.2 Engenharia de Requisitos

A Engenharia de *Software* (ES) pode ser definida como uma disciplina que discute técnicas sistemáticas e eficientes para o desenvolvimento de *software* (MALL, 2014). No processo de desenvolvimento, uma das etapas é a de definição e gerenciamento dos requisitos, representada pela ER. De acordo com Hull, Jackson e Dick (2004) ela é um subconjunto da engenharia de sistemas que se preocupa com a descoberta, desenvolvimento, rastreabilidade, análise, comunicação e gerenciamento dos requisitos, definindo o sistema em diversos níveis de abstração.

Outra definição é dada por Pohl e Rupp (2011), segundo a qual, a engenharia de requisitos é uma abordagem sistemática e disciplinada para especificação e gerenciamento dos requisitos, e tem os seguintes objetivos: definir os requisitos relevantes para o sistema, estabelecer um consenso entre os *stakeholders*; compreender e documentar as expectativas das partes interessadas, especificar e gerenciar os requisitos.

2.2.1 Processo de Engenharia de Requisitos

Segundo Sommerville (2010), o processo de ER tem como objetivo produzir um documento de requisitos que reflita as necessidades dos *stakeholders*. Ele pode ser definido por quatro atividades principais, que são:

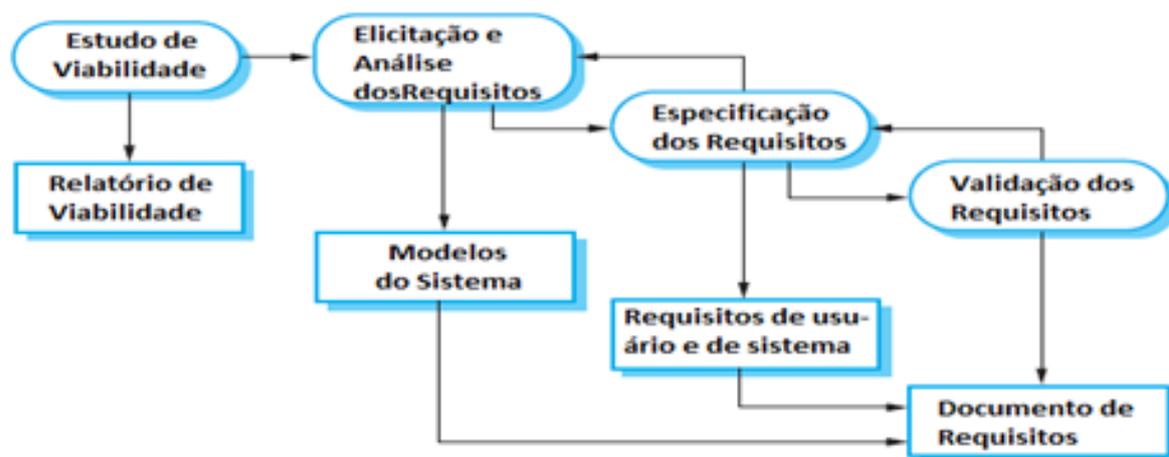
- Estudo de viabilidade: esta etapa visa identificar se as necessidades dos usuários podem ser realizadas, além disso, se o sistema proposto é rentável, dada as restrições orçamentárias existentes. O resultado desta etapa é a de continuação ou não do projeto;
- Elicitação e análise dos requisitos: etapa responsável pela obtenção dos requisitos do sistema. Os métodos para tal podem ir desde a observação de sistemas existentes

até discussões com usuários e potenciais compradores do sistema. Esta etapa pode envolver a criação de um ou mais protótipos para ajudar a compreender os requisitos;

- Especificação dos requisitos: especificação é a atividade responsável pela tradução da informação recolhida na etapa anterior em um documento que define os requisitos do sistema;
- Validação dos requisitos: esta etapa visa garantir a consistência e completude dos requisitos. Erros no documento de requisitos são eventualmente descobertos e consequentemente modificações são efetuadas para corrigi-los.

Como pode ser visto na Figura 5, a etapa de estudo de viabilidade resulta no relatório de viabilidade, a etapa de eliciação e análise dos requisitos produz os modelos do sistema, a etapa de especificação define e especifica os requisitos, podendo ter um alto grau de abstração (requisitos de usuário) ou serem mais detalhados (requisitos de sistema). Por fim, a etapa de validação é responsável por validar os requisitos. Como pode ser visto, essas etapas não são obrigatoriamente efetuadas de forma sequencial, com exceção do estudo de viabilidade, elas são conduzidas de forma intercalada (SOMMERVILLE, 2010).

Figura 5 – Processo genérico de engenharia de requisitos



Fonte: Sommerville (2010)

2.3 Modelagem de Requisitos

A modelagem dos requisitos do sistema tem como objetivo especificar o sistema a partir de diversos modelos. Um modelo pode ser definido como uma representação simplificada de um sistema, e pode ser conceitual, verbal, físico ou matemático. A modelagem permite uma melhor compreensão das questões arquiteturais e comportamentais do problema a ser resolvido. Uma boa modelagem deve permitir a representação da informação

a ser transformada pelo *software*, das funções responsáveis por essas transformações e do comportamento do sistema durante essas transformações. Um modelo, durante a etapa de requisitos, deve se concentrar no que o sistema deve fazer e não em como ele fará (SAYAMA, 2015).

Para a modelagem de sistemas tradicionais, a linguagem mais utilizada e que se tornou padrão é a *Unified Modeling Language* (UML). Ela pode ser utilizada para modelar os diversos aspectos do sistema (comportamental, estrutural, arquitetural) por meio de diversos diagramas, como por exemplo: diagramas de casos de uso, classe, componentes, interação, dentre outros (BEZERRA, 2015).

Como o foco deste trabalho é a ER e a modelagem de sistemas complexos, como SoS, as técnicas e abordagens descritas a seguir são voltadas para esse tipo de sistema, assim, não são abordados conceitos clássicos da UML.

2.3.1 GORE

A qualidade de um sistema depende criticamente do grau em que ele atende aos seus requisitos. Tais requisitos são muitas vezes concebidos, modelados e analisados como objetivos. O campo da *Goal-Oriented Requirements Engineering* (GORE) surgiu especificamente nas duas últimas décadas. Nele, os objetivos são conceitualizados em termos de modelos, que são utilizados como um meio eficaz para capturar as interações entre os requisitos (HORKOFF *et al.*, 2016).

Segundo Lamsweerde (2001) que sumarizou diversas classificações de objetivos, eles podem ser divididos em: objetivos funcionais, que se referem a serviços que o sistema deve entregar, e objetivos não funcionais, que se referem a qualidades gerais acerca do sistema. Essa classificação pode ainda ser refinada em outros tipos, como: objetivos de informação, segurança, performance, manutenibilidade, dentre outros.

Outra classificação divide os objetivos em: *soft goals*, são objetivos que não possuem critérios claros para sua realização e *hard goals*, que, ao contrário, podem ser satisfeitos e verificados por meio de técnicas claras e objetivas.

Por fim, uma última classificação aborda o tipo de comportamento do objetivo, podendo ser dividido em: *achieve goals*, que seria uma propriedade a ser eventualmente satisfeita em um futuro estado do sistema; *maintein goals*, são objetivos que descrevem uma propriedade que é permanentemente satisfeita pelo sistema. Por fim, os *optimize goals* comparam comportamentos para decidir qual atende melhor a uma propriedade do sistema.

Segundo Lapouchnian (2005) há uma série de benefícios associados a GORE, dentre eles, é possível citar:

- Os objetivos fornecem um critério para se aferir a completude da especificação dos requisitos, que estará completa se todos os objetivos do sistema podem ser alcançados por meio dos requisitos definidos;
- Os objetivos também fornecem um critério para a averiguação da relevância dos requisitos, um requisito será relevante em relação aos objetivos do domínio se sua especificação é utilizada para realizar pelo menos um objetivo;
- Os objetivos podem ajudar a fornecer uma espécie de rastreabilidade entre as metas gerais do sistema (alto nível) e requisitos técnicos (baixo nível);
- A modelagem dos objetivos promove um mecanismo natural para a estruturação de documentos complexos de requisitos.

2.3.1.1 Abordagens Orientadas a Objetivos

O i^* é um *framework* que pode ser usado na ER, engenharia de processo de negócio ou na modelagem de processo de *software*. O i^* suporta a modelagem desde as fases iniciais da ER, na qual ele é utilizado para modelar o ambiente que o sistema estará inserido, facilitando a análise do domínio e também a identificação dos *stakeholders* do sistema, além de ajudar a entender porque o sistema é necessário. Nas fases finais, o *framework* é responsável por propor novas configurações do sistema, além de modelar o projeto arquitetônico.

A modelagem do *framework* i^* é a base para a Tropos, uma metodologia de engenharia de *software* orientada a agentes que cobre todo o processo de desenvolvimento. A tropos utiliza o i^* para modelar os requisitos do sistema, além das possíveis configurações que o sistema pode apresentar. A Tropos também está associada a uma linguagem formal de especificação, chamada de *Formal Tropos*, para a adição de restrições, invariantes, pré e pós condições sobre o domínio, todos esses elementos enriquecem os modelos criados com o i^* (LAPOUCHNIAN, 2005).

Outra implementação da GORE é a KAOS (*Keep All Objectives Satisfied*), que é uma abordagem para descobrir e estruturar requisitos de um sistema. Ela se destaca por ser um dos métodos mais bem sucedidos da GORE (HEAVEN; FINKELSTEIN, 2004). Nessa abordagem, os objetivos podem ser divididos em requisitos, um tipo de objetivo a ser alcançado por um agente de *software*; expectativas, um tipo de meta a ser alcançada por um agente do ambiente; e *softgoals*, que podem ser representados por requisitos de qualidade. Na KAOS há a possibilidade de especificar conflitos entre objetivos. Além disso, outro conceito introduzido pela KAOS é o de obstáculo, que caracteriza-se como uma situação que impede a realização de um objetivo. Normalmente, a solução para o obstáculo é expresso sob a forma de um novo requisito (MONTEIRO *et al.*, 2012).

2.3.2 CML

A modelagem de SoS é desafiada por algumas características inerentes a esses sistemas, como por exemplo, a dificuldade de identificação dos SCs, sugerindo que os modelos não conseguem defini-los de maneira completa e determinística. A distribuição geográfica implica a necessidade de modelar a integração dos sistemas por meio da troca de mensagens entre eles. A constante evolução dos sistemas requer a capacidade de modelar as estruturas arquiteturais e diversas interfaces entre os sistemas.

Muitas linguagens formais foram desenvolvidas para modelar e analisar características particulares dos sistemas tradicionais. Contudo, no contexto de SoS, existe a necessidade de cobrir funcionalidades, simultaneidade, comunicação, herança, tempo e mobilidade. Porém, as linguagens formais presentes cobrem poucas dessas características.

Segundo Woodcock *et al.* (2012), A CML (*COMPASS Modelling Language*) é uma linguagem formal para a descrição e modelagem de SoS, criada para esse tipo específico de sistema. Para sua criação, os autores se basearam nos estudos descritos em Woodcock e Cavalcanti (2002), que por sua vez utilizou os formalismos descritos pela Linguagem Z, VDM (*Vienna Development Method*) e CSP (*Communication Sequential Processes*).

A CML baseia-se na SysML (*Systems Modeling Language*) para descrever a arquitetura do sistema, na CSP para descrever como se dá a comunicação entre os sistemas constituintes, e na VDM para modelar os dados e as funcionalidades. Ela estendeu a SysML para poder expressar rigorosas interfaces de contratos, dando aos engenheiros de SoS a habilidade de experimentar diferentes aspectos arquiteturais (FITZGERALD; BRYANS; PAYNE, 2012).

Nessa linguagem, um contrato descreve as garantias que um modelo CML deve prover, que por sua vez, caracteriza-se como uma coleção de definições de processos. Cada processo encapsula um estado e as operações descritas na VDM e interage com o ambiente por meio de comunicações síncronas, utilizando CSP. Woodcock *et al.* (2014) descreve a semântica da linguagem.

2.3.3 Linguagem de Modelagem de Sistemas

A Linguagem de Modelagem de Sistemas (do inglês, *Systems Modeling Language* - SysML) é uma linguagem para modelagem de propósito geral, que tem por objetivo especificar, analisar, projetar e verificar sistemas complexos, esses, que podem incluir *hardware*, *software*, sistemas embarcados, dentre outros. A SysML permite que os engenheiros modelem os requisitos, o comportamento e a estrutura do sistema. Embora os requisitos sejam tradicionalmente modelados pelo diagrama de casos de uso, que tem foco no comportamento dos requisitos, o diagrama de requisitos introduzido pela SysML permite estruturar os relacionamentos entre os requisitos a serem modelados (WEILKIENS,

2011).

A SysML é a linguagem utilizada pelos usuários da MDE quando querem criar modelos para visualização e comunicação entre os projetistas do sistema ou entre esses e os *stakeholders*. A SysML não é uma linguagem independente. Na realidade, ela é uma extensão da UML, que foi criada para ser a linguagem padrão para a modelagem de sistemas. Porém, muitos engenheiros viram que a UML não era capaz de atender a todos os aspectos de engenharia, assim, como forma de complementar alguns elementos, surgiu a SysML. É importante ressaltar que como a SysML é uma extensão da UML, algumas de suas regras são especificadas pela UML. Portanto, a especificação da SysML sozinha não é suficiente para definir a linguagem (DELLIGATTI, 2013).

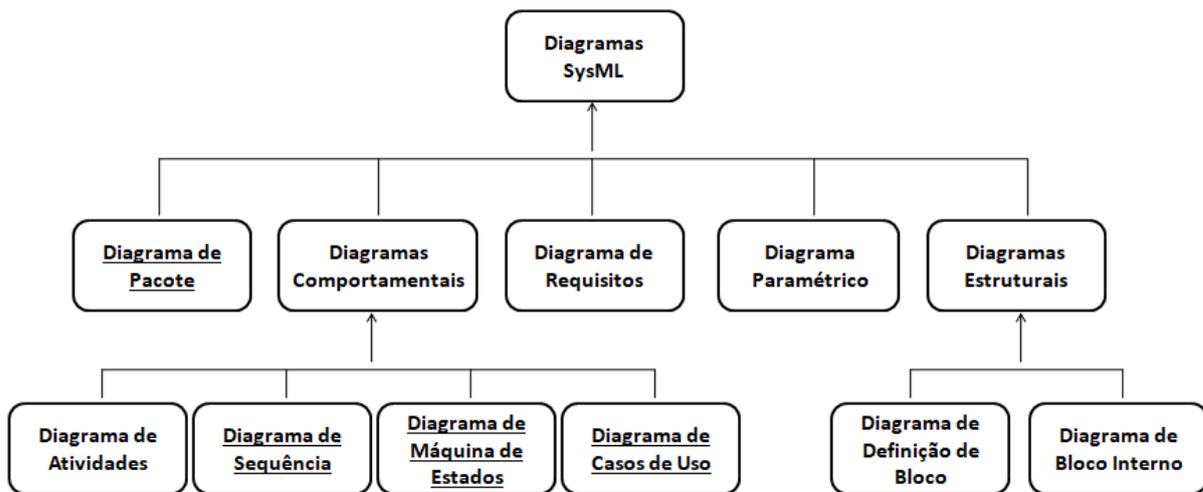
2.3.3.1 Diagramas SysML

De acordo com Friedenthal, Moore e Steiner (2011), a SysML apresenta nove diagramas, como pode ser visto na Figura 6. Observa-se que os diagramas sublinhados estão presentes na UML e não foram modificados pela SysML. A seguir, apresenta-se um breve resumo sobre cada um dos diagramas.

- Diagrama de pacote: usado para apresentar a organização dos modelos em termos de pacotes (Mesmo diagrama da UML);
- Diagrama de requisitos: usado para apresentar os requisitos e seus relacionamentos com outros requisitos, elementos de projeto e casos de teste, útil para a rastreabilidade dos requisitos (Não presente na UML);
- Diagrama de atividade: usado para apresentar o comportamento do sistema, representando o fluxo necessário para transformar as entradas em saídas por meio de uma sequência de ações (Modificado da UML);
- Diagrama de sequência: usado para apresentar o comportamento do sistema em termos da sequência de mensagens entre as diferentes partes do sistema (Mesmo da UML);
- Diagrama de máquina de estado: usado para apresentar o comportamento de uma entidade por meio dos estados que suas instâncias podem assumir (Mesmo da UML);
- Diagrama de casos de uso: usado para apresentar as funcionalidades do sistema por meio da representação de como o sistema é utilizado por entidades externas (Mesmo da UML);
- Diagrama de definição de bloco: usado para apresentar os elementos estruturais do sistema, chamados de blocos e suas composições e classificações (Modificado da UML);

- Diagrama de blocos internos: usado para apresentar as intercomunicações e interfaces entre as partes de um bloco (Modificado da UML);
- Diagrama paramétrico: usado para apresentar as restrições impostas aos elementos estruturais do sistema, que são construídas geralmente por meio de equações (Não presente na UML).

Figura 6 – Diagramas SysML



Fonte: Adaptado de Friedenthal, Moore e Steiner (2011)

2.4 Engenharia Orientada a Modelos

A MDE é uma abordagem de desenvolvimento de *software* que tem nos modelos as principais entidades do desenvolvimento, ao contrário dos códigos. Os modelos podem descrever diversos aspectos do *software*, como as funcionalidades, restrições, manutenibilidade, dentre outros (BAUDRY; NEBUT; TRAON, 2007).

A MDE utiliza modelos, notações e regras de transformações para elevar o nível de abstração dos sistemas e consequentemente aumentar sua produtividade. Seu objetivo é dividir a complexidade do sistema em diversos níveis, desde os modelos conceituais de níveis mais altos, até modelos mais específicos com informações sobre tecnologias. Como fundamentos dessa engenharia, estão: linguagens responsáveis por representar os modelos (abstrações que representam elementos do mundo real); os metamodelos, responsáveis por descrever a estrutura dos modelos; e as transformações, utilizadas para refinar e derivar os modelos em diversos níveis (PARREIRAS, 2011).

2.4.1 Modelos e Metamodelos

No contexto da MDE, modelo é visto como a simplificação de um sistema físico, ou como a representação de algum aspecto do sistema. É importante ressaltar que um sistema pode ter diversos modelos dependendo do ponto de vista, por exemplo: desenvolvedores e engenheiros podem utilizar UML e Java para representar diversos aspectos e visões do sistema. Outro aspecto interessante é a estreita relação entre os modelos e artefatos do sistema. Por exemplo, um programa Java pode ser um modelo de um sistema, como também um sistema para um determinado modelo da UML.

Enquanto modelos descrevem uma específica abstração da realidade, metamodelos são linguagens utilizadas para definir os modelos. Por exemplo, a estrutura da linguagem UML é um metamodelo, que é utilizada para descrever os diagramas da UML. As abordagens de desenvolvimento baseadas em modelos baseiam-se em uma arquitetura de modelos e metamodelos, nessa estrutura, os modelos de níveis mais baixos são os resultados dos refinamentos e transformações executadas nos modelos de níveis mais altos, o que acaba por definir uma arquitetura de diferentes níveis ou camadas (PARREIRAS, 2011).

2.4.2 Objetivos da Engenharia Orientada a Modelos

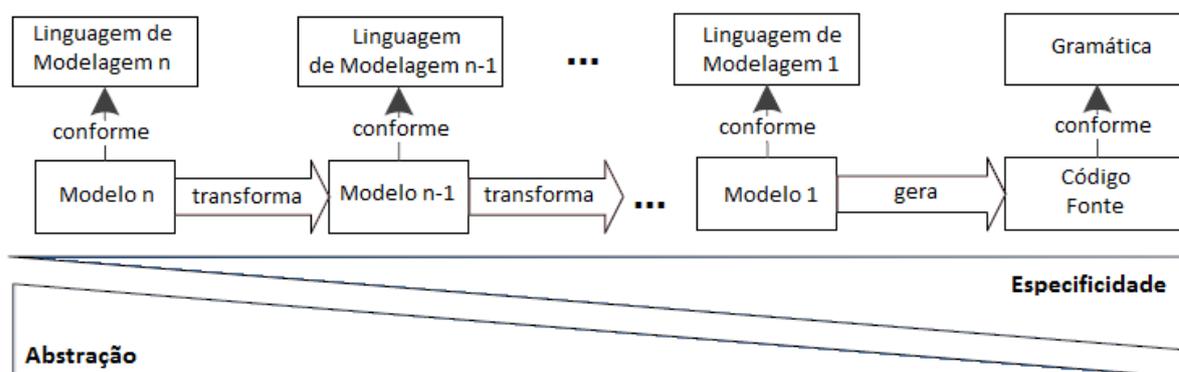
De acordo com Völter *et al.* (2013), antes de se discutir o processo da engenharia de sistemas dirigida a modelos, é importante definir quais são seus objetivos, que não são somente da MDE, mas da TI como um todo. É importante ressaltar que esses objetivos não são, necessariamente, realizáveis por essa engenharia, mas assim como outros marcos na história da TI (linguagens de alto nível, orientação a objetos), a MDE pode ser vista como mais uma que contribui para sua realização. Os objetivos são:

- Aumentar a velocidade do desenvolvimento: pode ser conseguido por meio da automatização da geração de código por meio de modelos;
- Melhorar a qualidade do *software*: as transformações automatizadas e as linguagens formalmente definidas permitem que a arquitetura seja definida uniformemente durante toda a implementação;
- Melhorar a manutenção dos sistemas: as alterações podem ser feitas em um único local, por exemplo, nas regras de transformação, isso faz diminuir os erros e evitar redundância;
- Aumentar a reusabilidade: uma vez definida a arquitetura e os modelos, as regras de transformação podem ser usadas para produzir *softwares* como uma espécie de manufatura, devido à reusabilidade.

2.4.3 Processo Genérico da Engenharia Orientada a Modelos

Segundo Staab *et al.* (2010), a MDE sugere desenvolver modelos que descrevam o sistema de forma abstrata, que, por meio de diversas etapas de transformações concebem um sistema real. Na Figura 7, ilustra-se um processo genérico de engenharia orientada a modelos, no qual, um projetista começa com a criação de um modelo n que está em conformidade com a linguagem de modelagem n , que descreve o sistema por meio de uma representação abstrata. Posteriormente, o modelo n é transformado em um modelo $n-1$, que está em conformidade com a linguagem $n-1$. Essa transformação aumenta a especificidade do sistema e simultaneamente diminui o nível de abstração, ao final do processo de transformação o código fonte do sistema pode ser gerado a partir de um determinado modelo.

Figura 7 – Processo genérico da MDE



Fonte: Adaptado de Staab *et al.* (2010)

Como pode ser visto no capítulo 5, a abordagem proposta por este trabalho é orientada pelo processo apresentado anteriormente. Partindo de um modelo inicial, um conjunto de diretrizes formais devem ser executadas para transformar e derivar os modelos requeridos pela abordagem. A principal diferença é que o resultado final do processo, no caso da abordagem, são os requisitos dos sistemas constituintes identificados e modelados.

2.5 Considerações Finais

Este capítulo abordou conceitos relevantes e necessários como fundamentação para o que se pretende propor, que é uma abordagem de engenharia de requisitos orientada a modelos aplicada a SoS. Assim, foram abordados temas como SoS, suas características, tipos e exemplos; engenharia de requisitos tradicional, modelagem de sistemas complexos e, por fim, foram apresentados conceitos sobre a engenharia orientada a modelos.

3 Revisão Sistemática e Trabalhos Relacionados

Neste capítulo, são apresentados o protocolo, os resultados da revisão sistemática da literatura executada e os trabalhos relacionados com o tema desta pesquisa.

O capítulo está organizado da seguinte forma: na seção 3.1, por meio da exposição do protocolo e resultados da revisão sistemática, são apresentados conceitos específicos sobre engenharia de requisitos aplicados ao desenvolvimento de sistema de sistemas. Na seção 3.2.1, é apresentada a abordagem SoS-ACRE, e na seção 3.2.2, são apresentados conceitos sobre uma abordagem de engenharia de requisitos baseada em SysML aplicada ao desenvolvimento de SoS.

3.1 Engenharia de Requisitos Aplicada a SoS

Os processos e abordagens da ER tradicional são definidos para dar apoio ao desenvolvimento de um sistema, muitas das principais características do sistema são definidas antes de se projetar a arquitetura e executar a implementação. Outra característica do desenvolvimento tradicional é que ele geralmente tem um único ente (usuário, empresa, cliente) que controla o desenvolvimento do sistema. Essa última característica dificulta a aplicabilidade da ER tradicional ao desenvolvimento de SoS, visto que ele é composto de inúmeros SCs, cada um com seus próprios objetivos e *stakeholders*, outra questão a se considerar é que um SC pode ser aplicado a mais de um SoS, o que pode levar a conflitos com os outros sistemas (HOLT *et al.*, 2015).

A maioria das abordagens tradicionais de ER enfocam em uma perspectiva sobre o comportamento do sistema, tentando determinar o que o sistema deve fazer. Contudo, quando se está no contexto de SoS ou de qualquer sistema complexo, as abordagens têm que se preocupar em identificar as informações necessárias e usadas pelos *stakeholders*, ao invés de determinar quais funcionalidades específicas devem ser projetadas (SOMMERVILLE; LOCK; STORER, 2012).

3.1.1 Revisão Sistemática sobre ER e SoS

Com o propósito de identificar abordagens, processos e metodologias de ER aplicadas a SoSs, além de lacunas de pesquisa. Foi executada em 2016, uma RSL. Segundo Biolchini *et al.* (2005), ela pode ser definido como uma metodologia específica de pesquisa, que visa reunir evidências pertinentes sobre um tópico específico. Diferente dos processos

usuais de pesquisa, como a pesquisa bibliográfica, a revisão sistemática é desenvolvida de maneira formal, pois segue uma sequência de passos bem definidos que são especificados no seu protocolo (BIOLCHINI *et al.*, 2005).

Primeiramente, foi definido o protocolo da RSL, que contém os objetivos, as questões de pesquisa, a estratégia de busca e os critérios de seleção e exclusão dos trabalhos. Na etapa de execução, com base nas diretrizes definidas no protocolo, foram selecionados os trabalhos desejados. Por fim, com base nesses trabalhos, as questões de pesquisa definidas no protocolo foram respondidas.

3.1.1.1 Objetivos da Revisão Sistemática

De forma geral, o objetivo da RSL foi identificar o estado da arte para o contexto de ER e SoS, assim, obtendo-se uma melhor compreensão e entendimento do domínio em questão, além disso, objetivou-se identificar abordagens, processos, desafios e lacunas de pesquisa da ER para SoS. A seguir, são detalhados esses objetivos:

- Objetivo 1: identificar abordagens, processos e métodos para a engenharia de requisitos no contexto de SoS;
- Objetivo 2: identificar as linguagens e modelos utilizados pelas abordagens encontradas;
- Objetivo 3: identificar os desafios e lacunas de pesquisa da ER no contexto de SoS;
- Objetivo 4: identificar a forma de validação das abordagens.

3.1.1.2 Questões de Pesquisa

Algumas questões de pesquisa foram criadas para alcançar os objetivos da revisão, elas foram divididas em: questão principal (QP), que é a questão mais importante e representa o principal objetivo da revisão; e, questões secundárias (QS), que complementam e adicionam valor à questão principal e também aos objetivos do trabalho. A seguir, as questões definidas são apresentadas.

- Questão Principal (QP): quais abordagens, processos ou métodos de ER têm sido aplicados ao desenvolvimento de SoS?
- Questão Secundária (QS1): quais modelos de requisitos são utilizados pelas abordagens encontradas?
- Questão Secundária (QS2): quais etapas são englobadas pelas abordagens encontradas?

- Questão Secundária (QS3): quais metodologias ou técnicas são utilizadas para validar essas abordagens?
- Questão Secundária (QS4): quais os principais desafios e lacunas de pesquisa da ER no contexto de SoS?

3.1.1.3 Bases de Dados e *String* de Busca

A pesquisa dos trabalhos foi efetuada em seis bases de dados: as quatro primeiras são as maiores bibliotecas digitais de grande relevância em ciência da computação; a quinta, é uma base brasileira, com o potencial de enriquecer a revisão; por fim, a última base é um diretório de pesquisas eletrônicas de acesso aberto, mantido pela *Lund University Libraries*, na Suécia.

- IEEE Xplorer Digital Library: <http://ieeexplore.org/>;
- ACM Digital Library: <http://dl.acm.org/>;
- Science Direct: <http://www.sciencedirect.com/>;
- Scopus: <http://www.scopus.com/>;
- Portal de Periódico Capes: <http://www.periodicos.capes.gov.br/>;
- Directory of Open Access Journals (DOAJ) <https://doaj.org/>.

Definidas as bases da pesquisa, o passo seguinte foi definir uma *string* de busca. Tentou-se englobar os sinônimos de sistema de sistemas e de engenharia de requisitos, além disso, também foram englobadas as atividades da ER, visto que podem existir abordagens que foquem apenas em algumas partes do processo. A seguir é apresentada a *string* de busca geral:

```
("system of system"OR "system of systems"OR "systems of systems"OR  
"system-of-system"OR "system-of-systems"OR "systems-of-systems"OR "SoS") AND  
("requirements"OR "requirement"OR "requirements engineering"OR "requirement  
engineering"OR "feasibility analyses"OR "requirements elicitation"OR "requirements  
specification"OR "requirements analysis"OR "requirements validation")
```

3.1.1.4 Critérios de inclusão e exclusão

Os critérios de inclusão têm como objetivo definir de forma direta, quais trabalhos estão diretamente relacionados com as questões de pesquisa. Os critérios de exclusão têm como objetivo excluir trabalhos não relacionados com o tema pesquisado e acrescentar um nível de qualidade aceitável aos trabalhos encontrados.

Os critérios de inclusão são:

- CI1: o estudo apresenta alguma abordagem, processo ou método de engenharia de requisitos para o contexto de SoS;
- CI2: o estudo aborda algum modelo ou linguagem para representar requisitos;
- CI3: o estudo traz alguma validação das abordagens, processos ou métodos propostos;
- CI4: o estudo traz desafios da ER no contexto de SoS.

Os critérios de exclusão são definidos a seguir. Observa-se que os critérios CE1, CE2, CE3 estão relacionados com o tema da RSL, já os critérios CE4, CE5, CE6, CE7 e CE8 estão relacionados à qualidade dos trabalhos.

- CE1: o estudo não trata de SoS;
- CE2: o estudo não trata de requisitos no contexto de SoS;
- CE3: o estudo trata de requisitos, mas não como foco principal do trabalho;
- CE4: o estudo não está escrito na língua inglesa ou portuguesa;
- CE5: o estudo é duplicado, a versão mais nova será incluída;
- CE6: o estudo é um *short paper* ou resumo;
- CE7: o estudo é um editorial, opinião, tutorial, pôster, painel, revisão de conferência ou livro;
- CE8: o estudo é anterior ao ano de 2006.

3.1.1.5 Metodologia de Seleção dos Trabalhos e Extração dos Dados

A metodologia utilizada na RSL consiste na definição dos passos que são executados, com o objetivo de coletar evidências que respondam aos questionamentos propostos. Na RSL, quatro atividades distintas foram executadas, conforme apresenta-se a seguir:

- Busca: a *string* de busca é aplicada em cada base e todos os trabalhos retornados são coletados;
- Seleção preliminar: nessa etapa, o título e resumo dos trabalhos são lidos. Nesse processo, diversos trabalhos são excluídos com base nos CEs;
- Seleção final: os trabalhos resultantes foram lidos completamente e novamente foram aplicados os critérios de inclusão e exclusão.

- Extração dos dados: após a seleção final, o formulário de extração dos dados foi preenchido para cada trabalho selecionado, as informações requeridas pelo formulário são: título, autores, ano, base de dados, atividades da ER contempladas pelo trabalho, técnica de validação, método de modelagem e desafios.

3.1.1.6 Resultados da Revisão Sistemática

Após a execução das *strings* de busca nas bases selecionadas e a aplicação dos critérios de inclusão e exclusão, trinta e dois (32) trabalhos foram selecionados para extração dos dados. Nas tabelas 2 e 3 são apresentados todos os trabalhos encontrados.

Tabela 2 – Trabalhos encontrados na RSL

id	Trabalho	Objetivo
#1	(BELLOIR <i>et al.</i> , 2014)	Utilizar a linguagem RELAX para identificar e definir as propriedades de segurança do SoS.
#2	(VIERHAUSER, 2014)	Monitorar a infraestrutura do SoS.
#3	(SOMMERVILLE; LOCK; STORER, 2012)	Propor um modelo de responsabilidade para identificar as informações necessárias para os <i>stakeholders</i> .
#4	(KOTEJOSHYER; SINGH; TITTIBHA, 2014)	Gerenciar conflitos e problemas nos requisitos de um SoS.
#5	(SAFWAT; SENOUSY, 2015)	Discutir as limitações da ER no contexto dos <i>Ultra Large Systems</i> (ULS).
#6	(FLANIGAN; BROUSE, 2013)	Propor um processo de ER para alocar os requisitos dos SCs e identificar a contribuição dos SCs para a realização dos objetivos do SoS.
#7	(FLANIGAN; BROUSE, 2012b)	Propor uma metodologia para medir o nível de contribuição dos requisitos de um SC para a realização dos objetivos do SoS.
#8	(HOLT <i>et al.</i> , 2015)	Propor uma abordagem para gerenciar todo o processo de ER.
#9	(NCUBE; LIM; DO- GAN, 2013)	Identificar os principais desafios da ER para o desenvolvimento de SoS.
#10	(CECCARELLI <i>et al.</i> , 2015)	Propor um modelo de dez pontos de vistas que devem ser observados na fase de elicitação e especificação dos requisitos de um SoS.
#11	(VIERHAUSER <i>et al.</i> , 2015)	Propor um modelo de monitoramento de requisitos.
#12	(LUO; SAHRAOUI; HESSAMI, 2015)	Propor uma metodologia para monitorar as mudanças nos requisitos de um SoS e o impacto gerado por essas mudanças.
#13	(CAVALCANTE <i>et al.</i> , 2015)	Propor uma abordagem para capturar os objetivos dos SCs e evoluir esses objetivos de tal forma que eles possam contribuir para os objetivos do SoS.
#14	(YANG <i>et al.</i> , 2009)	Propor um processo para o desenvolvimento dos requisitos de um SoS.
#15	(MOKHTARPOUR; STRACENER, 2017)	Propor uma metodologia para selecionar os SCs de um SoS.

Tabela 3 – Trabalhos encontrados na RSL (conclusão)

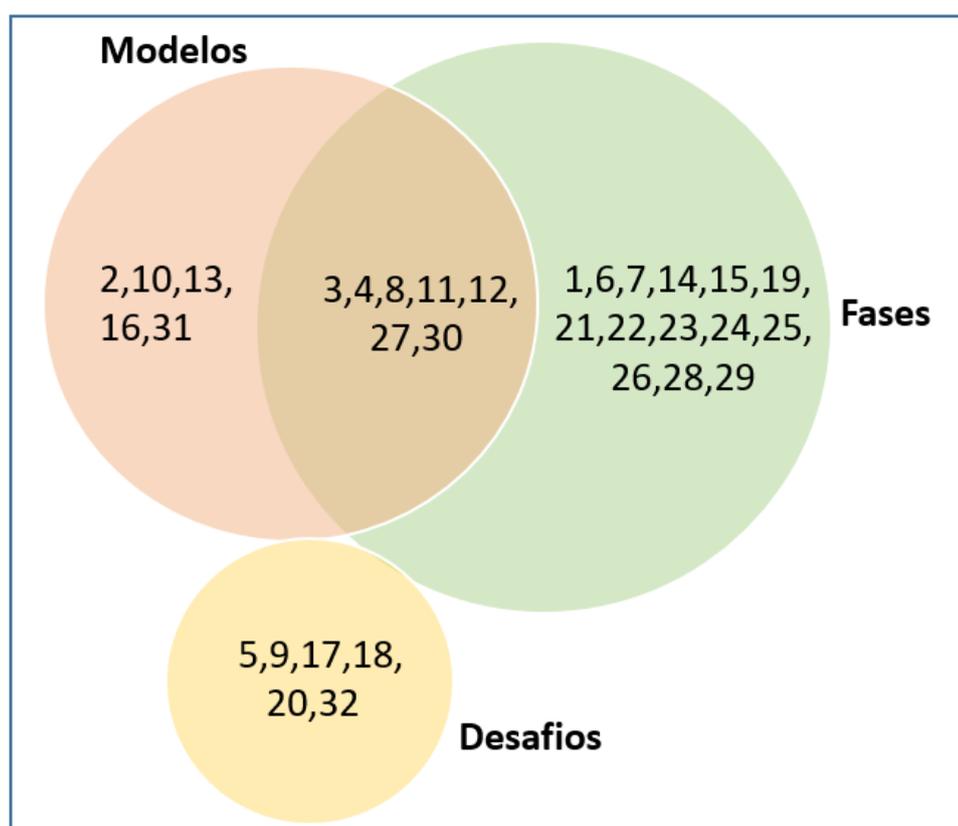
id	Trabalho	Objetivo
#16	(PENZENSTADLER; ECKHARDT, 2012)	Propor um modelo de conteúdo para elicitación e documentação dos requisitos nos diferentes níveis de abstração de um SoS.
#17	(SAVIO; ANITHA; IYER, 2011)	Discutir aspectos do processo geral de ER para SoS.
#18	(NCUBE, 2011)	Identificar os principais desafios da comunidade de ER no contexto de SoS, e descrever as atividades da ER que precisam ser trabalhadas por essa comunidade.
#19	(YANG-TURNER; LAU, 2011)	Propor uma estratégia pragmática para reunir os diferentes tipos de <i>stakeholders</i> com o objetivo de elicitar os requisitos de um SoS.
#20	(HALLERSTEDTE <i>et al.</i> , 2012)	Discutir, de maneira geral, os principais desafios encontrados na execução do projeto COMPASS.
#21	(LANE, 2014)	Propor uma abordagem para transformar os objetivos do SoS em requisitos dos SCs.
#22	(LEWIS <i>et al.</i> , 2008)	Apresentar uma visão geral da engenharia de sistemas para SoS, assim como discutir aspectos da ER.
#23	(LEWIS <i>et al.</i> , 2009)	Propor uma abordagem geral de ER para SoS.
#24	(SANTOS; LEONOR; STEPHANY, 2009)	Propor um mecanismo para executar a rastreabilidade dos requisitos de um SoS.
#25	(PETRINCA; GAMMALDI; TIRONE, 2012)	Propor uma abordagem para definição dos requisitos para sistemas complexos e SoS.
#26	(FLANIGAN; BROUSE, 2012a)	Propor uma abordagem para alocação dos requisitos de um SoS.
#27	(JOYCE <i>et al.</i> , 2011)	Propor um modelo genérico para guiar o desenvolvimento dos requisitos não funcionais de um SoS.
#28	(WALKER; KEATING, 2012)	Propor uma metodologia para derivar os requisitos de um SoS.
#29	(KATINA; KEATING; JARADAT, 2014)	Propor um <i>framework</i> para elicitar os requisitos em situações complexas.
#30	(VIERHAUSER; RABISER; GRÜNBACHER, 2014)	Propor uma infraestrutura para o monitoramento dos requisitos em <i>Very Large Scale Systems</i> (VLSS).
#31	(KLAMBAUER; HOLL; GRÜNBACHER, 2013)	Propor uma abordagem para analisar e formalizar os requisitos em <i>Multi Product Lines</i> (MLP).
#32	(LAHBOUBE <i>et al.</i> , 2014)	Propor uma abordagem para a modelagem dos requisitos em ambientes complexos baseada na linguagem SysML.

3.1.1.6.1 Quais abordagens, processos ou métodos de ER têm sido aplicados ao desenvolvimento de SoS?

Nesta seção, são mapeados os processos e abordagens de ER específicos para o desenvolvimento de SoS. Sabe-se que as abordagens tradicionais encontram dificuldades ao serem aplicadas no contexto de SoS, pois os requisitos nesse domínio são instáveis, fragmentados e conflitantes.

Foram identificados vinte e seis (26) trabalhos que apresentam alguma abordagem, processo ou metodologia de ER para SoS, o que representa oitenta e um virgula vinte e cinco por cento (81,25%) dos trabalhos encontrados. Na Figura 8, são apresentados esses trabalhos, eles estão classificados em (i) modelos, artigos que definem estruturas ou elementos a serem desenvolvidos; (ii) fases, artigos que definem abordagens e atividades para execução da ER para SoS; (iii) desafios, trabalhos que não definem abordagens, mas discutem os desafios da ER para SoS.

Figura 8 – Trabalhos que apresentam alguma abordagem de ER para SoS



Fonte: Autoria Própria

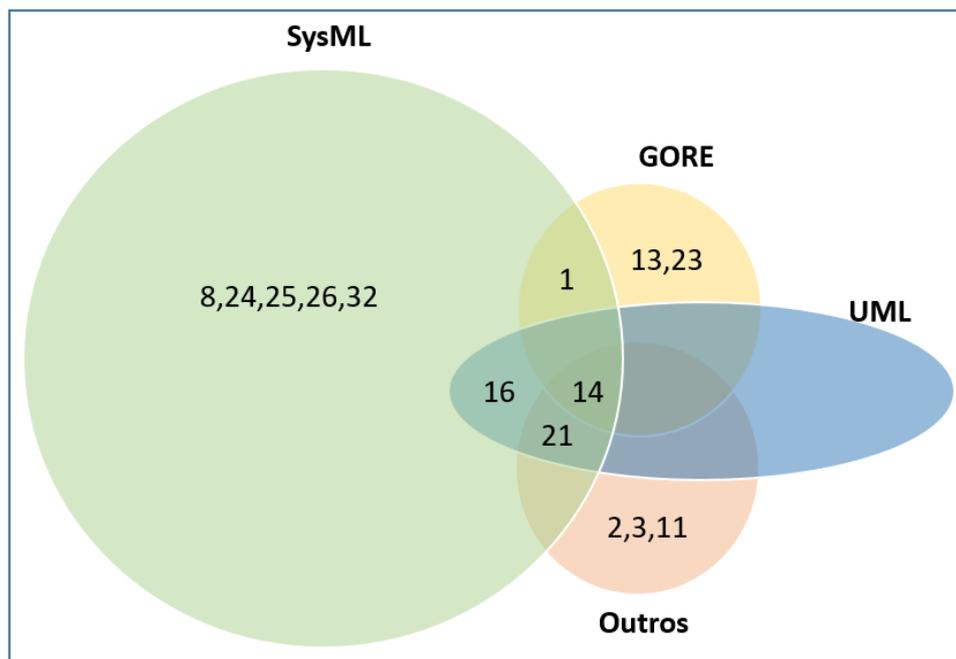
Dentre os trabalhos selecionados, os artigos #10, #15, #16, #21, #23, #25, #26, #28 e #29 se destacam, pois são trabalhos que definem claramente os seus problemas e objetivos. Além disso, eles são maduros, bem estruturados, coesos; e oferecem diretrizes que auxiliam no desenvolvimento do modelo proposto ou na execução das atividades definidas.

3.1.1.6.2 Quais modelos de requisitos são utilizados pelas abordagens encontradas?

Nesta seção, são mapeadas as linguagens utilizadas para representar os requisitos de um SoS. Nesse contexto, diversas técnicas podem ser utilizadas, dentre elas, podem-se citar: SySML, UML, *The Integration Definition (Idef)*, GORE e suas implementações

(KAOS e i*). Na Figura 9, são apresentados os trabalhos que utilizaram ou indicaram alguma linguagem para modelagem dos requisitos.

Figura 9 – Trabalhos e suas técnicas para modelagem dos requisitos



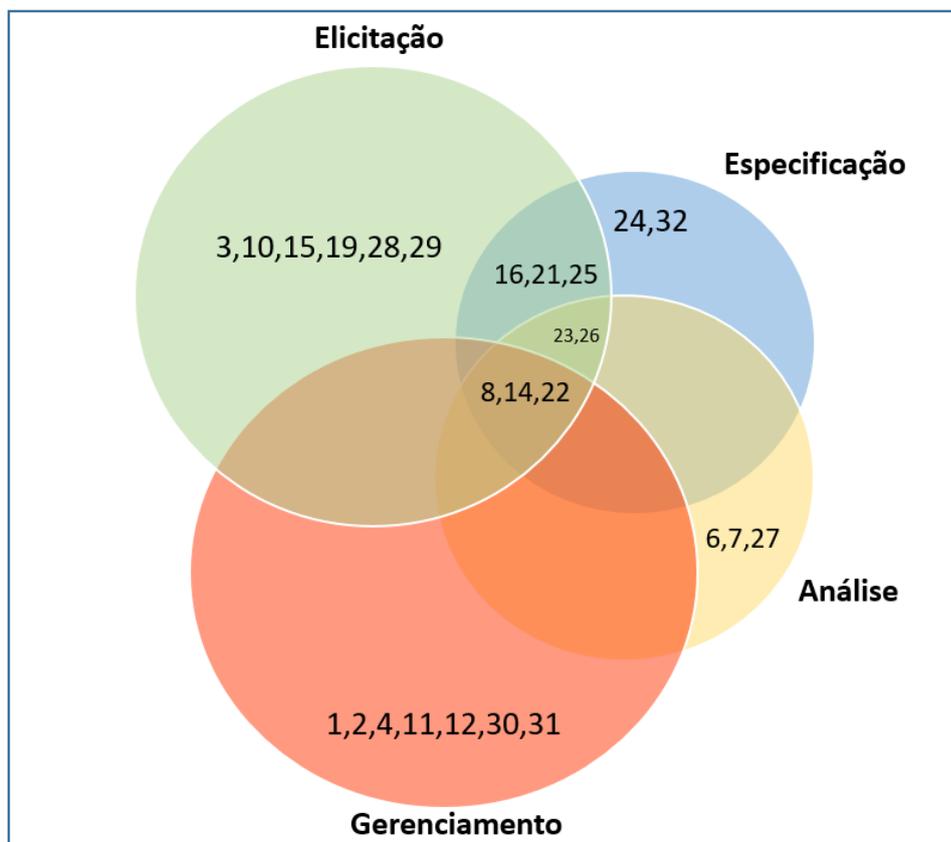
Fonte: Autoria Própria

Dos trabalhos que utilizaram ou indicaram alguma técnica ou linguagem de modelagem dos requisitos, sessenta e quatro vírgula vinte e oito por cento (64,28 %) citaram SysML, vinte e oito vírgula cinquenta e sete por cento (28,57 %) citaram o uso da GORE ou uma de suas implementações, vinte e um vírgula quarenta e dois por cento (21,42 %) citaram a UML e, finalmente, trinta e cinco vírgula setenta e um por cento (35,71 %) citaram outras formas de modelagem de requisitos, como linguagem natural ou modelos próprios. Observa-se com esses dados que a maioria dos trabalhos utilizou ou citou a linguagem SysML para a modelagem dos requisitos dos SoSs.

3.1.1.6.3 Quais etapas são englobadas pelos trabalhos encontrados?

Nesta seção, são mapeadas as etapas da ER abordadas pelos trabalhos selecionados. Desses, cinquenta e três vírgula oitenta e quatro por cento (53,84 %) abordam a elicitação dos requisitos, trinta e oito vírgula quarenta e seis por cento (38,46 %) abordam a especificação dos requisitos, trinta vírgula setenta e seis por cento (30,76 %) abordam a análise dos requisitos, trinta e oito vírgula quarenta e seis por cento (38,46 %) abordam o gerenciamento dos requisitos e, finalmente, apenas onze vírgula cinquenta e três por cento (11,53 %) cobrem todas as fases da ER.

Figura 10 – Trabalhos e suas respectivas fases da ER



Fonte: Autoria Própria

Na Figura 10, são apresentados os trabalhos selecionados e as fases da ER abordadas por eles. Como pode ser observado, poucos trabalhos abordaram todo o processo de ER, a maioria abordou as etapas de elicitação, especificação e gerenciamento dos requisitos, nessa ordem.

3.1.1.7 Quais metodologias ou técnicas são utilizadas para validar essas abordagens?

Nesta seção, são mapeadas as técnicas utilizadas para validação das abordagens propostas. De acordo com Wohlin, Runeson e St (2012), algumas técnicas podem ser formalmente utilizadas na engenharia de *software* para validar um estudo, dentre elas, podem ser citadas:

- Estudo de caso: um estudo de caso é uma pesquisa empírica que investiga um fenômeno contemporâneo dentro do seu contexto real, especialmente quando as fronteiras entre o fenômeno e o seu contexto não são claramente definidas;
- *Survey*: uma forma estruturada de coletar informações padronizadas de indivíduos utilizando questionários ou entrevistas;

- Experimento controlado: um estudo no qual um procedimento ou experimento é intencionalmente realizado e o seu resultado é observado. Ele é utilizado quando se quer controlar de forma sistemática uma determinada situação.

Dos trabalhos selecionados, quarenta vírgula sessenta e dois por cento (40,62 %) realizaram a validação dos seus estudos. Desses, 100% utilizaram a técnica estudo de caso. Alguns dos trabalhos descreveram no artigo o estudo realizado, outros trabalhos indicaram projetos reais nos quais os conceitos descritos foram aplicados. Na Tabela 4, é apresentado um mapeamento dos trabalhos e dos seus respectivos estudos de caso.

Tabela 4 – Trabalhos e seus estudos de caso

Trabalho	Estudo de Caso
#1	Sistema de segurança marítimo
#2	<i>Primetals Technologies' Plant Automation System (PAS)</i>
#3	Sistema coordenador de emergências
#4	MYGROUP Project - Sistema aéreo
#6	Sistema alfandegário
#7	Sistema de transporte de aviação naval
#8	Sistema de serviços de emergência
#10	Sistema ferroviário
#11	<i>Primetals Technologies' Plant Automation System (PAS)</i>
#24	Satélite ITASAT
#25	Sistema de vigilância marítima
#28	Cozinha inteligente
#32	Sistemas de informação hospitalar

3.1.1.7.1 Quais os principais desafios e lacunas de pesquisa da ER no contexto de SoS?

Nesta seção, são apresentadas as lacunas de pesquisa e os desafios da ER no contexto de SoS. Dos trabalhos selecionados, vinte e cinco por cento (25%) apresentaram desafios. A seguir, na Tabela 5, eles são sumarizados.

Como lacunas de pesquisa, foi observado que poucos trabalhos abordam todas as etapas da ER no contexto de SoS. Além disso, dos trabalhos que abordam mais de uma etapa, não houve um detalhamento das atividades propostas. Outra lacuna observada foi a falta de abordagens com o objetivo de derivar os requisitos dos SCs, muitas delas trabalhavam apenas com as missões do SoS. Assim, as principais lacunas de pesquisa identificadas foram: a falta de abordagens que trabalhem com todas as etapas da ER, e a falta estratégias que tenham como objetivo derivar os requisitos dos SCs do SoS.

Tabela 5 – Trabalhos e os desafios abordados

Trabalho	Desafios
#2	Monitoramento em diferentes níveis e com granularidade variada, monitoramento de requisitos sobre diferentes sistemas, monitoramento de sistemas com diferentes tecnologias.
#5	Grande quantidade de <i>stakeholders</i> , requisitos conflitantes, incompletos e em constante modificação.
#9	Escala e complexidade dos sistemas, modificações nas interfaces dos SCs, o que diminui a integridade do SoS como um todo.
#16	Sistemas grandes, complexos, descentralizados e com grande número de <i>stakeholders</i> .
#18	Natureza dos requisitos: fragmentados, conflitantes e instáveis.
#20	O autor trabalha como as próprias características do SoS, independência, distribuição, evolução e comportamento emergente, e como elas são desafios para a ER.
#23	Escala dos sistemas, domínios múltiplos, contexto operacional variado, controle descentralizado, rápida evolução do ambiente, colaboração entre os sistemas e, por fim, sistemas em diferentes fases do ciclo de vida.
#28	Diversos <i>stakeholders</i> com perspectivas diferentes sobre os sistemas, domínios dos sistemas ambíguos e incertos, fronteiras não podem ser definidas claramente.

3.2 Trabalhos Relacionados

Nesta seção, apresenta-se duas abordagens de ER aplicadas ao desenvolvimento de SoS. Dentre as abordagens encontradas na RSL, essas obtiveram destaque, pois apresentam e detalham sua estrutura, fases, atividades e objetivos. Assim, elas serviram de inspiração para as abordagens propostas nos capítulos 4 e 5.

3.2.1 Abordagem SoS-ACRE

A abordagem definida por Holt *et al.* (2015) denominada de SoS-ACRE baseia-se na metodologia *Context-Based Requirements Engineering* (ACRE). O autor afirma que ela foi escolhida por três razões: a ACRE define uma ontologia de requisitos que é usada como base para uma série de visões dos requisitos do sistema; ela não é voltada para nenhum contexto específico e pode ser usada em conjunto com diversas abordagens; além disso, ela pode ser utilizada e representada por qualquer notação ou combinação de notações.

A abordagem em questão define três elementos principais, que são: ontologia, responsável pela definição de todos os conceitos chave e terminologias; *framework*, a partir do qual todas as visões dos sistemas são definidas; e o processo, responsável por definir o processo de como as visões do sistema serão construídas. A ontologia e o *framework* são elementos presentes na ACRE, porém sofreram algumas adaptações para se adequar ao

contexto do SoS, já o processo é um novo elemento inserido pela SoS-ACRE.

3.2.1.1 Ontologia

A Ontologia é a responsável pela descrição de todos os conceitos chave, a terminologia usada para a descrição desses conceitos e os inter-relacionamentos entre eles. A ontologia definida não se trata apenas de um dicionário de dados, segundo os autores, ela é uma poderosa ferramenta para descrição e formalização do *framework*, que é responsável pela definição de diversas visões do sistema. Uma modificação feita pela SoS-ACRE em relação a ACRE é com relação ao contexto do sistema, que no caso de SoS existem dois: sistemas constituintes e SoS.

3.2.1.2 Framework

O *framework*, assim como a ontologia, também é um elemento presente na ACRE. Com o uso dele, descreve-se sete visões do sistema. É importante frisar que essas visões são baseadas nos conceitos definidos na ontologia, são elas:

- Visão de elementos fonte: contém informações relevantes para a especificação dos requisitos, é usada como um mecanismo para a rastreabilidade entre os requisitos e outros aspectos do sistema;
- Visão de descrição dos requisitos: contém as descrições de cada necessidade, objetivos ou capacidades do sistema. O objetivo dessa visão é descrever cada requisito individual com base em um conjunto de atributos;
- Visão de definição de regras: contém as regras aplicadas às definições dos requisitos. Podem ser regras mais complexas baseadas em equações, ou mais gerais;
- Visão de requisitos de contexto: visão responsável por contextualizar os requisitos, dando a eles significado baseado em algum ponto de vista;
- Visão de definição de contexto: identifica e define os diversos pontos de vista, ou contextos usados na visão de requisitos de contexto;
- Visão de validação: essa visão tem como propósito apresentar métodos que indiquem que os requisitos podem ser realizados de alguma maneira. Pode-se fazer isso de maneira informal, por meio de cenários, ou por meio de demonstrações matemáticas;
- Visão de rastreabilidade: visão responsável por definir a rastreabilidade do sistema, porém, aqui, de forma mais específica, classificando os relacionamentos de rastreabilidade de forma implícita - inerentes à linguagem de modelagem, ou explícita - não inerentes a essa linguagem.

Segundo os autores da abordagem, foi identificado que as visões definidas pela ACRE podem ser aplicadas tanto no nível do sistema, como no nível do SoS, porém, nessa aplicação ainda ficavam algumas lacunas, principalmente no tocante à iteração entre os níveis do sistema. Assim, foram definidas mais duas visões, são elas:

- Visão do contexto da interação: visão responsável por fazer a ligação entre o nível do SoS e o nível dos sistemas constituintes. Por exemplo, é necessário mapear as necessidades do SoS como um todo e quais requisitos dos sistemas constituintes serão os responsáveis por realizá-las;
- Visão de validação da interação: visão que tem por objetivo garantir a aplicabilidade do SoS por meio do gerenciamento das diversas visões definidas para os sistemas constituintes.

3.2.1.3 Processo

O processo descreve a abordagem definida para gerar as diversas visões do sistema com base na ontologia. O processo como um todo foi dividido em duas partes, são elas: processo de engenharia de requisitos para SoS e processo de gerenciamento de requisitos para SoS. Uma visão geral desse processo é apresentada na Figura 11. A seguir são apresentadas breves descrições de cada etapa.

Processo de engenharia de requisitos para SoS:

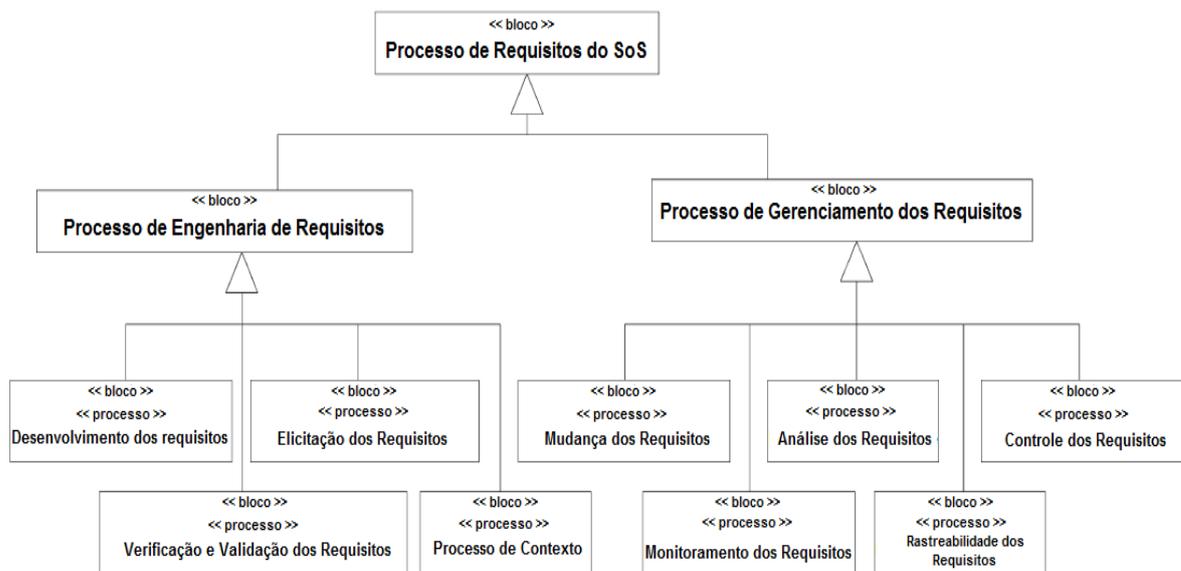
- Processo de desenvolvimento dos requisitos para SoS: gerencia o processo de engenharia de requisitos para SoS como um todo;
- Processo de elicitação dos requisitos: responsável pelo levantamento inicial dos requisitos do sistema e do SoS;
- Processo de contexto: define o contexto de cada sistema constituinte e também do SoS como um todo;
- Processo de verificação e validação: define os critérios de verificação e validação para o SoS.

Processo de gerenciamento de requisitos para SoS:

- Processo de mudança de requisitos: controla as mudanças dos requisitos dos sistemas constituintes ou do SoS;
- Processo de análise dos sistemas constituintes: permite compreender como será a gestão desses sistemas;

- Processo de controle dos requisitos: garante que todas as solicitações de mudança são passíveis de serem realizadas;
- Processo de monitoramento dos requisitos: responsável por identificar todas as mudanças nos sistemas constituintes e no SoS;
- Processo de rastreabilidade: permite a configuração das visões de rastreabilidade.

Figura 11 – Processo da abordagem SoS-ACRE



Fonte: Adaptado de Holt *et al.* (2015)

Sobre a abordagem proposta por Holt *et al.* (2015), pode-se elencar os seguintes pontos positivos: tem como base uma abordagem bem definida, modificando apenas o necessário para se adequar ao desenvolvimento de SoS; bem estruturada, apresentando boa coesão entre seus elementos e fases, além disso, ela é bem completa, no sentido de abranger todas as atividades da ER. Como ponto negativo, pode-se citar a falta de diretrizes e regras para as fases do processo, a abordagem não deixa claro como elicitar, desenvolver e manter os requisitos, assim como não define técnicas e abordagens para a execução dessas atividades.

Com relação a abordagem proposta por esse trabalho, apresentada no capítulo 5. A forma como ela é estruturada é semelhante a SoS-ACRE; apresentando um processo, composto por fases e atividades e um *framework*. No tocante as diferenças, pode-se citar que: a SoS-ACRE aborda todas as etapas da ER, porém, ela não descreve como executar cada uma das atividades do seu processo, ao contrário da abordagem proposta por este trabalho. Outra diferença é que a abordagem proposta neste trabalho é orientada a modelos, proporcionando, assim, algumas vantagens, como: definição de um processo mais ágil e

objetivo para elicitación e modelagem dos requisitos; aumento da reusabilidade e qualidade do sistema, visto que as linguagens formalmente definidas permitem que os modelos sejam definidos uniformemente durante o processo de ER.

3.2.2 Abordagem Baseada em SysML para Especificação de Sistemas Complexos

A abordagem definida por Petrinca, Gammaldi e Tirone (2012) é um processo iterativo e *top-down* que tenta seguir o fluxo típico de um processo de definição dos requisitos, começando pela definição das necessidades dos usuários, passando pela definição dos requisitos por parte dos *stakeholders*, até chegar nos requisitos dos sistemas constituintes.

Na Figura 12, são apresentadas as etapas da abordagem necessárias para produzir um conjunto de requisitos dos sistemas constituintes do SoS a partir das necessidades dos *stakeholders*. Os requisitos, contudo, são apenas a saída final do processo, que envolve a construção de um conjunto de modelos bem estruturados desenvolvidos com a linguagem SysML.

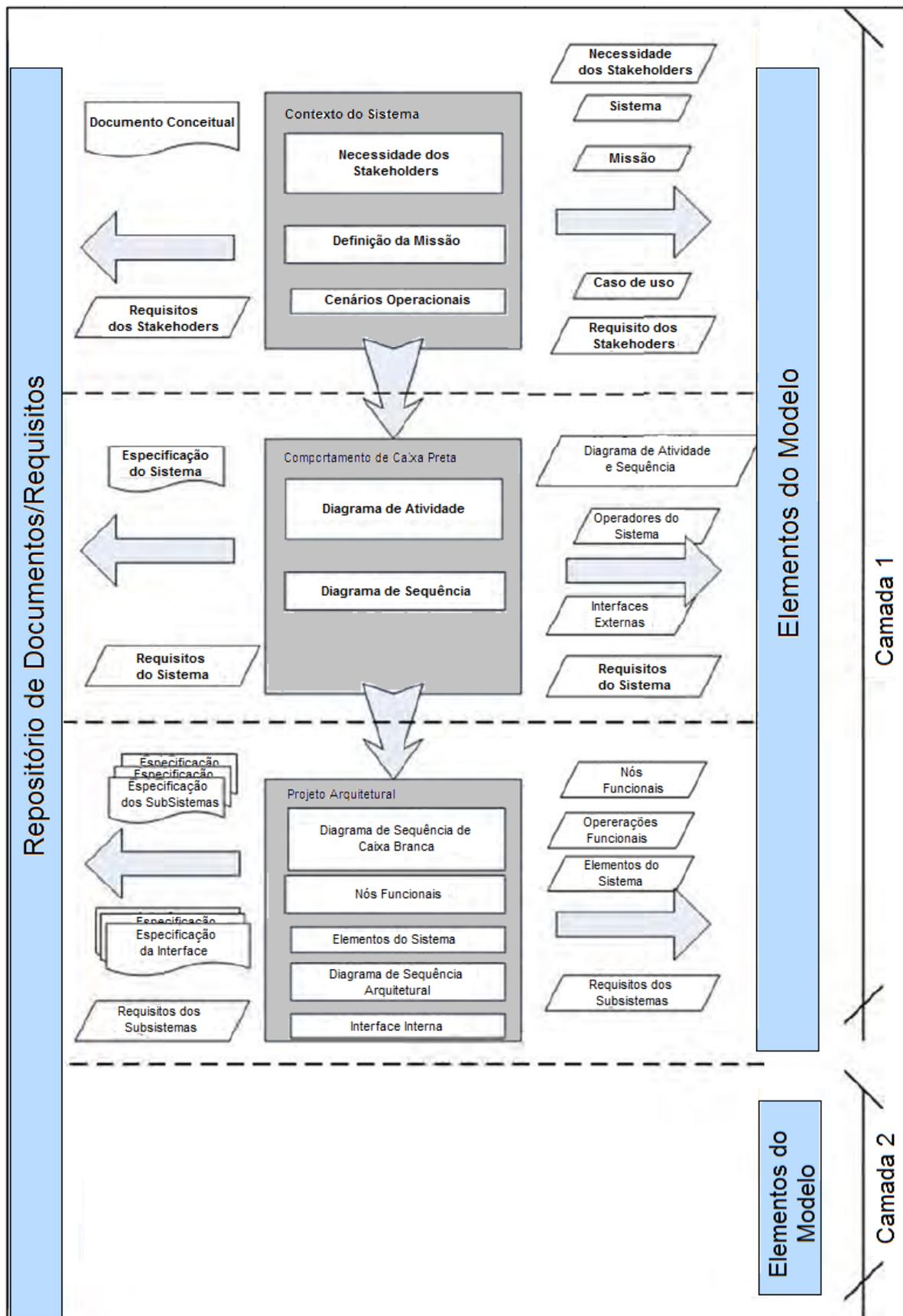
Os modelos SysML são construídos por meio de uma abordagem *top-down*, que começa com a análise do contexto do sistema (fronteiras, atores externos, interfaces), passa pela análise do comportamento independente dos componentes do SoS (comportamento de caixa preta) e finalmente chegando a definição de uma arquitetura. Esse processo é conduzido por três fases: contexto do sistema, comportamento de caixa preta e projeto arquitetural. Nas próximas seções esses três elementos são definidos com mais detalhes.

3.2.2.1 Contexto do Sistema

A primeira fase da abordagem descreve como definir os casos de uso do sistema com base na análise de cenários operacionais, a interação começa com a definição dos *stakeholders* do sistema, então prossegue com a obtenção de suas necessidades, que são postas em um modelo e por fim, a definição dos requisitos dos *stakeholders*.

Para a obtenção dos requisitos dos *stakeholders* (objetivo dessa fase), o comportamento do sistema é definido como um conjunto de missões. Cada missão é definida por uma descrição textual das interações do sistema com seu ambiente. Essas missões são modeladas por meio do diagrama de bloco da SysML.

Figura 12 – Etapas da abordagem baseada em SySML



Ainda nessa fase, a etapa seguinte é a de identificação dos casos de uso do sistema a partir da análise das missões definidas na fase anterior. Essa análise é produzida por meio da criação de um diagrama de atividades, usado para descrever as principais capacidades do sistema em relação a sua missão, seguido pela derivação em alto nível de um diagrama de sequência, usado para descrever as principais interações entre o sistema e seu ambiente. O processo de identificações das missões e criação dos diagramas de atividade e sequência é o responsável por produzir os requisitos dos *stakeholders*.

3.2.2.2 Comportamento de Caixa Preta

O principal objetivo dessa fase é definir o conjunto de requisitos que descrevem o comportamento do sistema. Esse processo começa com uma análise detalhada dos casos de uso identificados na fase anterior. É importante ressaltar que na fase anterior os casos de uso são apenas identificados e uma breve descrição é feita. Nesta fase eles são melhorados e um diagrama de casos de uso formal é desenvolvido, que mostra a lista completa dos atores do sistema e uma descrição formal dos casos de uso.

Após a definição do diagrama de casos de uso, um conjunto de iterações é feito no diagrama de sequência, são eles:

- Observar as linhas de vida do diagrama;
- Observar a descrição dos casos de uso presentes nas mensagens entre os atores e o sistema. Ao contrário das mensagens padrões desse diagrama, essas mensagens representam importantes informações, pois trazem descrições formais do comportamento do sistema;
- Desenhar auto-chamadas antes de cada mensagem de saída da linha de vida do sistema, essas mensagens representarão operações do sistema.

Após esse procedimento, no diagrama de sequência, um conjunto de operações é identificado, cada uma dessas operações se torna uma forte candidata a um requisito funcional, pois descreve o que o sistema deve fazer a partir de um conjunto de entrada com o objetivo de produzir uma saída específica.

3.2.2.3 Projeto de Arquitetura

O objetivo dessa fase é decompor o sistema em um conjunto de sistemas independentes, descrever suas interações e derivar os requisitos funcionais associados a cada um deles, esses sistemas podem ser vistos como os sistemas constituintes do SoS.

A abordagem utilizada é recursiva, o que implica que as mesmas atividades serão usadas para derivar diversos níveis de sistemas. Primeiramente o sistema é decomposto em

uma série de nós funcionais que são derivados da análise dos diagramas de caso de uso e sequência. Cada nó funcional será representado por um diagrama de bloco da SySML. Nesse processo, nenhum requisito foi derivado, eles só serão definidos quando os elementos físicos do sistema forem identificados, para isso, é necessário definir uma solução arquitetural que realize o SoS como um todo. Quando essa arquitetura for definida, um mapeamento entre os sistemas constituintes e os nós funcionais será estabelecido. Dessa relação sairão os requisitos dos sistemas constituintes.

Sobre a abordagem definida por Petrinca, Gammaldi e Tirone (2012) pode-se elencar os seguintes pontos positivos: definição de um processo que, com base nos objetivos gerais do sistema, deriva os requisitos dos SCs; é orientada a modelos, o que apresenta benefícios, como os citados no capítulo 2. Como pontos negativos, destaca-se a complexidade excessiva, principalmente em relação a quantidade de fases do processo; além disso, é uma abordagem de difícil compreensão.

Com relação a abordagem proposta por esse trabalho, apresentada no capítulo 5. A sua principal semelhança com a abordagem descrita anteriormente consiste no fato de que ambas são orientadas a modelos e executam um processo *top-down*. Sobre as diferenças, pode-se citar: a abordagem descrita nesta pesquisa preocupa-se com o ambiente ao qual o SoS está inserido. Além disso, como pode ser observado nos capítulos seguintes, alguns modelos utilizados pelas abordagens são diferentes, assim como suas estruturas.

3.3 Considerações Finais

Este capítulo abordou a revisão sistemática executada durante esta pesquisa, apresentando o seu protocolo, os resultados encontrados e as lacunas de pesquisa. Também foram apresentados trabalhos relacionados que definem abordagens de ER para SoS. É importante ressaltar que esses trabalhos serviram de inspiração para as abordagens propostas nos capítulos seguintes.

4 REAp-SoS: Uma Abordagem de Engenharia de Requisitos para Sistema de Sistemas

Neste capítulo, são apresentadas a versão inicial da abordagem proposta nesta pesquisa, o estudo de caso realizado para a sua avaliação e uma breve análise das limitações encontradas nessa versão da abordagem.

Este capítulo se baseia no seguinte trabalho: DUARTE, F. L.; CASTRO, A. F.; QUEIROZ, P. G. G. *REAp-sos: A Requirement Engineering Approach for System of Systems*. In: *The 6th International Conference on Computational Science and Engineering (CSE 2018)*, 2018. p. 131–148. ISBN 9783642290435. ISSN 2231-5403.

Este capítulo está organizado da seguinte forma: na seção 4.1, são apresentados os conceitos, elementos, fases e diretrizes da abordagem REAp-SoS; na seção 4.2, são apresentados os artefatos gerados pelo estudo de caso realizado e, por fim, na seção 4.3, são apresentadas as principais limitações dessa versão da abordagem.

4.1 REAp-SoS: *Requirement Engineering Approach for SoS*

Segundo Duarte, Castro e Queiroz (2018), a REAp-SoS (*Requirements Engineering Approach for SoS*) é uma abordagem *top-down*, no qual os requisitos dos SCs são derivados a partir das missões definidas para o SoS. Esse processo é conduzido por meio de uma série de iterações sobre modelos SysML. Por missões, entendem-se como objetivos, recursos ou tarefas gerais do sistema. Elas podem se relacionar e contribuir para a realização uma das outras. Em SoS, existem dois tipos de missões, que são: Missões Individuais (MI), atribuídas aos sistemas constituintes; e as Missões Globais (MG), que são objetivos atribuídos ao SoS (SILVA; BATISTA; OQUENDO, 2015) e (SILVA *et al.*, 2014).

Como pode ser observado nas seções posteriores, a abordagem REAp-SoS utiliza a mesma estrutura da abordagem SoS-ACRE, formada por um processo contendo um conjunto de atividades que são responsáveis por gerar os modelos de requisitos. A REAp-SoS, assim como a abordagem baseada em SysML, apresentada no capítulo 3, também utiliza um processo *top-down* para derivação dos requisitos dos sistemas constituintes. Além disso, a REAp-SoS também utiliza a SysML e o diagrama de definição de bloco para representar as missões do SoS.

Os objetivos da abordagem REAp-SoS são:

- Análise e definição do domínio ao qual o SoS está inserido;

- Identificação dos SCs do SoS;
- Identificação das missões do SoS;
- Definição dos requisitos dos SCs por meio das missões do SoS;
- Modelagem das missões do SoS e dos requisitos dos SCs;
- Garantia de rastreabilidade dos requisitos.

A REAp-SoS, além dos objetivos descritos anteriormente, aborda outros aspectos, como por exemplo: confirmação da existência do SoS; definição do domínio ao qual o SoS está inserido; definição de um estudo de viabilidade; modelagem da estrutura, missões e requisitos do SoS; dentre outros aspectos.

A abordagem é composta por três elementos:

- Fase de definição de contexto: definição de todo o contexto do domínio ao qual o SoS está inserido;
- Fase de concepção e modelagem: desenvolvimento dos elementos do *framework*;
- *Framework*: definição dos modelos e artefatos a serem criados pela abordagem.

Na Figura 13, a abordagem REAp-SoS é apresentada por meio da notação *Business Process Model and Notation* (BPMN). Pode-se observar a interação entre a fase de definição do contexto e a fase de concepção e modelagem. Essa última é composta por cinco atividades, que são: identificação e modelagem da estrutura do SoS; identificação e modelagem das missões do SoS; identificação, modelagem e especificação dos requisitos dos SCs; modelagem das atividades do SoS e, por fim, modelagem dos estados dos SCs. Como pode ser observado, todas as as fases e atividades são executadas de forma iterativa e incremental.

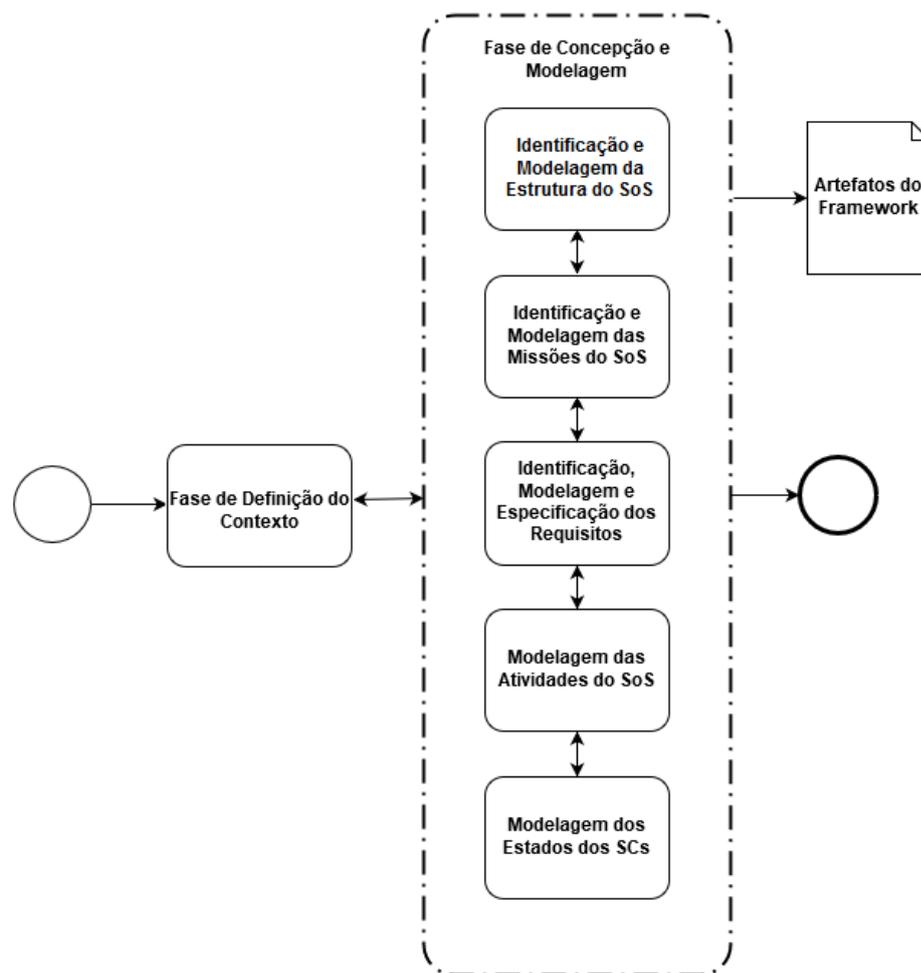
4.1.1 Fase de Definição do Contexto

Na fase de definição do contexto da abordagem REAp-SoS, os analistas são responsáveis por identificar alguns aspectos do domínio ao qual o SoS está inserido. O primeiro aspecto a ser identificado é se o sistema realmente se trata de um SoS, para isso, a abordagem recomenda a identificação de pelo menos dois SCs do SoS e a verificação se esses sistemas apresentam as seguintes características: independência operacional, gerencial, comportamento emergente e distribuição geográfica. Em adicional, recomenda-se também verificar se esses sistemas contribuem para a realização de um único objetivo. A abordagem não recomenda a identificação de todos os SCs do SoS, mas aconselha a identificação do máximo de sistemas possíveis, pois isso facilitará nas etapas posteriores da abordagem.

Depois de confirmado a real existência do SoS, a abordagem recomenda a identificação de aspectos do domínio ao qual o SoS está inserido. Essa atividade é importante pois cada SC é influenciado por aspectos diferentes do seu ambiente, o que pode influenciar seus objetivos e requisitos. Os aspectos do domínio a serem definidos são:

- Entidades: são todos os entes, externos ou internos à organização, que podem influenciar o SoS ou seus sistemas constituintes. É importante ressaltar que o conceito de entidade difere um pouco do conceito de *stakeholders*, pois elas não necessariamente tem interesse no projeto, assim como também podem ou não serem impactadas pelo mesmo. Entretanto, elas sempre influenciarão de alguma forma os SCs ou o SoS;
- Grau de influência: são os níveis de influência que a entidade apresenta nos sistemas constituintes ou no SoS.

Figura 13 – Abordagem REAp-SoS representada em BPMN



Fonte: Autoria Própria

Depois de identificadas as entidades, a REAp-SoS recomenda o desenvolvimento de uma tabela representando os níveis de influência que as entidades exercem nos SCs e no SoS. Essa atividade é importante, pois auxilia em etapas como o gerenciamento de requisitos, facilitando o processo de priorização de mudanças. A abordagem recomenda a utilização de três níveis de influência, que são: baixo (B), médio (M) e alto (A). O uso de apenas três níveis é uma tentativa da abordagem de não gerar complexidade excessiva. Outra recomendação da abordagem é verificar se a entidade especificada é um *stakeholder* de um determinado SC, caso seja, ela exercerá um alto grau de influência nesse sistema.

Para finalizar a fase de definição do contexto, a abordagem REAp-SoS recomenda o desenvolvimento de um estudo de viabilidade, os principais objetivos dessa atividade são:

- Garantir que o SoS realmente contribui para os objetivos da organização;
- Garantir que o SoS e os SCs podem ser construídos com tecnologia e pessoal disponíveis pela organização, ou se é possível selecionar sistemas existentes;
- Garantir que o SoS pode ser integrado com os sistemas presentes na organização.

4.1.2 Fase de Concepção e Modelagem

A fase de concepção e modelagem consiste em um conjunto de atividades cujo objetivo principal é conceber e modelar as missões do SoS, além dos requisitos dos SCs. É importante notar que elas podem e devem ser feitas de forma iterativa e incremental. Além disso, cada atividade desta fase é responsável por gerar um artefato do *framework*. Na Figura 14, são apresentadas as atividades desta fase.

4.1.2.1 Identificação e Modelagem da Estrutura do SoS

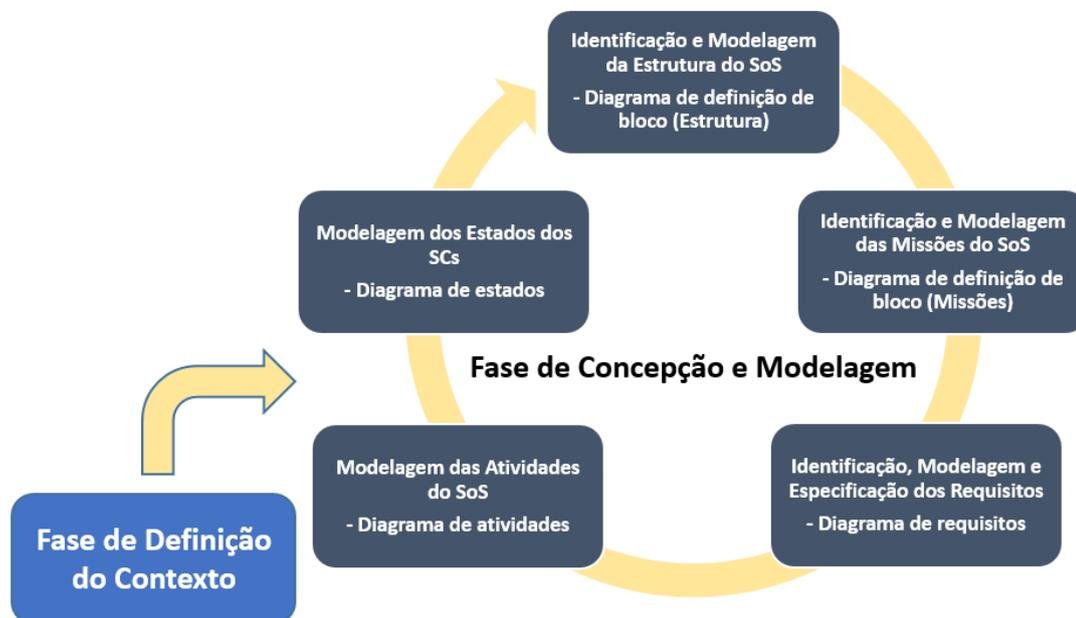
O objetivo desta atividade é identificar e formalizar a estrutura do SoS, identificando os SCs que fazem parte dele. É importante ressaltar que um SoS pode fazer parte de outro, e isso também deve ser modelado.

A modelagem dessa estrutura é feita por meio do diagrama de bloco da SysML, cada bloco representa um SoS ou um SC. A abordagem recomenda que aos blocos que representam um SoS, sejam adicionados o estereótipo « SoS »; e as estruturas que representam SCs, sejam adicionados o estereótipo « SC ». Um exemplo dessa atividade podem ser encontrado na seção 4.2.3.

- Entradas: sistemas constituintes identificados na fase de definição do contexto;
- Objetivo: criar e modelar um diagrama de definição de bloco da SysML que represente a estrutura do SoS a ser desenvolvida;

- Saídas: estrutura do SoS (diagrama de definição de bloco da SysML).

Figura 14 – Atividades da fase de concepção e modelagem da abordagem REAp-SoS



Fonte: Autoria Própria

4.1.2.2 Identificação e Modelagem das Missões do SoS

O objetivo desta atividade é identificar e modelar as missões do SoS e dos SCs. As missões do SoS como um todo são definidas como missões globais, já os objetivos dos SCs são definidos como missões individuais. Observa-se que as missões individuais são as responsáveis por realizar as missões globais do SoS.

Para elicitação das missões, a abordagem REAp-SoS não exige a utilização de técnicas específicas, ela deixa a cargo do profissional a escolha da melhor abordagem para execução desse levantamento. Porém, ela recomenda algumas técnicas que podem ser utilizadas no contexto de SoS, como a GORE e a análise de domínio.

A técnica GORE, que foi discutida na seção 2.3.1, pode ser usada para a identificação das missões do SoS, visto que existe uma similaridade entre os conceitos: objetivos e missões. Nessa técnica, embora os objetivos possam ser explicitamente declarados pelos *stakeholders*, a maioria deles são implícitos, o que dificulta sua identificação. Uma análise preliminar do sistema/organização é uma importante fonte de identificação de objetivos. Essa análise pode resultar em uma lista de problemas e deficiências, nas quais as soluções podem ser definidas como objetivos. Além disso, os objetivos também podem ser obtidos a partir de documentos disponíveis e transcrições de entrevistas. Uma vez que alguns objetivos foram identificados pelos engenheiros de requisitos, o propósito é refiná-los em objetivos

progressivamente mais simples até que eles possam ser facilmente operacionalizados e implementados. (LAPOUCHNIAN, 2005).

Outra técnica que pode ser utilizada, ou mesmo complementar à técnica anterior, é a análise de domínio. De acordo com Lisboa *et al.* (2010), ela é a atividade de identificação de objetos e operações de uma classe de sistemas em um domínio específico. Nessa técnica, analisam-se os objetivos e funcionalidades de uma série de sistemas pertencentes a uma mesma área de aplicação. Nesse processo, alguns artefatos devem ser analisados, como por exemplo: documentos de projeto, manuais de instrução, formulários e arquivos usados nos processos de negócios; além da especificação de outros sistemas similares, o que necessita do auxílio de outras técnicas de elicitação, como a observação (ZOWGHI; COULIN, 2005).

Para modelar as missões do SoS, a abordagem recomenda o uso do diagrama de definição de bloco da SysML. Nesse modelo, cada missão deve ser representada por um bloco, as missões globais devem conter o estereótipo « Missão Global », ao passo que as missões individuais devem apresentar o estereótipo « Missão Individual ». Observa-se que uma estrutura em árvore deve ser formada, sendo as folhas as prováveis missões individuais dos SCs. Além disso, deve-se compreender que as missões representadas pelos nós filhos realizam as missões representadas pelos nós pais. Um exemplo dessa atividade pode ser encontrado na seção 4.2.4.

- Entradas: estrutura do SoS (diagrama de definição de bloco da SysML) e entidades identificadas na fase de definição do contexto;
- Objetivo: identificar e modelar as missões do SoS com base nos SCs identificados. Técnicas recomendadas: GORE e análise de domínio;
- Saídas: missões do SoS e dos SCs (diagrama de definição de bloco da SysML).

4.1.2.3 Identificar, Modelar e Especificar os Requisitos

O objetivo desta atividade é identificar as principais funcionalidades e restrições dos SCs responsáveis por realizar o SoS. Nesta atividade, a partir das operações definidas anteriormente, os requisitos que as realizam devem ser elicitados. Para realizar essa tarefa, assim como na atividade de identificação e modelagem das missões, a abordagem proposta não exige nenhuma técnica específica. Entretanto, recomendam-se técnicas como: entrevistas, questionários, etnografia, observação e *brainstorming*. No tocante a modelagem dos requisitos, a REAp-SoS recomenda o uso do diagrama de requisitos da SysML. Um exemplo dessa atividade pode ser encontrado na seção 4.2.5.

- Entradas: diagramas de definição de bloco (Missões e Estrutura);

- Objetivo: derivar os requisitos dos SCs, técnicas recomendadas: entrevistas, etnografia, *brainstorming*;
- Saídas: requisitos dos SCs (diagrama de requisitos da SysML).

4.1.2.4 Modelagem das Atividades do SoS

O objetivo principal desta atividade é a especificação do comportamento do SoS como um todo e como ocorre, do ponto de vista funcional, a colaboração entre os SCs. Além disso, especifica-se o que será posteriormente projetado na arquitetura, reduzindo o nível de abstração e facilitando a compreensão do SoS pela equipe de desenvolvimento. Um exemplo dessa atividade pode ser encontrado na seção 4.2.6.

- Entradas: estrutura do SoS (diagrama de definição de bloco da SysML) e requisitos dos SCs (diagrama de requisitos da SysML);
- Objetivo: identificar e modelar as atividades responsáveis por descrever o comportamento do SoS;
- Saídas: atividades do SoS e dos SCs (diagrama de atividades da SysML).

4.1.2.5 Modelagem dos Estados dos Sistemas Constituintes

O objetivo dessa atividade é prover um melhor entendimento dos SCs que realizam o SoS, de forma que facilite a implementação desse sistema pela equipe de desenvolvimento. Para isso, a abordagem REAp-SoS recomenda o desenvolvimento de um diagrama de máquina de estados da SysML. Um exemplo dessa atividade pode ser encontrado na seção 4.2.6.

- Entradas: estrutura do SoS (diagrama de definição de bloco da SysML) e requisitos dos SCs (diagrama de requisitos da SysML);
- Objetivo: modelar os estados dos SCs;
- Saídas: estados dos SCs (diagrama de máquina de estados da SysML).

4.1.3 Framework

O *framework* é o elemento da abordagem responsável pela definição formal dos artefatos que devem ser desenvolvidos na fase de concepção e modelagem, são eles:

- Diagrama de definição de bloco (Estrutura): usado para modelagem da estrutura do SoS, por meio da representação dos seus SCs;

- Diagrama de definição de bloco (Missões): usado para modelagem das missões individuais e globais do SoS;
- Diagrama de requisitos: usado para modelagem dos requisitos dos SCs;
- Diagrama de atividades: usado para modelagem das atividades que descrevem o comportamento do SoS como um todo;
- Diagrama de estados: usado para modelagem dos estados dos SCs.

4.2 Estudo de Caso

O sistema escolhido para realizar o estudo de caso com o objetivo de refinar e avaliar a abordagem REAp-SoS é um SoS de controle e monitoramento de tráfego urbano, sua principal funcionalidade é melhorar o fluxo de veículos diminuindo o congestionamento nas vidas da cidade.

O tráfego intenso e o aumento do número de acidentes de trânsito causam um alto custo (econômico, social e ambiental) para as empresas. No Brasil, especialmente nas grandes metrópoles, os gestores públicos têm na mobilidade urbana um dos seus principais desafios. Várias alternativas têm sido feitas para solucionar o problema, dentre elas, pode-se citar: melhoria da infraestrutura, rodízio de veículos e cobrança de tarifas em áreas urbanas. No entanto, elas não tiveram o efeito desejado (BUARQUE, 2008).

O uso de tecnologias nos transportes e infraestrutura de veículos resulta nos chamados *Intelligent Transport Systems* (ITS). Esses sistemas, por meio da tecnologia, fornecem soluções para problemas de tráfego urbano. Dentre os seus objetivos, destacam-se: controle de tráfego e semáforos, gerenciamento de serviços de emergência, cobrança automática de tarifas em transporte coletivo, dentre outros (COLOMBO; VECCHIO, 2012).

Os ITS estão diretamente relacionados ao tema das *Smart Cities*, que podem ser vistas como um conjunto de ações que um cidadão executa, juntamente com serviços de tecnologias, para tornar a cidade mais confortável. Dentre os aspectos essenciais para a criação de uma *Smart Cities*, é necessária uma grande infraestrutura digital moderna e inteligente capaz de atender às demandas que essas cidades necessitam (MONZON, 2015).

4.2.1 Metodologia Aplicada ao estudo de Caso

Antes de apresentar a especificação e modelagem do sistema, é importante elucidar alguns aspectos deste estudo de caso, dentre eles, o seu principal objetivo, que é refinar e avaliar a abordagem REAp-SoS. Assim, optou-se por seguir uma metodologia formal para garantir um maior grau de rigor ao estudo. De acordo com Runeson *et al.* (2012), um

estudo de caso é viável quando um fenômeno contemporâneo é observado dentro do seu contexto real. Para a realização de um bom estudo de caso, o autor preconiza que sejam formalizados aspectos como: a motivação, os objetivos, o método de coleta de dados, a estratégia de condução, os métodos de validação e replicação dos dados. Na Tabela 6, são apresentados esses aspectos para o presente estudo.

Tabela 6 – Metodologia aplicada ao estudo de caso da abordagem REAp-SoS

Aspectos	Procedimento
Motivação	A motivação desse estudo é auxiliar o processo de refinamento das fases, atividades e artefatos da abordagem REAp-SoS.
Objetivos	O objetivo desse estudo é testar a estrutura da abordagem REAp-SoS, identificando, ao final, possíveis dificuldades e incoerências da abordagem. Isso deve ser feito por meio da especificação e modelagem de um sistema de controle e monitoramento de tráfego urbano.
Objeto de estudo	A especificação e modelagem de um projeto que representa um sistema de controle e monitoramento de tráfego urbano.
Método de coleta dos dados	Os objetivos gerais do sistema, assim como seus requisitos foram obtidos por meio da leitura e observação dos trabalhos publicados pelo projeto SIMTUR.
Estratégia de condução do estudo	O estudo de caso deve se iniciar por meio do desenvolvimento dos artefatos exigidos pela abordagem REAp-SoS na fase de definição do contexto. Após essa etapa, os artefatos exigidos pelas atividades da fase de concepção e modelagem da REAp-SoS devem ser desenvolvidos. Todos os artefatos devem seguir as diretrizes propostas pela abordagem REAp-SoS.
Método de validação	O estudo deve ser avaliado por especialistas da área na forma de <i>feedbacks</i> recebidos pela publicação deste trabalho em conferências.
Replicação dos dados	Os resultados do estudo de caso podem ser replicados por meio da modelagem de outros sistemas para o monitoramento do tráfego urbano, desde que sigam estritamente as fases, atividades e diretrizes propostas pela abordagem REAp-SoS.

4.2.2 Contexto do Sistema de Controle e Monitoramento de Tráfego Urbano

Nesta seção, conforme indicado pela abordagem REAp-SoS (Seção 4.1.1), a confirmação da existência do SoS e a definição das características do seu domínio foram determinadas. Todas as informações descritas a seguir, como os SCs e entidades, assim como as missões definidas nas seções posteriores, foram obtidas da observação e leitura dos trabalhos relacionados ao projeto SIMTUR, que podem ser acessadas no seguinte endereço: <http://projeto.unisinos.br/simtur>. Algumas razões levaram à escolha do projeto SIMTUR como sistema de apoio para a realização deste estudo de caso, dentre elas, é possível citar: ele é um projeto com objetivos semelhantes ao proposto nesse estudo, é um

projeto robusto e completo, além de possuir diversos materiais e trabalhos publicados para leitura e análise.

Foram identificados alguns SCs responsáveis por realizar o SoS de controle e monitoramento de tráfego urbano, são eles:

- *Smart cars* (SC1): além de todas as funcionalidades de um carro convencional, este sistema é o principal responsável pela coleta e transmissão de informações sobre o trânsito, tais como: localização, tempo médio parado, velocidade média, dentre outras;
- *Smart buses* (SC2): muitas pessoas dependem dos sistemas de transporte coletivo para se deslocarem nas grandes cidades, assim, também é importante ter sistemas inteligentes de transporte público. Esse SC tem praticamente as mesmas funcionalidades dos *Smart Cars*, porém, suas informações podem ser utilizadas para outros fins, como: tempo de percurso e tempo médio de chegada em uma determinada parada ou terminal;
- *Waze* (SC3): para viabilizar o desenvolvimento, pode ser feita uma adaptação ao projeto por meio da utilização de aplicativos que apresentem funcionalidades requeridas pelos *Smart Cars*. Para isso, uma alternativa mais barata seria o uso de aplicativos como o *Waze* (<https://www.waze.com>) ou similares (*Google Maps*, *Tom Tom Go*). Assim, essas aplicações seriam responsáveis pelo envio da localização dos veículos ao sistema de controle de tráfego;
- Semáforo inteligente (SC4): com base nas informações coletadas em tempo real, os semáforos inteligentes seriam os principais atuadores no fluxo de controle das vias das cidades, tendo como objetivo principal o controle do tempo de abertura e fechamento do semáforo;
- Controlador de tráfego (SC5): este sistema seria o principal controlador do tráfego urbano, com a finalidade de reduzir o congestionamento. Ele seria o responsável por receber as informações dos veículos ou aplicativos, identificar possíveis congestionamentos e acionar os semáforos inteligentes, mudando sua configuração;
- Aplicativo de transporte público (SC6): este sistema é o responsável por fornecer ao usuário informações sobre o transporte público, como: estações de ônibus, rotas, tempo de chegada do próximo ônibus, dentre outras;

Depois de identificar os sistemas, foi verificado se pelo menos dois deles apresentam as características de um SoS. A seguir, são apresentadas as características de um SoS para os sistemas *Smart Cars* e *Semáforos Inteligentes*.

No caso dos *Smart Cars*, suas características principais que o definem como um SC do SoS são:

- Independência Operacional: todos os carros podem ser vistos como um sistema que opera sem a necessidade de outros sistemas, por exemplo: um carro e seus sensores de coleta de informação funcionam sem que outros veículos precisem estar em funcionamento;
- Independência Gerencial: todas as funcionalidades presentes nos carros, sejam relacionadas à locomoção ou à coleta de informações, são controladas pelo próprio veículo, sem a interferência de outros sistemas;
- Distribuição Geográfica: cada veículo ocupa e opera em um espaço geograficamente distante de outros sistemas;
- Comportamento emergente: cada carro, por meio do aplicativo *Waze*, pode coletar e transmitir informações para as unidades de controle e outros veículos, colaborando assim para o funcionamento do SoS;

No caso dos *Semáforos Inteligentes*, os principais aspectos que o definem como um SC do SoS são:

- Independência Operacional: o semáforo deve operar independentemente da operação de outros semáforos ou veículos;
- Independência Gerencial: cada semáforo pode ser gerenciado independentemente de outros semáforos, ou seja, sua configuração pode ser alterada sem a necessidade de modificação dos outros semáforos;
- Distribuição Geográfica: cada semáforo será geograficamente localizado pelas vias da cidade;
- Comportamento emergente: controla e melhora o fluxo de veículos, diminuindo assim os congestionamentos.

Depois de identificar os SCs e garantir que o sistema realmente se trata de um SoS, o próximo passo é definir as entidades, assim como o grau de influência em cada SC. As entidades identificadas são:

- Motorista: responsável pela condução de veículos nas vias da cidade. Além disso, um dos principais beneficiários pelo sistema;

- Usuário de transporte público: usuários dos sistemas de transporte coletivo como ônibus, metrô etc;
- Pedestre: embora não tenha influência direta sobre os problemas que o SoS se propõem a resolver, faz parte do organograma do trânsito nas grandes cidades;
- Usuário do sistema controlador de tráfego: usuários do sistema de controle de tráfego. Embora este sistema possa ser autônomo, deve haver usuários para gerenciar e controlar alguns aspectos e configurações desse sistema;
- Governo local: provavelmente o financiador do projeto, uma vez que é função e dever dos gestores públicos solucionar os problemas de mobilidade em suas cidades;
- Concessionária de transporte público: no Brasil, a maioria das empresas de transporte são privadas, elas tem parcerias com o governo e podem explorar essa atividade por um período de tempo determinado. Assim, também é uma entidade que exerce influência no projeto;
- Equipe de desenvolvimento: equipe da empresa ou organização responsável por obter os requisitos das outras entidades, especificar e desenvolver os diversos sistemas que compõem o SoS.

No tocante à influência das entidades nos SCs. Na Tabela 7, são sumarizados seus níveis para cada SC identificado. Entidades como motoristas e usuários de transporte público, além de apresentarem altos níveis de influência nos SCs como *Smart Cars* e aplicativos de transporte público, também apresentam alto nível de influência no SoS como um todo, visto que toda a estrutura do sistema tem como principais beneficiários essas entidades.

Tabela 7 – Níveis de influência das entidades

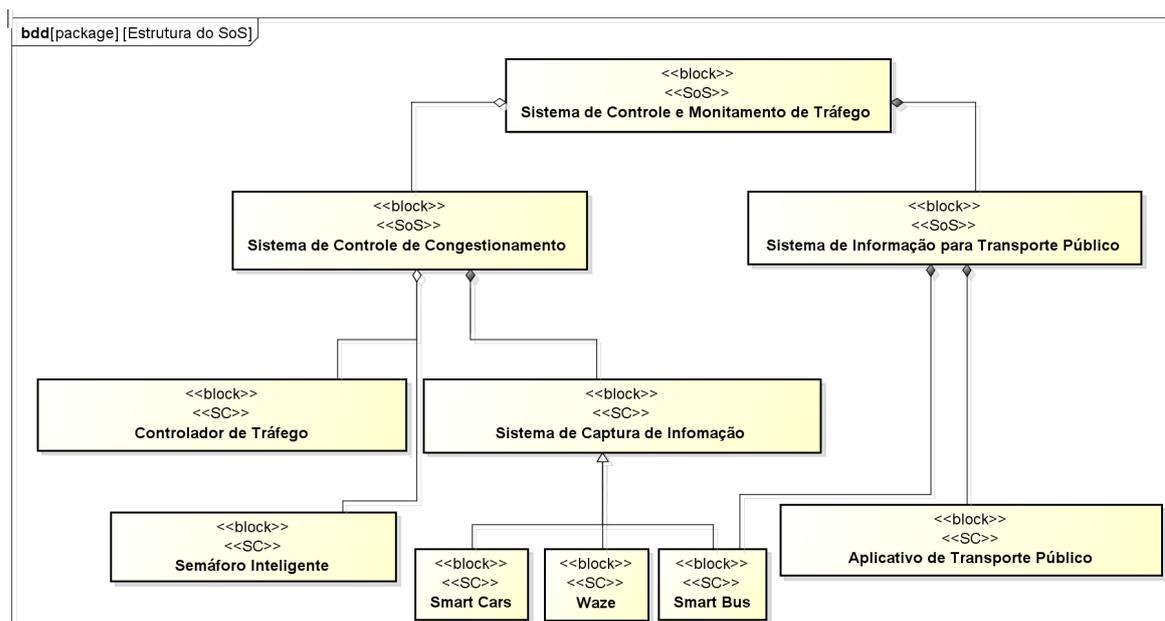
Entidades	S1	S2	S3	S4	S5	SC6	SoS
Motoristas	A	B	M	B	B	B	A
Usuários dos transportes públicos	B	M	M	B	B	A	A
Pedestres	M	M	B	B	B	M	M
Usuários do SCT	B	B	B	B	A	B	M
Governo local	B	A	B	A	M	A	A
Concessionária de transporte público	B	A	B	B	B	A	A
Equipe de Desenvolvimento	M	M	M	M	M	M	M

Para concluir essa fase, a abordagem recomenda a realização de um estudo de viabilidade. Ele tem como objetivo avaliar se o sistema pode realmente ser construído, analisando aspectos como: tecnologia disponível, tempo, pessoal e custo. Como este estudo de caso destinou-se a ajudar no refinamento e avaliação das atividades, fases e outros aspectos da abordagem REAp-SoS, essa atividade não foi executada.

4.2.3 Estrutura do Sistema de Controle e Monitoramento de Tráfego Urbano

Conforme recomendado pela abordagem REAp-SoS (seção 4.1.3.1), foram identificados e modelados os SCs que representam a estrutura do SoS, como pode ser visto na Figura 15. O estereótipo « SoS » foi adicionado para identificar os blocos como um sistema de sistemas e o estereótipo « SC » foi adicionado para identificar o bloco como um sistema constituinte do SoS.

Figura 15 – Estrutura do SoS de controle e monitoramento de tráfego



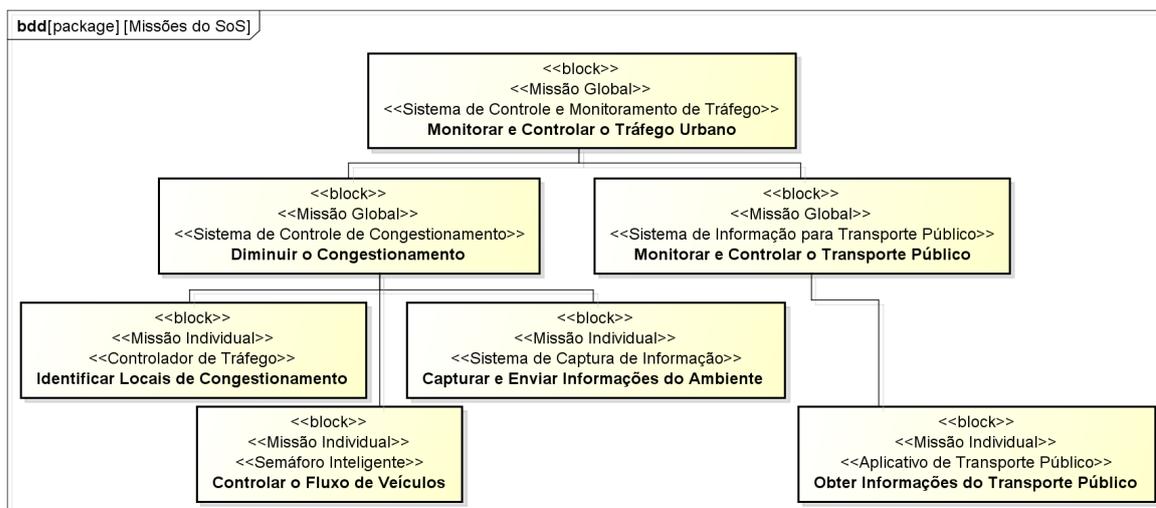
Fonte: Autoria Própria

A estrutura principal do modelo é o *Sistema de Controle e Monitoramento de Tráfego*, que representa o SoS como um todo. Ele divide-se em dois outros SoS, que são: o *Sistema de Controle de Congestionamento* e o *Sistema de Informação de Transporte Público*. Esses SoSs, por sua vez, são compostos por vários SCs, que são: o *Controlador de Tráfego*, *Semáforo Inteligente*, *Aplicativo de Transporte Público* e, por fim, o *Sistema de Captura de Informação*, que pode ainda ser especificado pelos sistemas *Waze*, *Smart Car* ou *Smart Bus*. É importante salientar que o relacionamento composição indica que o SoS ou SC existe mesmo que o SoS ao qual ele pertence deixe de existir.

4.2.4 Missões do Sistema de Controle e Monitoramento de Tráfego Urbano

Conforme recomendado pela abordagem REAp-SoS (seção 4.1.3.2). As missões individuais e globais do SoS e dos SCs foram identificadas e modeladas, como pode ser visto na Figura 16. Cada bloco é representado por uma missão, os estereótipos determinam o tipo da missão. Além disso, foi adicionado um estereótipo ao bloco informando o sistema ao qual a missão pertence.

Figura 16 – Missões do SoS de controle e monitoramento de tráfego urbano



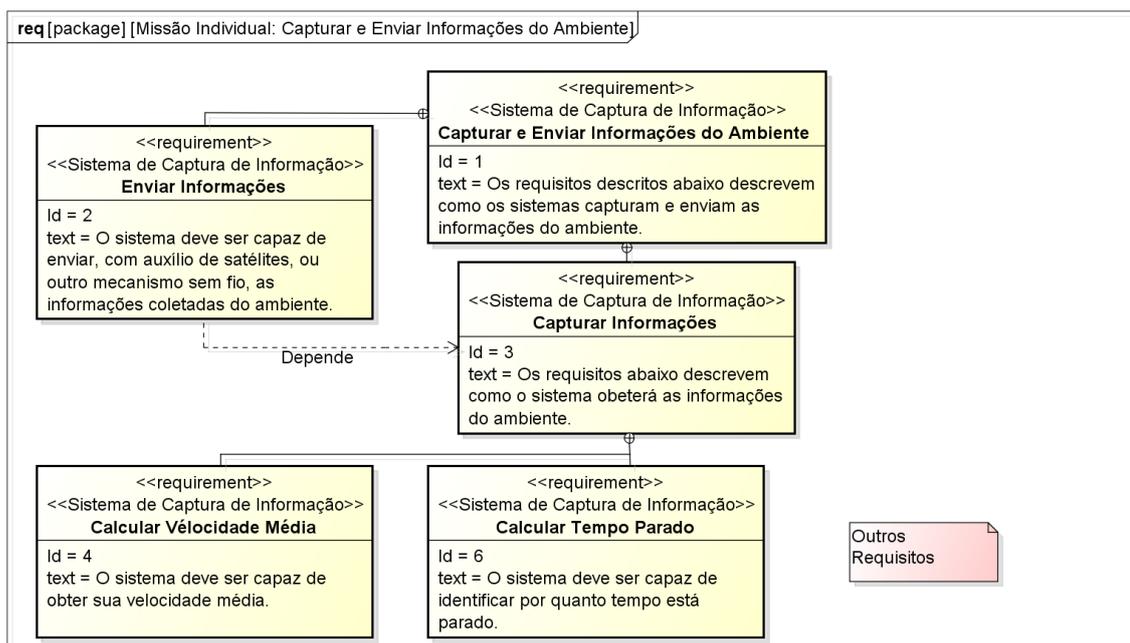
Fonte: Autoria Própria

As missões foram elicítadas por meio da observação e análise das informações contidas no *website* do projeto SINTUR, além dos trabalhos publicados por esse projeto que também se encontram no referido sítio. As missões globais identificadas são: *monitorar e controlar o tráfego urbano*, *diminuir o congestionamento* e *monitorar e controlar o transporte público*. As missões individuais identificadas são: *identificar locais de congestionamento*, *controlar o fluxo de veículos*, *capturar e enviar informações do ambiente* e, finalmente, *obter informações do transporte público*.

4.2.5 Requisitos da Missão Capturar e Enviar Informações

Conforme recomendado pela abordagem REAp-SoS (seção 4.1.3.3), após a modelagem das missões e da estrutura, os diagramas de requisitos devem ser criados. Como exemplo, na Figura 17, é apresentado o diagrama que modela alguns dos requisitos da missão *Capturar e Enviar Informações do Ambiente*. O diagrama completo se encontra na Figura 34, apêndice B. Neste modelo, a missão a ser modelada transforma-se em um requisito geral que, por meio do relacionamento *containment* é dividido nos requisitos *Enviar Informações* e *Capturar Informações*. Esse último também é subdividido nos seguintes requisitos: *Calcular Velocidade Média*, *Calcular Distância*, *Calcular Tempo Parado* e, por fim, *Identificação de Outros Veículos*. É importante destacar a dependência que há entre o requisito *Enviar Informações* e *Capturar Informações*. O primeiro só poderá enviar as informações para o controlador de tráfego, caso essas informações possam ser obtidas pelos outros requisitos.

Figura 17 – Requisitos da missão capturar e enviar informações do ambiente (parcial)



Fonte: Autoria Própria

4.2.6 Atividades do SoS de Controle e Monitoramento de Tráfego

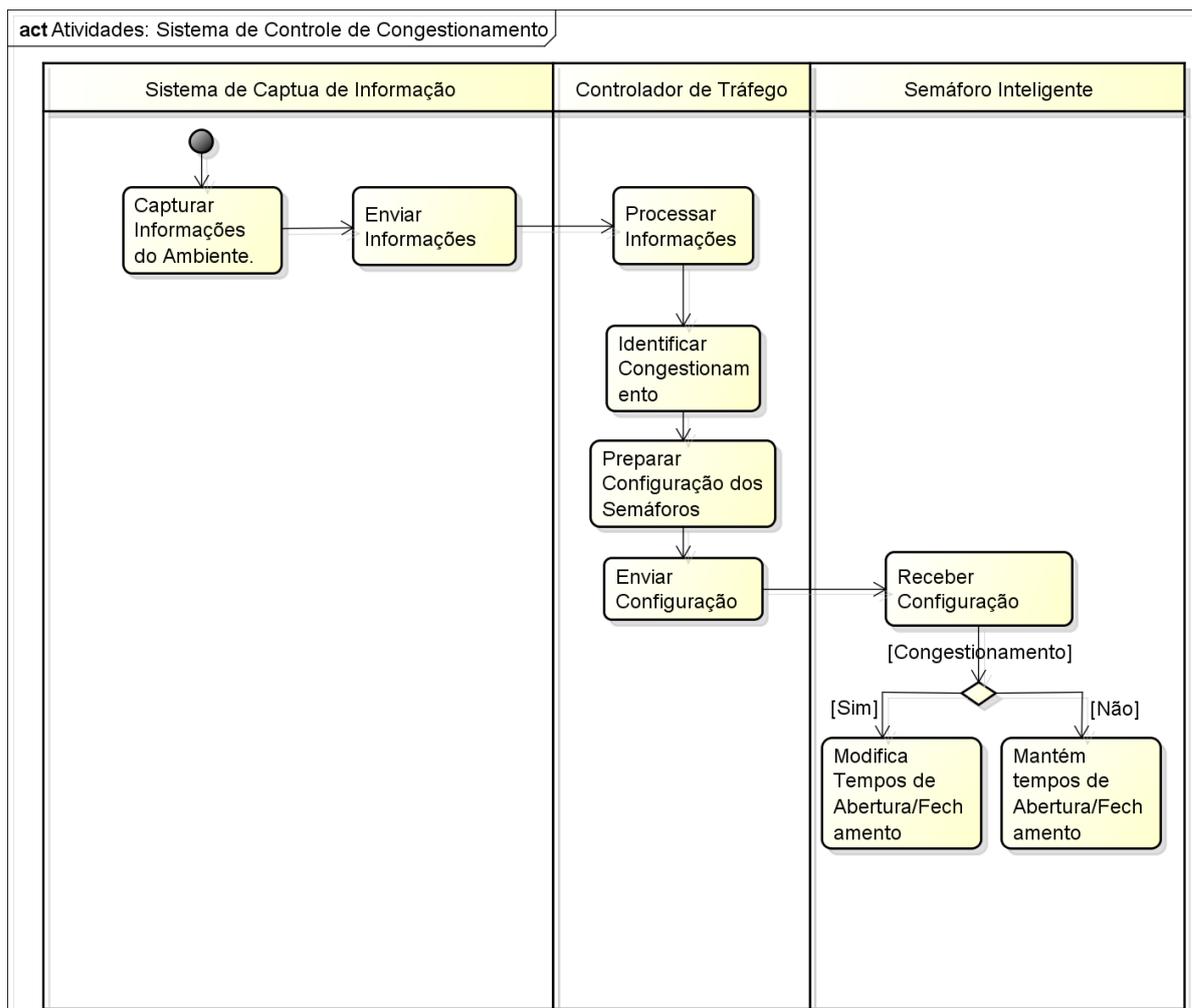
Conforme recomendado pela abordagem REAp-SoS (seção 4.1.3.4), foi criado um diagrama de atividade para o SoS *Sistema de Controle de Congestionamento*, como pode ser visto na Figura 18. A intenção é apresentar o comportamento desse SoS. Nesse modelo, o *Sistema de Captura de Informação*, que pode ser representado pelas suas especificações (*Smart Car*, *Smart Bus* e *Waze*), obtêm as informações necessárias, definidas no modelo de requisitos, enviam essas informações para o *Controlador de Tráfego*, que por sua vez é responsável por processar essa informação, identificar os congestionamentos e, por fim, preparar e enviar a configuração dos semáforos. O SC *Semáforo Inteligente* recebe essa configuração e caso seja necessário, modifica seu parâmetro de tempo para a abertura/fechamento do sinal verde.

4.2.7 Estados do Sistema Semáforo Inteligente

Conforme recomendado pela abordagem REAp-SoS (seção 4.1.3.5) e identificada a efetiva necessidade, modelos de estados podem ser criados para os SCs. Nesse estudo de caso, optou-se por desenvolver um diagrama de estados para o sistema *Semáforo Inteligente*. Como pode ser visto na Figura 19, há dois semáforos modelados, um que representa a via com direção norte-sul e outro que representa a via com direção leste-oeste, um estereótipo foi adicionado aos estados para representar a qual semáforo ele pertence. Três estados foram definidos: verde, amarelo e vermelho. A principal diferença desse semáforo para

um convencional é o uso da variável X na transição do estado verde para o amarelo (em ambos os sinais). Note que: enquanto o tempo de um semáforo se modifica (aumentando ou diminuindo), o inverso acontece no outro. Isso é feito para permitir a modificação da configuração dos semáforos e, assim, proporcionar uma maior permanência no estado verde para vias mais congestionadas.

Figura 18 – Atividades do sistema de controle de congestionamento

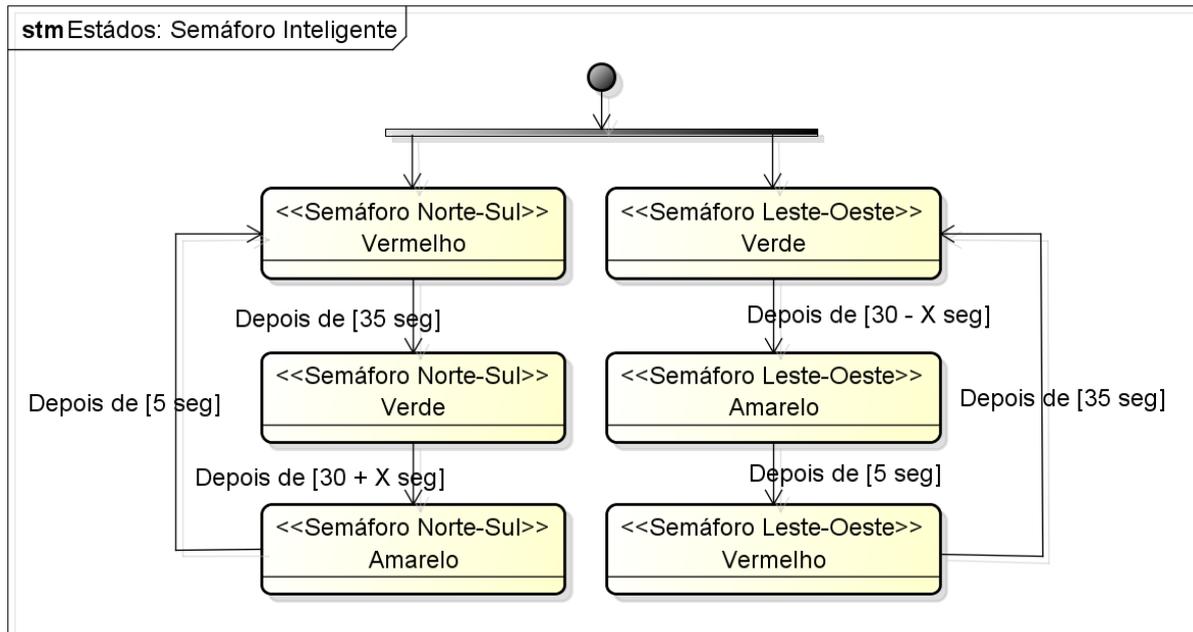


Fonte: Autoria Própria

4.3 Análise da Abordagem REAp-SoS

Após a finalização dos conceitos da abordagem REAp-SoS e da realização do estudo de caso, algumas limitações foram percebidas nesta versão da abordagem, dentre elas, é possível citar:

Figura 19 – Estados do semáforo inteligente



Fonte: Autoria Própria

- A necessidade de, no início da fase de definição do contexto, formalizar o objetivo geral do sistema;
- A necessidade de formalizar as atividades da fase de definição do contexto;
- A necessidade de formalizar a possibilidade de busca e utilização de sistemas constituintes existentes, aumentando o grau de reuso da abordagem.
- Em relação a forma como os graus de influência são representados, percebeu-se que as tabelas não são intuitivas e de fácil utilização. Além disso, sentiu-se a necessidade de acrescentar um nível para medir o impacto gerado por mudanças nos requisitos dos SCs;
- A falta de diretrizes formais para a passagem de um modelo para outro da fase de concepção e modelagem;
- A atividade de identificação e modelagem das missões do SoS deve se apresentar antes da atividade de definição da estrutura do SoS, visto que os SCs têm como objetivo realizar as missões identificadas.

4.4 Considerações Finais

Este capítulo abordou os principais aspectos da versão inicial da abordagem proposta por esse trabalho, a abordagem REAp-SoS. Foram apresentados seus conceitos, elementos, fases e atividades. Também foram apresentados os artefatos gerados pelo estudo de caso realizado para auxiliar no desenvolvimento e avaliação da abordagem. Por fim, foram discutidos aspectos a serem revistos para a nova versão da REAp-SoS, que é apresentada no capítulo 5.

5 MORE-SoS: Uma Abordagem Orientada a Modelos para Engenharia de Requisitos de Sistemas de Sistemas.

Os objetivos deste capítulo são: apresentar a versão final da abordagem proposta, o estudo de caso realizado para a sua validação e uma análise da pesquisa e dos objetivos modelados no estudo de caso.

O capítulo está organizado da seguinte forma: na seção 5.1, são apresentados os conceitos, elementos, fases e diretrizes da abordagem proposta; na seção 5.2, são apresentados os artefatos gerados no estudo de caso realizado e, por fim, na seção 5.3, são apresentados aspectos da condução e dos resultados do *survey* realizado para avaliação dos objetivos do SoS identificados pela abordagem.

5.1 Abordagem Orientada a Modelos para Engenharia de Requisitos de SoS

A MORE-SoS (*Model Oriented Approach to Requirements Engineering for SoS*) é uma nova versão da abordagem REAp-SoS, ela tenta solucionar as limitações da REAp-SoS apresentadas na seção 4.3. Assim como na sua versão inicial, ela também propõem um processo *top-down*, no qual os requisitos dos SCs são derivados a partir das missões definidas para o SoS. Esse processo é conduzido por meio de uma série de iterações sobre modelos SysML.

A MORE-SoS mantém os objetivos presentes na abordagem REAp-SoS, são eles:

- Análise e definição do domínio ao qual o SoS está inserido;
- Identificação das missões do SoS;
- Identificação dos SCs do SoS;
- Transformação das missões do SoS em requisitos dos SCs;
- Modelagem das missões do SoS e dos requisitos dos SCs;
- Garantia de rastreabilidade dos requisitos.

Assim como a REAp-SoS, a MORE-SoS também é composta por três elementos, são eles:

- Fase de definição de contexto: definição de todo o contexto do domínio ao qual o SoS está inserido;
- Fase de concepção e modelagem: desenvolvimento dos elementos do *framework*;
- *Framework*: definição dos modelos e artefatos a serem criados pela abordagem.

Na Figura 20, por meio da notação *Business Process Model Notation* (BPMN), são apresentadas as fases e atividades da abordagem MORE-SoS. O sublinhado indica modificações e atualizações sofridas pelas atividades. Como pode ser observado, a *Fase de Definição do Contexto* agora contém três atividades, que são: *Confirmação da Existência do SoS*, *Identificação do Domínio* e *Estudo de Viabilidade*. Pode-se observar também que, diferente da REAp-SoS, a *Fase de Concepção e Modelagem* agora se inicia por meio da atividade *Identificação e Modelagem das Missões do SoS*.

Um dos principais problemas da REAp-SoS era a falta de critérios formais para o desenvolvimento dos modelos exigidos pelo *framework*. Como pode ser observado nas seções posteriores, a MORE-SoS soluciona esse problema e propõem, em cada uma das atividades da *Fase de Concepção e Modelagem*, diretrizes para a construção dos modelos exigidos.

A abordagem MORE-SoS recomenda que, ao terminar a *Fase de Concepção e Modelagem*, seja executada uma análise sobre os SCs identificados com base nos modelos de requisitos modelados. O seu objetivo é decidir se os SCs serão desenvolvidos pela organização ou adquiridos no mercado. Caso a organização opte pela aquisição dos sistemas, sejam eles *softwares livres* ou proprietários, as fases de projeto e implementação se tornarão mais ágeis, evitando modelagem e desenvolvimento desnecessários. Assim, a identificação e modelagem dos requisitos dos SCs efetuadas na MORE-SoS oferecem liberdade de decisão sobre a implementação ou aquisição dos SCs.

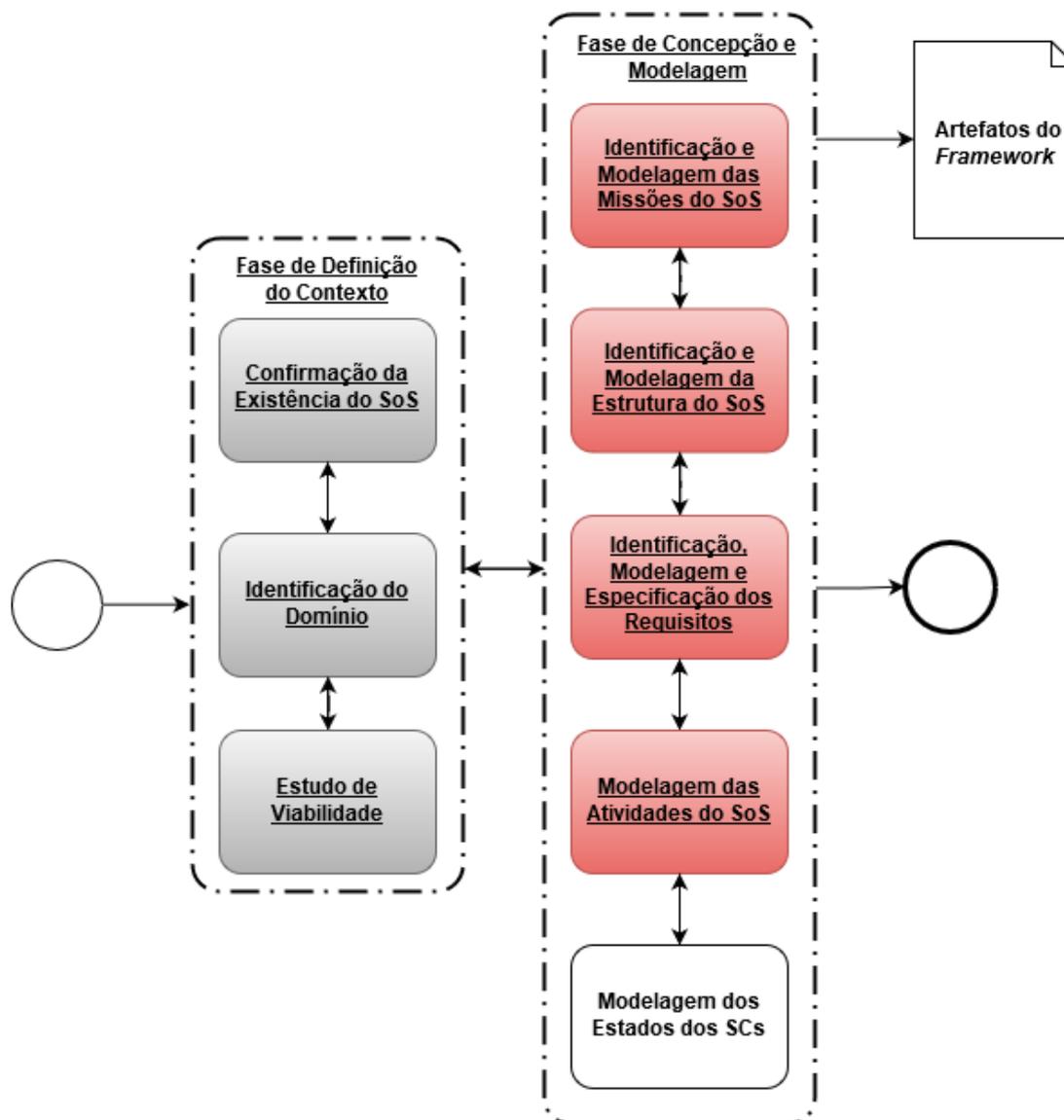
5.1.1 Fase de Definição do Contexto

Na *Fase de Definição do Contexto* da abordagem MORE-SoS, são identificados vários aspectos do domínio do SoS a ser desenvolvido. Ela é dividida em três atividades, que são: *Confirmação da Existência do SoS*, *Identificação do Domínio do SoS* e, por fim, *Estudo de Viabilidade*.

5.1.1.1 Atividade de Confirmação da Existência do SoS

O primeiro aspecto a ser identificado é se o sistema a ser especificado realmente se trata de um SoS. O objetivo é confirmar que a abordagem estará inserida no contexto adequado. Para isso, a MORE-SoS recomenda, primeiramente, a identificação do **objetivo geral do sistema**, ou seja, o principal requisito a ser desenvolvido. Partindo desse objetivo,

Figura 20 – Abordagem MORE-SoS representada em BPMN



Fonte: Autoria Própria

deve-se observar se ele não pode ser realizado por meio de um único sistema, caso não possa, deve-se identificar pelo menos dois sistemas constituintes imprescindíveis para realizar esse objetivo. Além disso, para cada SC, devem-se identificar as seguintes características: independência operacional, gerencial, comportamento emergente e distribuição geográfica. Um exemplo dessa atividade pode ser encontrado na seção 5.2.2.

5.1.1.2 Identificação do Domínio

Depois da confirmação da existência do SoS, a abordagem recomenda identificar alguns aspectos do domínio ao qual o SoS está inserido, são eles: entidade, influência e impacto, esse último não presente na versão inicial da abordagem. Uma descrição sobre cada um desses elementos é apresentada a seguir:

- Entidade: é todo elemento que pode influenciar o SoS, por exemplo: tecnologias, abordagens, pessoas, empresas, partes interessadas, dentre outras. É importante enfatizar que o conceito de entidade difere um pouco do conceito de *stakeholders*, uma vez que as entidades não têm, necessariamente, interesse no projeto, mas podem influenciar positivamente ou negativamente os SCs. Pode-se definir que todo *stakeholder* é uma entidade, mas nem toda entidade é um *stakeholder*;
- Influência: se refere ao nível de poder que uma entidade tem de influenciar os requisitos dos SCs, ou mesmo os objetivos de um determinado SoS;
- Impacto: se refere ao nível de impacto que uma entidade sofre por mudanças nos requisitos dos SCs.

Diferente da REAp-SoS, que utiliza o auxílio de uma tabela para representar as entidades e seus níveis de influência. A MORE-SoS recomenda que se faça um gráfico de influência x impacto para as entidades identificadas, plotando no gráfico os diversos SCs que representam o SoS. Sabe-se que os requisitos de um SoS são instáveis e conflitantes, principalmente pela existência de diversos *stakeholders* em diferentes SCs. Assim, o objetivo desta atividade é ajudar em futuras priorizações de mudanças, além de ajudar a determinar quem será afetado por mudanças nos SCs.

A abordagem recomenda um gráfico bidimensional com o eixo X representando a influência e o eixo Y representando o impacto, recomenda-se também valores mínimos de zero e máximo de dez. Assim, para uma entidade com um SC representado por valores (10,10), significa que essa entidade exerce grande influência nos requisitos desse SC, além disso, mudanças nos seus requisitos afetarão de forma significativa essa entidade.

É importante salientar que dependendo da natureza do SoS, podem ser identificadas poucas entidades e, além disso, elas podem apresentar graus semelhantes de influência e impacto nos diversos SCs. Porém, também podem haver situações em que a quantidade de entidades e SCs são elevados, o que tornaria muito custoso fazer essa análise para cada entidade identificada. Portanto, a MORE-SoS deixa a cargo dos engenheiros de requisitos e analistas a decisão de executar ou não essa atividade, além de decidir quais entidades merecem receber essa análise. Um exemplo dessa atividade pode ser encontrado na seção 5.2.2.

5.1.1.3 Estudo de Viabilidade

Assim como na sua versão inicial, a MORE-SoS recomenda a execução de um estudo de viabilidade que observe os seguintes aspectos:

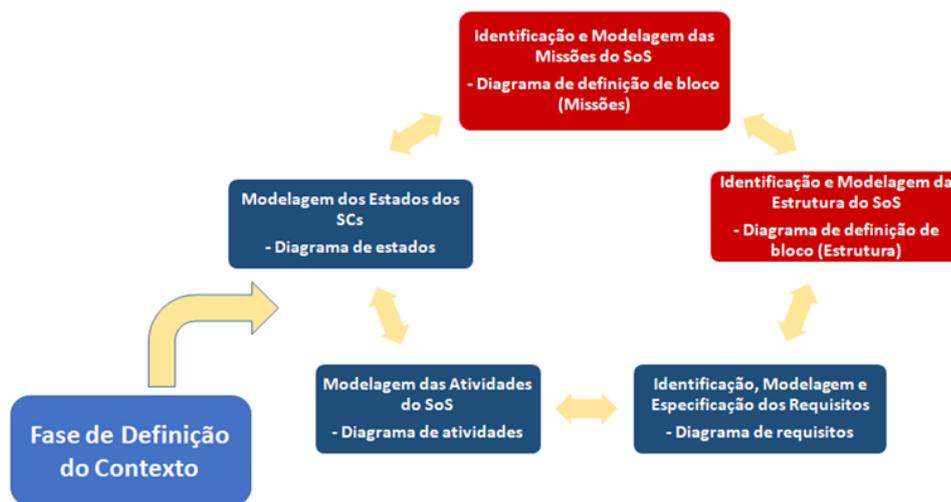
- Se o SoS realmente contribui para os objetivos da organização;

- Se o SoS e os SCs podem ser construídos com tecnologia e pessoal disponíveis pela organização, ou se é possível selecionar sistemas existentes;
- Se o SoS pode ser integrado com os sistemas presentes na organização;

5.1.2 Fase de Concepção e Modelagem

O objetivo desta fase é, assim como na sua versão inicial, conceber e modelar as missões do SoS e os requisitos dos SCs. As atividades desta fase devem ser executadas de forma iterativa e incremental. Além disso, em cada atividade um artefato do *framework* deve ser desenvolvido. Diferente da REAp-SoS, nesta versão, a *Fase de Concepção e Modelagem* se inicia pela execução da atividade de *Identificação e Modelagem das Missões do SoS*. Na Figura 21, são apresentadas as atividades desta fase, além dos respectivos artefatos gerados por elas.

Figura 21 – Atividades da fase de concepção e modelagem



Fonte: Autoria Própria

5.1.2.1 Identificação e Modelagem das Missões do SoS

Assim como na REAp-SoS, o objetivo desta atividade é identificar e modelar as missões do SoS e dos SCs. Para isso, esta versão da abordagem segue as recomendações da REAp-SoS, deixando a cargo do profissional a escolha da melhor abordagem para a elicitacão dessas missões, ela também recomenda a utilização dos princípios da GORE e a técnica de análise de domínio. Porém, como pode ser observado a seguir, a MORE-SoS apresenta um conjunto de diretrizes formais para a execução desta atividade. Um exemplo pode ser encontrado na seção 5.2.3.1.

- Entradas: objetivo geral identificado na fase de definição do contexto;

- Diretrizes:
 - Identificar as missões globais e individuais;
 - Modelar as missões em formato de árvore por meio do diagrama de bloco da SysML;
 - Adicionar os estereótipos « Missão Global » e « Missão Individual » para cada missão identificada.
 - Blocos não folha devem conter o estereótipo « Missão Global », já os blocos folhas devem conter o estereótipo « Missão Individual ».
- Saídas: missões do SoS e dos SCs (diagrama de definição de bloco SysML);

5.1.2.2 Identificação e Modelagem da Estrutura do SoS

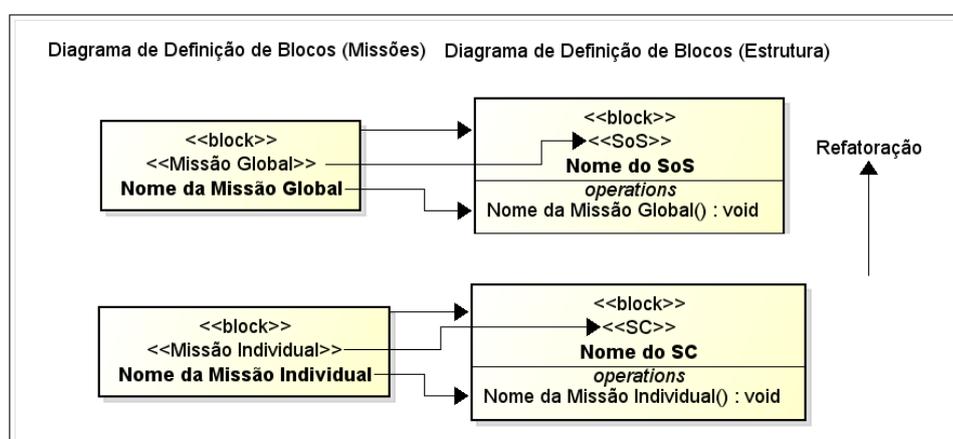
Assim como na REAp-SoS, o objetivo desta atividade é identificar e formalizar a estrutura do SoS, definindo os SCs que fazem parte dele. Como modificação, a MORE-SoS traz um conjunto de diretrizes para desenvolver o modelo estrutural. Um exemplo dessa atividade pode ser encontrado na seção 5.2.3.2.

- Entradas: missões globais e individuais (diagrama de definição de bloco SysML);
- Diretrizes:
 - Cada bloco com estereótipo « Missão Global » transforma-se em um bloco com estereótipo « SoS »;
 - Cada bloco com estereótipo « Missão individual » transforma-se em um bloco com estereótipo « SC »;
 - Definem-se nomes para os novos blocos. Eles devem representar os possíveis sistemas que serão implementados ou adquiridos;
 - Para cada novo bloco, o nome da missão que o originou transforma-se em uma operação da nova estrutura.
 - Refatora-se o diagrama por meio de um processo *bottom-up*. Caso seja identificado que um único sistema pode realizar mais de uma missão, mesclam-se os blocos, transformando as missões em operações da nova entidade. Nesse processo, missões globais se tornarão operações dos blocos modelados com estereótipo « SoS », enquanto que as missões individuais se tornarão operações dos blocos modelados com estereótipo « SC ». Também podem surgir, como parte do processo de refatoração, novos SCs como especializações de um SC mais geral.

- Saídas: estrutura SoS, SoS e sistemas constituintes identificados (diagrama de definição de bloco SysML);

Na Figura 22, o processo de transformação do modelo de missões para o modelo estrutural é apresentado. Como pode ser observado, os blocos com estereótipo « Missão Global » transformam-se em blocos com estereótipo « SoS », já os blocos com estereótipo « Missão individual » transformam-se em blocos com estereótipos « SC ». Por fim, as missões transformam-se em operações dos novos blocos e um processo de refatoração *bottom-up* deve ser executado para corrigir eventuais distorções.

Figura 22 – Transformação do modelo de missões no modelo estrutural



Fonte: Autoria Própria

5.1.2.3 Identificação, Modelagem e Especificação dos Requisitos

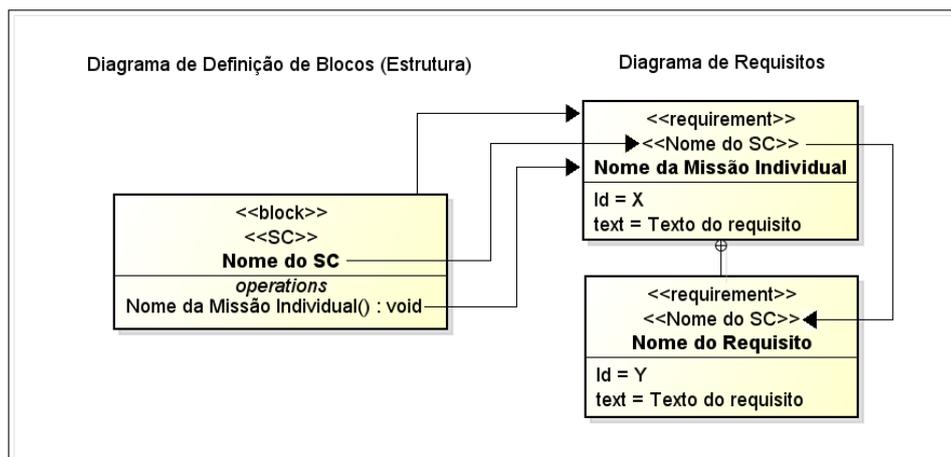
O objetivo desta atividade não foi modificado pela MORE-SoS, que é a identificação das principais funcionalidades e restrições dos SCs responsáveis por realizar o SoS. A MORE-SoS também recomenda, para a elicitación dos requisitos, a utilização de técnicas como: entrevistas, questionários, etnografia, observação e *brainstorming*. Como pode ser observado a seguir, diferente da REAp-SoS, essa atividade só apresenta como entrada a estrutura do SoS, representado pelo diagrama de definição de bloco (Estrutura). Também foram adicionadas diretrizes formais para o desenvolvimento do modelo de requisitos dos SCs. Um exemplo dessa atividade pode ser encontrado na seção 5.2.3.3.

- Entradas: estrutura do SoS (diagramas de definição de bloco da SysML);
- Diretrizes:
 - Cada operação identificada no *Diagrama de definição de bloco (Estrutura)* transforma-se em um requisito do diagrama de requisitos;

- O nome do SC responsável por realizar a missão transforma-se em um estereótipo do requisito modelado;
 - Derivam-se para um menor grau de abstração, os diversos requisitos responsáveis por realizar o requisito modelado anteriormente;
 - Para todos os requisitos filhos, acrescenta-se o estereótipo presente no requisito pai.
- Saídas: requisitos dos SCs (diagrama de requisitos a SysML);

Na Figura 23, o processo de transformação do modelo estrutural no modelo de requisitos é apresentado. Cada bloco com estereótipo « SC » transforma-se em um requisito e o nome da operação desse bloco transforma-se no nome do requisito. Por fim, o nome do SC transforma-se em um estereótipo do requisito modelado. A partir dessa estrutura, identificam-se outros requisitos responsáveis por realizar o requisito proveniente da missão individual identificada.

Figura 23 – Transformação do modelo estrutural no modelo de requisitos



Fonte: Autoria Própria

5.1.2.4 Modelagem das Atividades do SoS

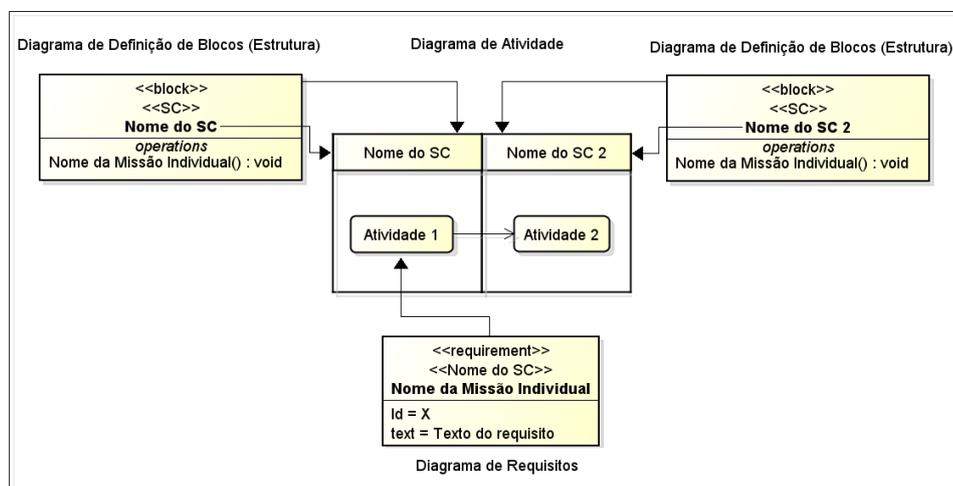
Na MORE-SoS, o objetivo desta atividade também é a especificação e modelagem do comportamento do SoS, elucidando como ocorre, do ponto de vista funcional, a colaboração entre os SCs. Assim como nas outras atividades, diretrizes formais foram adicionadas para a execução desta atividade. Um exemplo pode ser encontrado na seção 5.2.3.4.

- Entradas: estrutura do SoS (diagrama de definição de bloco da SysML) e requisitos dos SCs (diagrama de requisitos da SysML);
- Diretrizes:

- Cada bloco com estereótipo « SC » transforma-se em uma partição do diagrama de atividades;
 - Identificam-se as atividades responsáveis por realizar um determinado requisito;
 - Modelam-se as atividades, decisões, *forks* e *joins* responsáveis por representar dinamicamente os requisitos especificados anteriormente;
 - As atividades identificadas devem permanecer na raia indicada pelo estereótipo do requisito que as originou (esse estereótipo indica o SC ao qual o requisito pertence);
- Saídas: atividades do SoS e dos SCs (diagrama de atividade da SysML);

Na Figura 24, o processo de transformação dos modelos estruturais e de requisitos para o modelo de atividades é apresentado. Cada bloco com estereótipo « SC » transforma-se em uma partição do diagrama de atividades. O nome do SC transforma-se no nome da sua respectiva partição. Do diagrama de requisitos, são identificadas e derivadas as atividades. É importante observar que nem todo requisito se transformará em uma atividade, da mesma forma que um requisito pode resultar em mais de uma atividade.

Figura 24 – Transformação dos modelos estrutural e de requisitos no modelo de atividades



Fonte: Autoria Própria

5.1.2.5 Modelagem dos Estados dos Sistemas Constituintes

Esta atividade praticamente não foi modificada na versão atual da abordagem, seu objetivo é proporcionar aos projetistas e desenvolvedores uma melhor compreensão dos SCs identificados, que é feito por meio da modelagem de um diagrama de máquina de estados para os SCs. A MORE-SoS também dispensa a criação desse diagrama para todos os SCs, deixando a cargo da equipe de desenvolvimento a modelagem dos SCs que ela julgar necessários.

- Entradas: estrutura do SoS (diagrama de definição de bloco SysML) e requisitos dos SCs (diagrama de requisitos da SysML);
- Diretrizes (Respeitando o contexto descrito anteriormente):
 - Cada bloco com estereótipo « SC » transforma-se num diagrama de máquina de estados;
 - Analisar os requisitos do SC a serem modelados;
 - Criar e modelar os estados e transições responsáveis por realizar esses requisitos.
- Saídas: estados dos SCs (diagrama de estado da SysML);

5.1.3 Framework

No *Framework*, são formalizados e apresentados os artefatos que devem ser criados na fase de concepção e modelagem, são eles:

- Diagrama de definição de bloco (Missões): modelagem das missões do SoS (missões globais) e dos SCs (missões individuais);
- Diagrama de definição de bloco (Estrutura): modelagem dos SCs responsáveis por realizar o SoS. Esse diagrama se diferencia do apresentado anteriormente por meio da utilização de estereótipos diferentes para representar missões e SCs;
- Diagrama de requisitos: modelagem dos requisitos dos SCs, que têm como objetivo realizar as missões modeladas no *diagrama de definição de bloco (Missões)*;
- Diagrama de atividades: modelagem do comportamento dinâmico dos SCs, por meio da apresentação do fluxo das atividades;
- Diagrama de estados: modelagem dos possíveis estados de um SC.

5.2 Estudo de Caso

O sistema escolhido para realizar o estudo de caso com o objetivo de validar a abordagem MORE-SoS é um SoS de controle e monitoramento da produção animal, sua principal funcionalidade é identificar anomalias no comportamento bovino, que podem indicar doenças, má qualidade do alimento, dentre outros aspectos.

O Brasil vem se destacando na produção e exportação de carne bovina no mercado internacional. A busca pelo aumento da produtividade, aperfeiçoamento da produção e redução de custos têm permitido sua inserção no mercado internacional. Um dos fatores que contribuíram para esse fenômeno é o aumento da utilização da tecnologia em áreas

como a zootecnia e a agricultura. Assim, foram observados nos últimos anos grandes avanços na zootecnia de precisão, que pode ser definida como a gestão da produção animal baseada na tecnologia e nos princípios da engenharia de processos (TECH, 2012).

Dada a relevância do tema e aliado ao aspecto de que a Universidade Federal Rural do Semi-Árido (UFERSA) se destaca nessa área do saber, optou-se por, com a participação dos programas de graduação, mestrado e doutorado em zootecnia da UFERSA, especificar um sistema de controle e monitoramento da produção animal, chamado SisBovi, que tem como objetivos: diminuir custos de produção, aumentar o lucro, identificar má alimentação e possíveis doenças, aumentar a qualidade de vida dos produtores rurais, melhorar o conforto e o bem estar do animal, dentre outros. Para alcança-los, é necessário observar aspectos como:

- Posicionamento do animal: localização exata do animal;
- Estado fisiológico: aquisição da temperatura corporal para avaliação do bem estar e sanidade nos animais;
- Comportamento animal: aquisição dos tempos pastejo, ruminação e descanso.

5.2.1 Metodologia Aplicada ao estudo de Caso

Assim como o estudo de caso apresentado no capítulo 4, a condução deste estudo também foi orientada pelas diretrizes propostas por Runeson *et al.* (2012), A seguir, nas Tabelas 8 e 9, são apresentados a motivação, os objetivos, o objeto de estudo, os métodos de coleta de dados, a estratégia de condução, o método de validação e a forma de replicação dos dados.

Tabela 8 – Metodologia aplicada ao estudo de caso da abordagem MORE-SoS

Aspectos	Procedimento
Motivação	A motivação desse estudo é a validação da abordagem proposta por esse trabalho, no caso, a abordagem MORE-SoS.
Objetivos	O objetivo desse estudo é validar a abordagem MORE-SoS, que deve ser feito por meio da especificação e modelagem de um sistema de controle e monitoramento da produção animal. Como parte do processo, devem-se definir os aspectos do contexto do SoS, modelar sua estrutura, suas missões, bem como os requisitos e atividades dos SCs, e, se necessário, os estados dos SCs.
Objeto de estudo	A especificação e modelagem de um projeto que representa um sistema de controle e monitoramento da produção animal.
Método de coleta dos dados	Os objetivos gerais e requisitos do sistema devem ser obtidos por meio da observação e análise de sistemas e trabalhos semelhantes. Dentre esses sistemas, podem-se citar os desenvolvidos pelas empresas Tigrinus, Nortronic e ChipInside.

Tabela 9 – Metodologia aplicada ao estudo de caso da abordagem MORE-SoS (conclusão)

Aspectos	Procedimento
Estratégia de condução do estudo	O estudo de caso deve se iniciar por meio da identificação e modelagem dos principais requisitos e objetivos dos sistemas do SoS. Por meio de um <i>survey</i> e de um teste de hipótese, as principais funcionalidades especificadas devem ser avaliadas por pessoas qualificados da área, como discentes dos cursos de graduação, mestrado e doutorado em ciência animal da UFERSA, além de pesquisadores da Empresa Brasileira de Pesquisa Agropecuária (Embrapa).
Método de validação	Os principais objetivos do SoS, identificados pela abordagem MORE-SoS no estudo de caso, devem ser avaliados por meio de um teste de hipótese conduzido com profissionais da área de zootecnia. Como insumo para esse teste, as respostas dos profissionais devem ser obtidas por meio de um <i>survey</i> .
Replicação dos dados	Os resultados do estudo de caso podem ser replicados por meio da modelagem de outros sistemas para o monitoramento da produção animal, desde que sigam estritamente as fases, atividades e diretrizes propostas pela abordagem MORE-SoS. Além de utilizar, como insumo para o novo estudo, os mesmos sistemas deste estudo de caso (Tigrinus, Nortronic e ChipInside).

5.2.2 Contexto do Sistema SisBovi

Nesta seção, para o estudo de caso sobre o sistema de controle e monitoramento da produção animal, chamado de SisBovi, são apresentadas as atividades da fase de definição do contexto recomendadas pela abordagem MORE-SoS. Assim, iniciou-se pela definição do objetivo geral do sistema, que é apresentado a seguir:

O sistema deve ser capaz de controlar e monitorar a produção bovina por meio de sensores acoplados aos animais, de tal forma que possam ser identificados aspectos do seu comportamento, como: ruminção, descanso, pastejo e cio.

Para a realização desse objetivo geral, alguns SCs devem trabalhar e colaborar entre si. A seguir, seguem os SCs identificados para esse SoS:

- Sistema de monitoramento de comportamento: responsável por analisar o comportamento dos animais, como: pastejo, ruminção, descanso e cio.
 - Sensor de pressão: sensor acoplado a garupa do animal para identificar a monta de outros animais;
 - Sensor de atividade: sensor sonoro responsável por identificar o nível de atividade do animal. Por meio dos diferentes sons emitidos, é possível identificar aspectos como pastejo, ruminção e descanso;

- Sensor de temperatura corporal: sensor responsável por monitorar a temperatura corporal do animal, emitindo um alerta caso a temperatura sofra uma aumento significativo;
 - Transmissor VHF (*Very High Frequency*): recebe os sinais dos sensores e transmite via sinal de rádio;
 - Receptor VHF: representado por antenas, responsáveis por captar os sinais VHF emitidos pelos transmissores e repassar a informação para o sistema controlador de comportamento;
 - *Software* controlador de comportamento: sistema responsável por centralizar as informações provenientes dos diversos sensores e identificar anomalias no pastejo, ruminação, descanso e temperaturas. Além disso, trabalhar na identificação de períodos de cio.
- Sistema de identificação de desaparecimento: sistema responsável por determinar se o gado está dentro dos limites da fazenda.
 - Receptor GPS (*Global Positioning System*): dispositivo responsável por receber sinais provenientes dos satélites, com o objetivo de determinar sua exata localização por meio de cálculos de distância e triangulação;
 - Satélite: sistema responsável por enviar sinais aos receptores GPS;
 - Transmissor GSM (*Global System for Mobile Communications*): sistema responsável por enviar as coordenadas (latitude e longitude) ao sistema de detecção;
 - *Software* controlador de comportamento: *software* responsável por receber a localização dos dispositivos transmissores, calcular se o animal se encontra no perímetro determinado, caso ocorra a identificação de animais fora do perímetro, emite um alerta às entidades responsáveis.

Identificados os sistemas, a abordagem recomenda a identificação das características de um SoS para pelo menos dois SCs. A seguir, elas são apresentadas para o sistema *sensor de atividade*:

- Independência Operacional: o sensor presente em um animal, opera e captura suas informações sonoras de forma independente dos outros sensores de atividade, bem como dos demais SCs;
- Independência Gerencial: toda a operação de registro e alerta das informações de um sensor são gerenciadas de forma independente de outros sensores e SCs;
- Distribuição Geográfica: cada sensor estará acoplado a um determinado animal e distribuído espacialmente no campo/fazenda;

- Comportamento emergente: cada sensor registra informações do animal e as envia por meio de sinais de rádio frequência, colaborando para a análise do comportamento animal.

Para o *software controlador de comportamento*, as características que o identificam como um SC de um SoS são:

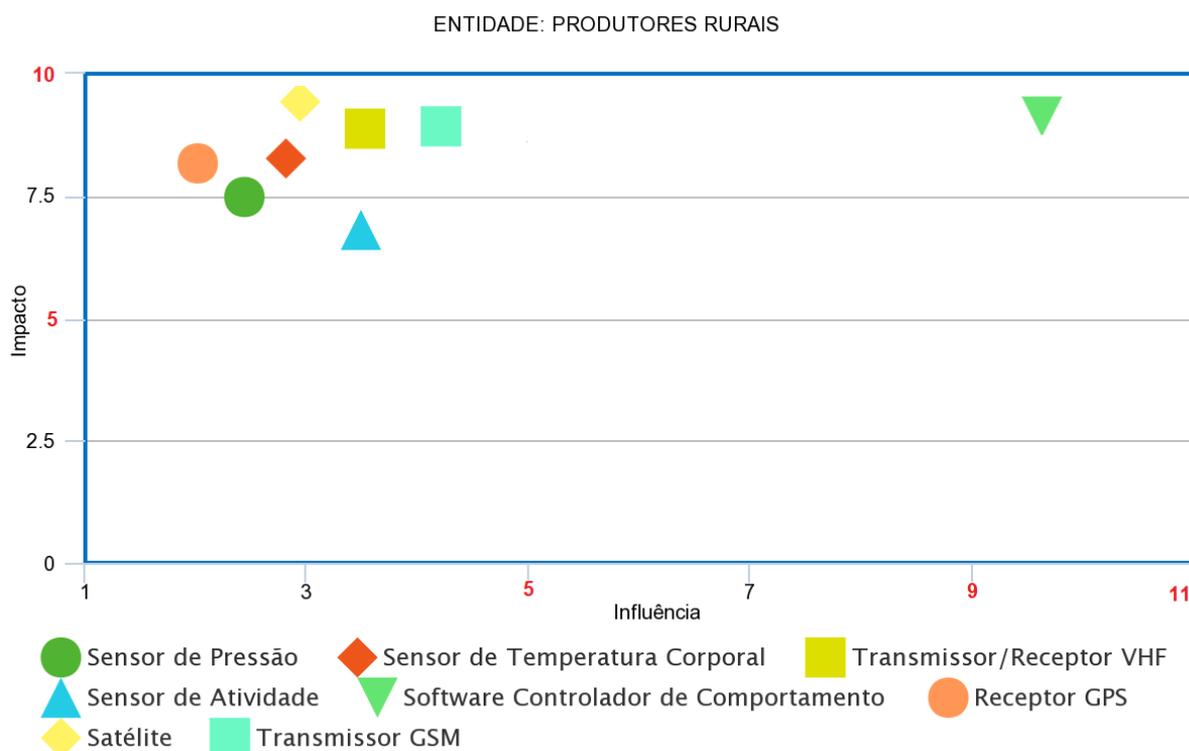
- Independência Operacional: independente da falha de um ou outro sensor ou receptor VHF, desde que ele receba as informações, ele consegue operar de forma independente;
- Independência Gerencial: o seu gerenciamento e a forma como os dados são comparados para obtenção das anomalias de comportamento são definidas e gerenciadas pelo próprio sistema;
- Distribuição Geográfica: esse sistema opera em um dispositivo separado geograficamente dos outros sistemas que compõem o *sistema de monitoramento de comportamento*;
- Comportamento emergente: como unidade central do sistema, ele recebe informações dos outros SCs, detecta anomalias de comportamento e emite relatórios e análises para as entidades interessadas.

Definido que o sistema a ser modelado realmente se trata de um SoS, o próximo passo consiste na identificação das entidades que influenciam o SoS e do desenvolvimento do gráfico de influência x impacto. O objetivo desta atividade é auxiliar na análise das mudanças solicitadas para os SCs, definindo quais das entidades terão prioridade nas suas solicitações. As entidades identificadas para esse sistema foram: *pesquisadores da EMBRAPA, alunos do curso de Zootecnia da UFERSA, equipe responsável por modelar o sistema, equipe responsável por projetar a implementação do sistema, equipe de desenvolvimento, produtores rurais, equipe de pesquisadores em Zootecnia da UFERSA.*

Na figura 25, são apresentados os níveis de influência e impacto para a entidade *produtores rurais*. No tocante a influência, como a maioria dos sistemas identificados provavelmente serão adquiridos no mercado, foram modelados baixos níveis de influência, com exceção do sistema *Software Controlador de Comportamento*, que, no futuro, ela exercerá altos níveis. Com relação ao impacto, foram modelados níveis altos de impacto, pois os requisitos dos sistemas adquiridos podem não atender as necessidades do projeto, o que acarretaria impactos significativos para essa entidade.

Para finalizar a fase de definição do contexto, uma sumarização das principais características sobre a viabilidade do projeto é apresentada a seguir. Como recomendado pela abordagem, devem ser analisados aspectos como: pessoal, tecnologia, integração de

Figura 25 – Influência x impacto para entidade produtores rurais



Fonte: Autoria Própria

sistemas legados, dentre outros. Para esse projeto, as justificativas para sua viabilidade são:

- **Objetivos:** Um projeto desta natureza envolve áreas como: ciência da computação, engenharia elétrica e zootecnia; assim, tratando a universidade como a organização responsável pelo sistema. Acredita-se que é do seu interesse, bem como dos profissionais dessas áreas, o desenvolvimento de um projeto de pesquisa multidisciplinar como esse, pois fomenta a pesquisa e o avanço do conhecimento, além da possibilidade de gerar artigos e patentes aos envolvidos;
- **Pessoal e tecnologia:** sobre a tecnologia, ela envolveria basicamente sensores, de diversos tipos, receptores e transmissores VHF, *softwares* e outros equipamentos de *hardware*. Assim, acredita-se que há tecnologia para a realização do projeto. Sobre o pessoal, devido a UFERSA possuir os cursos de ciência da computação, engenharia elétrica e zootecnia, além de cursos de pós graduação nessas áreas, acredita-se também que há capital humano disponível para o desenvolvimento do projeto;
- **Integração a outros sistemas:** de acordo com o *survey* aplicado, pode-se concluir que há poucos projetos dessa natureza na instituição. Assim, não haverá problema de integração com sistemas legados ou outros dispositivos.

5.2.3 Concepção e Modelagem do Sistema SisBovi

Nesta seção, para o sistema de controle e monitoramento da produção animal, os conceitos da fase de concepção e modelagem da abordagem MORE-SoS são apresentados. Os artefatos desenvolvidos nessa fase foram: o diagrama de definição de bloco (Missões), o diagrama de definição de bloco (Estrutura), os diagramas de requisitos e os diagramas de atividades.

5.2.3.1 Missões do Sistema SisBovi

Para a obtenção das missões do sistema SisBovi, utilizou-se a técnica análise de domínio, observando informações públicas de projetos semelhantes, como os apresentados pelas empresas Tigrinus, Nortronic e ChipInside. Além disso, foi executado um *survey* com alunos da UFERSA e pesquisadores da EMBRAPA. Esse estudo teve a finalidade de avaliar os objetivos do sistema por profissionais da área.

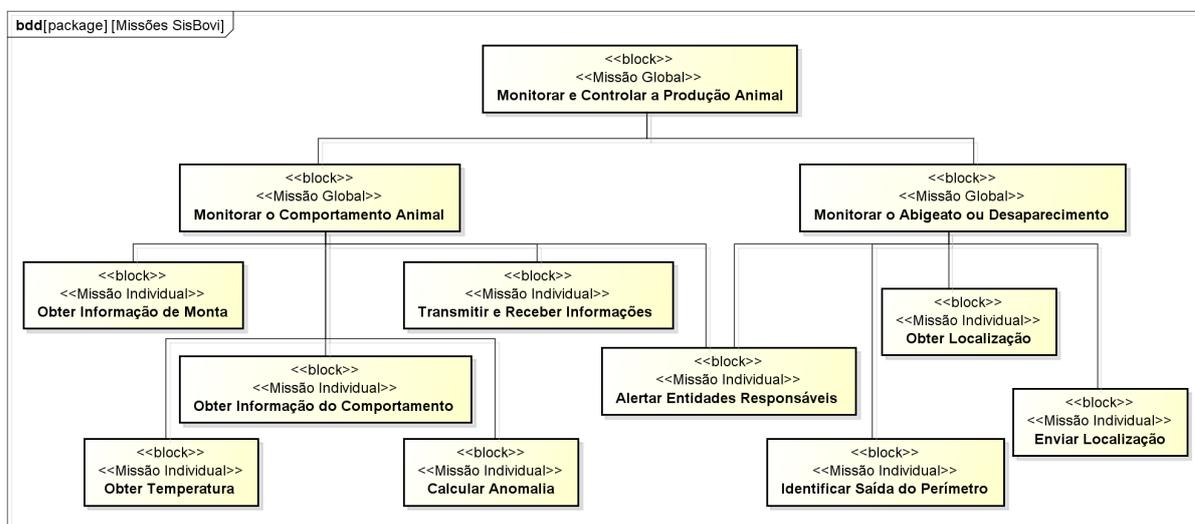
Como pode ser visto na Figura 26; o objetivo geral, identificado na fase de definição do contexto, foi modelado como a missão global *Monitorar e Controlar a Produção Animal*. Dessa missão, derivaram-se duas missões globais adicionais, que são: *Monitorar o Comportamento Animal* e *Monitorar o Abigeano ou Desaparecimento*. Delas, diversas missões individuais foram identificadas e modeladas.

A missão *Obter Informação de Monta* é responsável por detectar quando o animal está no cio, pois ajuda na identificação do período de inseminação e evita o deslocamento de um funcionário para ficar observando o animal. A missão *Obter Informação do Comportamento* é responsável por coletar informações comportamentais dos animais, registrando aspectos como: períodos de pastejo, ruminação, descanso e ócio. Isso é importante pois uma mudança no comportamento do animal pode caracterizar má alimentação, baixa qualidade da ração ou forragem, desconforto, estresse, doenças, dentre outros fatores. A missão *Obter Temperatura* é responsável por medir a temperatura corporal do animal, ela é útil para identificar possíveis doenças que têm como sintoma a febre. A missão *Transmitir e Receber Informações* é responsável por receber os sinais dos sensores presentes nos animais e enviar para o sistema que detectará anomalias e mudanças de comportamento. Por fim, a missão *Calcular Anomalia* é responsável por receber informações dos sensores, comparar com um padrão desejável de comportamento e identificar possíveis anomalias de comportamento.

Sobre as missões individuais derivadas da missão global *Monitorar o Abigeano ou Desaparecimento*, a missão *Obter Localização* é responsável pela obtenção das coordenadas dos animais por meio do sistema GPS. A missão *Enviar Localização* é responsável por enviar a localização do animal por meio da tecnologia GSM. A missão *Identificar Saída do Perímetro* é responsável por calcular se o animal está dentro do perímetro configurado. Por fim, a missão *Alertar Entidades Responsáveis* é encarregada da emissão de sinais de

alerta, sejam por anomalias no comportamento, ou desaparecimento do animal.

Figura 26 – Missões do sistema SisBovi



Fonte: Autoria Própria

5.2.3.2 Estrutura do Sistema SisBovi

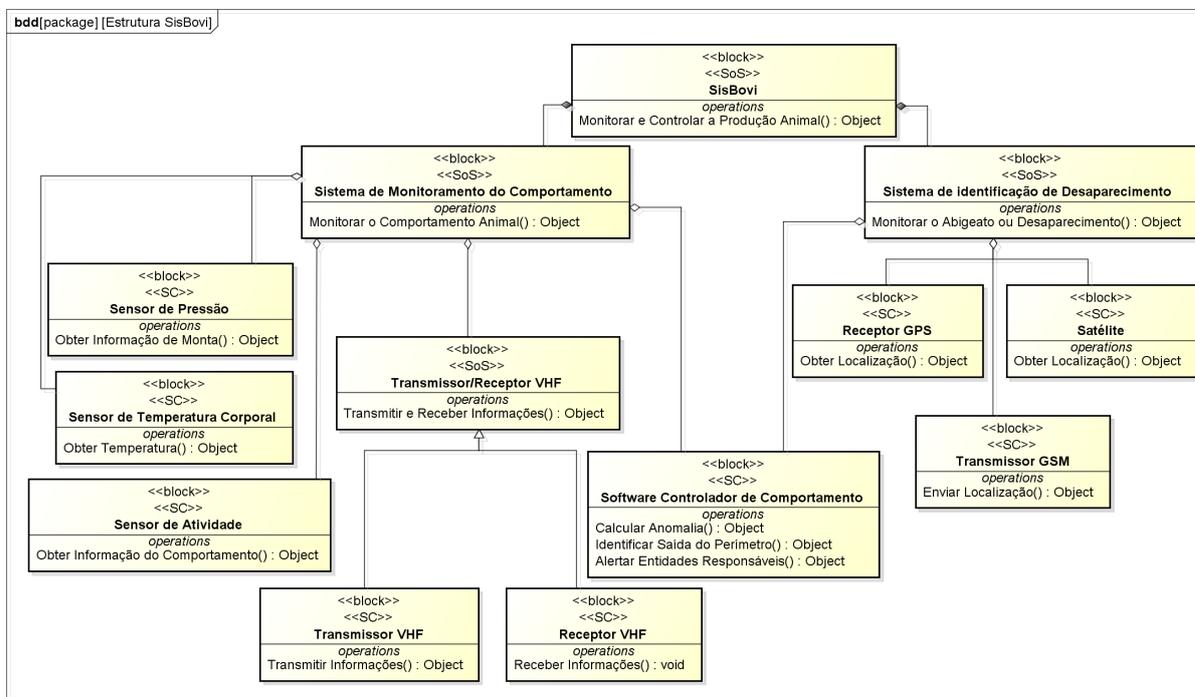
Nesta seção, a estrutura do sistema SisBovi foi modelada. Com base nas missões identificadas na atividade anterior e nas diretrizes da seção 5.1.2.2, o modelo presente na Figura 27 foi criado. Nele, o SoS SisBovi foi dividido em dois SoS distintos, que são: *Sistema de Monitoramento do Comportamento* e *Sistema de Identificação de Desaparecimento*. O primeiro foi dividido nos sistemas: *Sensor de Pressão*, *Sensor de Temperatura Corporal*, *Sensor de Atividade*, *Transmissor/Receptor VHF* e *Software Controlador de Comportamento*. O SoS *Sistema de Identificação de Desaparecimento* foi dividido nos sistemas *Receptor GPS*, *Satélite*, *Transmissor GSM* e *Software Controlador de Comportamento*, esse, que participa dos dois SoS identificados.

Algumas considerações sobre esse modelo: por meio de um processo de revisão *top-down*, a missão *Transmitir e Receber Informações* foi dividida e sua atribuição alocada a dois sistemas constituintes (*Transmissor VHF* e *Receptor VHF*). Com relação ao SC *Software Controlador de Comportamento*, percebeu-se que ele era capaz de realizar mais de uma missão. Assim, elas foram atribuídas a esse sistema e foram modeladas como operações desse bloco.

Sobre a forma de implantação desses sistemas, deve-se projetá-los da forma menos invasiva possível, pois podem causar desconforto e estresse nos animais, inviabilizando os sistemas. Como recomendação, indica-se o acoplamento dos sistemas a uma coleira colocada no pescoço do animal. Porém, no caso da implantação de todos os SCs modelados,

avaliar aspectos como tamanho e peso desses sistemas, pois, novamente, eles podem causar desconforto nos animais.

Figura 27 – Estrutura do Sistema SisBovi



Fonte: Autoria Própria

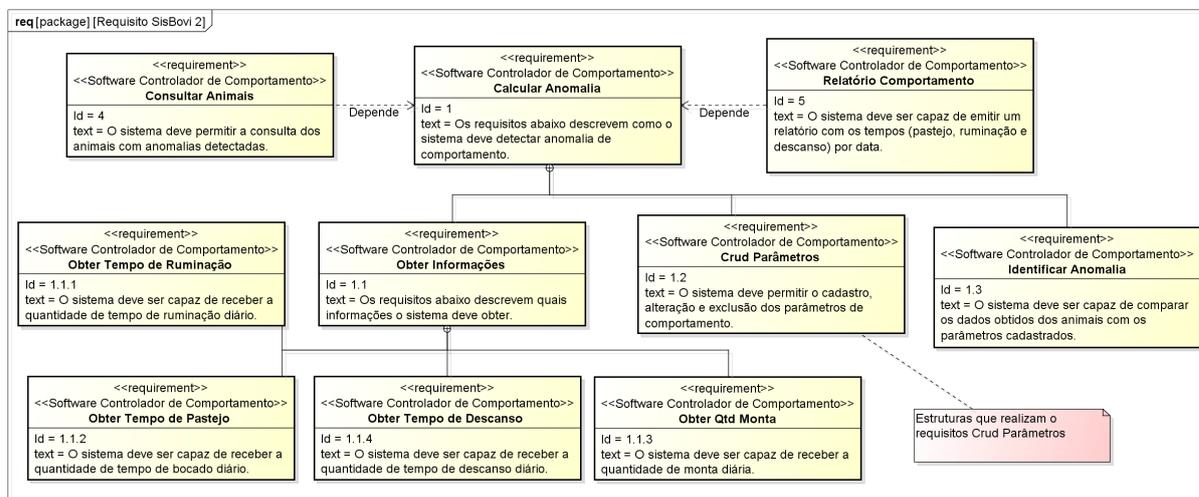
5.2.3.3 Requisitos dos Sistema Constituintes

Depois de definidos os objetivos gerais, representados pelas missões, e a estrutura do SoS. O próximo modelo desenvolvido foi o diagrama de requisitos. Nesse estudo de caso específico, como se pode perceber, diversos SCs são formados por sistemas embarcados, como os sensores, os quais provavelmente serão adquiridos e não desenvolvidos. Além disso, esses sistemas são adquiridos completos e totalmente funcionais. Assim, constatou-se que não era necessário a modelagem dos seus requisitos.

Os diagramas de requisitos elaborados referem-se ao sistema *Software Controlador de Comportamento*. O modelo apresentado na Figura 28 especifica alguns dos requisitos necessários para realizar a missão *Calcular Anomalia*, o modelo completo se encontra na Figura 35, Apêndice B. O sistema deve apresentar diversas funcionalidades para receber as informações provenientes de outros sistemas, como tempo de ruminção, descanso, dentre outros. Além disso, deve-se ter um CRUD (*Create, Retrieve, Update, Delete*) para manipular os parâmetros de comportamento ideal dos animais, assim como uma funcionalidade para comparar as informações coletadas com os parâmetros cadastrados, a fim de identificar anomalias de comportamento. Por fim, o sistema deve permitir um

mecanismo para informar as entidades necessárias; como pode ser visto no modelo, ele é executado por meio da consulta e emissão de relatórios.

Figura 28 – Requisitos da missão calcular anomalia (parcial)



Fonte: Autoria Própria

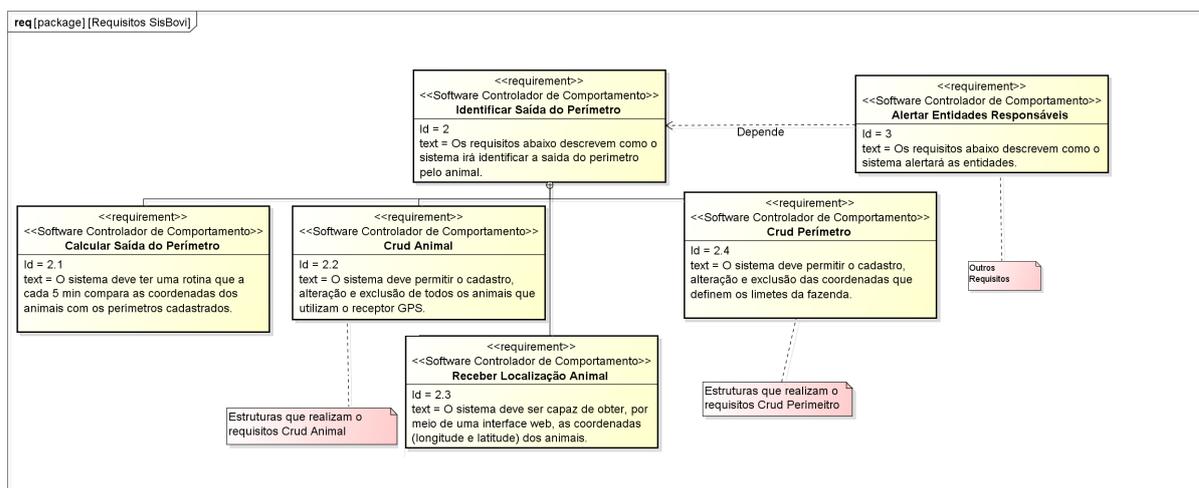
Na Figura 29, também estão presentes alguns dos requisitos do sistema *Software Controlador de Comportamento*, o modelo completo se encontra na Figura 36, Apêndice B. Nesse artefato estão modelados os requisitos que realizam a missão *Identificar Saída do Perímetro*. Como pode ser observado na figura, o sistema deve conter interfaces para obter a localização dos animais, além das funcionalidades de cadastro dos parâmetros que delimitam o perímetro da fazenda. Além disso, deve existir uma rotina que compara a localização obtida em tempo real com os parâmetros do perímetro definidos. Por fim, caso seja identificado a saída ou desaparecimento do animal, o sistema deve alertar as entidades responsáveis, por meio de mensagem de celular, ou emissão de relatório.

5.2.3.4 Atividades do Sistema SisBovi

Depois de identificados e modelados os requisitos dos SCs, diagramas de atividades foram modelados. Como recomendado pela abordagem, os diversos sistemas tornaram-se partições desse modelo, assim, pode-se visualizar com clareza quais os sistemas que as atividades pertencem. Dois modelos foram criados, representando os dois principais fluxos do sistema, que são: *Calcular Anomalia* e *Identificar Saída do Perímetro*. Na Figura 30, é apresentado o diagrama de atividades representando o fluxo *Calcular Anomalia*. Ele começa pela obtenção das informações por parte dos sensores. Essa informação é transmitida ao *Software Controlador de Comportamento* por meio dos receptores/transmissores VHF. Em paralelo, os parâmetros podem ser cadastrados, excluídos e alterados. As informações dos sensores e dos parâmetros são utilizadas para o cálculo da anomalia, que é representada

por uma atividade contínua, ou seja, permanece sempre em execução. Por fim, pode-se consultar se alguma anomalia foi detectada. O diagrama de atividades que representa o fluxo da funcionalidade *Identificar Saída do Perímetro* também foi modelado e pode ser encontrado na Figura 37, Apêndice B.

Figura 29 – Requisitos da missão identificar saída do perímetro (parcial)



Fonte: Autoria Própria

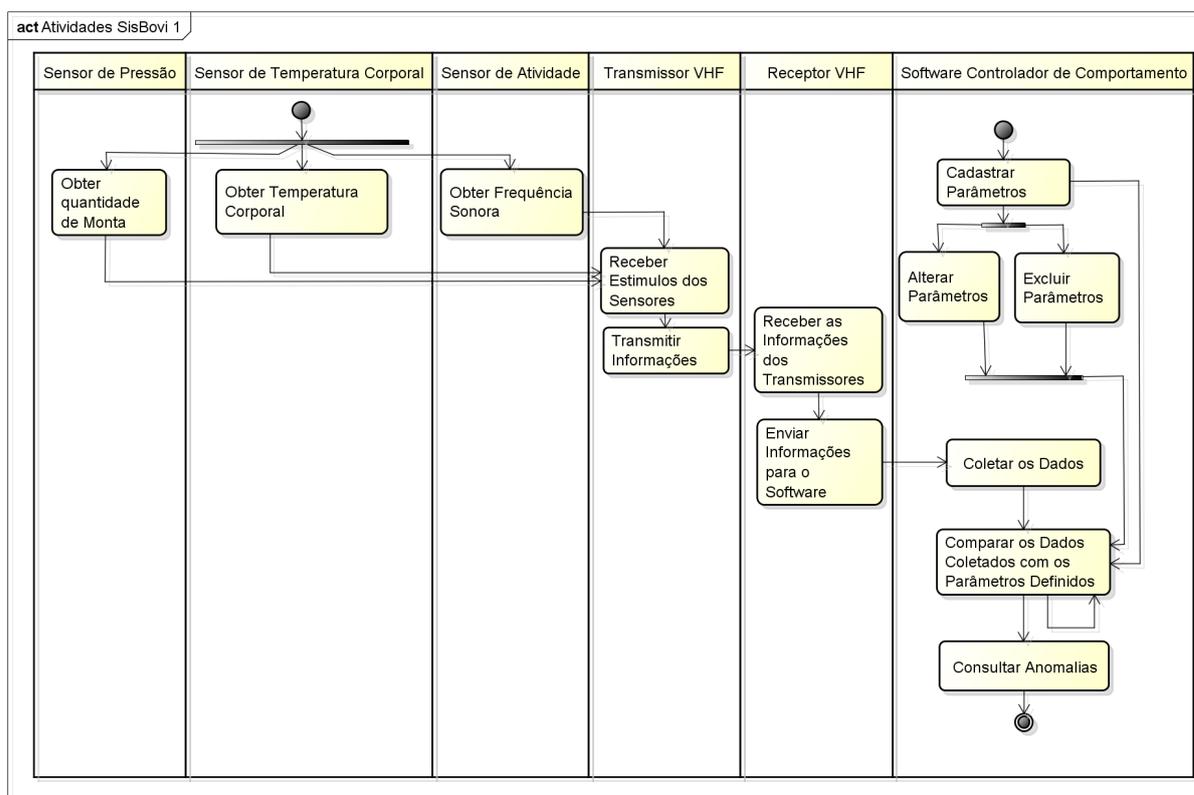
5.2.3.5 Estados dos Sistema Constituintes

Essa atividade da abordagem MORE-SoS é indicada para que se tenha uma melhor compreensão de um determinado sistema constituinte. Como discutido na seção 5.1.2.5, a MORE-SoS deixa a cargo dos analistas e engenheiros de *software* a análise de quais sistemas necessitam da modelagem dos seus estados. Em relação a este estudo de caso, o autor desse trabalho não encontrou a necessidade de modelar os estados dos SCs, pois em sua maioria, eles são compostos de sensores ou sistemas embarcados que são adquiridos prontos, ou seja, sem a necessidade de projeto e implementação, como por exemplo o sensor sonoro, de pressão, receptores e transmissores VHF, dentre outros sistemas. Assim, diferente do semáforo inteligente apresentado no capítulo anterior, no qual os estados se mostram importantes para o entendimento desse sistema, modelar os estados dos diversos sensores apenas geraria complexidade excessiva a essa fase do desenvolvimento.

5.3 Análise do Estudo de Caso

O principal objetivo desta seção é mostrar a análise dos resultados obtidos por meio do *survey* realizado, sua condução teve como objetivos: verificar a importância da pesquisa proposta e avaliar os objetivos gerais do sistema SisBovi identificados por meio da técnica análise de domínio, dessa forma, é possível identificar a efetividade da abordagem

Figura 30 – Atividades do fluxo calcular anomalia



Fonte: Autoria Própria

MORE-SoS. Nas seções seguintes são abordados aspectos da formulação e condução do questionário aplicado, assim como os resultados obtidos ao longo da sua execução.

5.3.0.1 Formulação do Questionário

Questionários são uma forma rápida e simples para avaliar as opiniões, objetivos e anseios das pessoas. Mas por serem simples, se mal conduzidos, podem levar a um viés considerável. O uso de questionários envolve algumas atividades, como: elaboração das perguntas e respostas, definição do público alvo, avaliação das respostas e análise dos resultados (WAINER, 2007). Para a avaliação dos objetivos gerais do sistema a ser modelado no estudo de caso, um *survey* foi aplicado aos alunos do curso de zootecnia da UFERSA, nos níveis de graduação, mestrado e doutorado. Além disso, ele também foi aplicado a um conjunto de pesquisadores da EMBRAPA. As seções seguintes abordam o desenvolvimento do questionário, sua condução e, por fim, a análise dos resultados.

De acordo com Wainer (2007), no processo de aplicação de um *survey* é recomendável a utilização de questionários elaborados, pois eles, muitas vezes, já passaram por um processo de verificação e validação, o que garante um maior grau de confiabilidade. Porém, na necessidade de elaboração, recomendam-se algumas diretrizes, como: as perguntas

devem ser claras, simples, concisas, diretas, neutras, não fraseadas na negativa e não compostas por dois ou mais questionamentos. Com relação a elaboração das respostas, é importante que haja um balanço de igual intensidade e em direções opostas.

Como recomendado, tentou-se encontrar questionários elaborados para o tema apresentado no estudo de caso, porém, a busca não obteve resultados satisfatórios. Assim, optou-se por elaborar um questionário específico para avaliação dos objetivos de um sistema de controle e monitoramento da produção animal.

Como pode ser visto no Apêndice A, as questões seguiram as recomendações: elas são claras, sucintas, não há questões na negativa e nem compostas. Com relação às respostas, tentou-se também manter o balanço, optando por itens variando do muito baixo até o muito alto. Sobre a estrutura do questionário em si, ele foi dividido em três seções: (i) perfil profissional, identificação do nível de conhecimento dos profissionais sobre o tema; (ii) viabilidade do estudo, identificação da relevância do tema; (iii) funcionalidades do sistema, avaliação dos principais objetivos de um sistema para controle e monitoramento da produção animal.

O questionário conta com dez questões, sendo quatro delas referentes ao perfil profissional, três questões sobre a pesquisa e sua relevância e, por fim, três questões sobre aspectos da produção animal e objetivos do sistema. O baixo número de questões foi pensado para tornar o questionário objetivo e não exaustivo, tentando, assim, elevar a quantidade de respostas. É importante salientar que uma pesquisa prévia tinha sido executada e era de conhecimento alguns dos possíveis objetivos do sistema. Assim, as questões tinham como objetivo avaliar as funcionalidades modeladas.

5.3.0.2 Condução do questionário

No final de fevereiro de 2018, um questionário piloto foi desenvolvido e aplicado a um conjunto pequeno de alunos do curso de mestrado em ciência animal da UFERSA. Após avaliação das respostas e das sugestões obtidas, o questionário foi modificado. No início de março de 2018, a versão final do questionário foi enviada para os alunos de graduação, mestrado e doutorado em zootecnia da UFERSA. Posteriormente, a pesquisa foi expandida e o questionário enviado para cinquenta pesquisadores da EMBRAPA, os contatos desses profissionais foram obtidos pelo próprio *site* da empresa. Após vinte (20) dias do início da execução, trinta e três respostas foram obtidas.

5.3.0.3 Análise dos Resultados

Nesta seção, os principais resultados do *survey* aplicado são sumarizados e apresentados, analisando aspectos como os perfis profissionais, importância da pesquisa e, principalmente, aspectos funcionais dos sistemas modelados.

Como pode ser visto na Figura 31, sessenta e três virgula seis por cento (63,6%) dos respondentes são mestres e doutores, dezoito virgula dois por cento (18,2%) são bacharéis. Sobre a experiência com projetos de informatização animal, setenta e cinco virgula oito por cento (75,8%) responderam que têm experiência acadêmica, esse fato era esperado, pois o público alvo é composto de alunos e pesquisadores da EMBRAPA. Sobre o conhecimento do tema zootecnia de precisão, setenta e cinco virgula sete por cento (75,7%) tem conhecimento médio, alto ou muito alto. Portanto, conclui-se que os respondentes são pessoas preparadas e que apresentam domínio do tema em questão.

Figura 31 – Perfil profissional



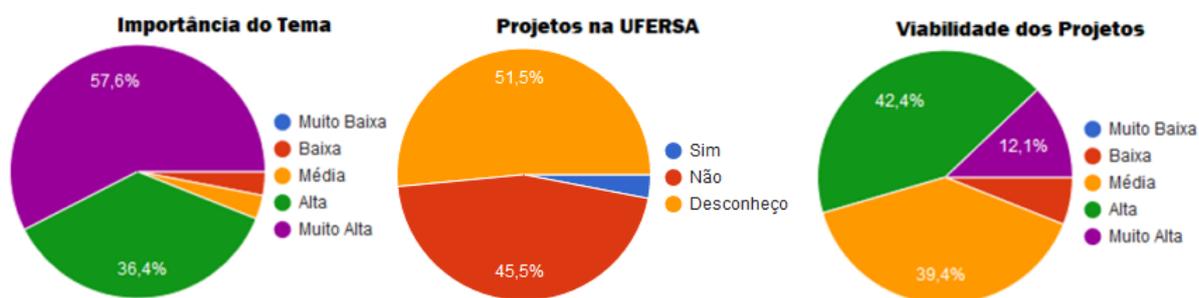
Fonte: Autoria Própria

Sobre a viabilidade da pesquisa, como pode ser visto na Figura 32, noventa e quatro por cento (94%) responderam que o tema é de relevância alta ou muito alta. Sobre a existência de projetos dessa natureza na UFERSA, apenas três por cento (3%) indicaram que sim, quarenta e cinco virgula cinco por cento (45,5%) responderam que não existem, e os outros cinquenta e um por cento (51%) responderam desconhecer. Isso se deve, principalmente, pela grande quantidade de respostas por parte dos pesquisadores da EMBRAPA, que não conhecem a instituição. Por fim, sobre a viabilidade de projetos com o tema, cinquenta e quatro virgula cinco por cento (54,5%) responderam alta e muito alta e trinta e nove virgula quatro por cento (39,4%) responderam média. Conclui-se que o tema é relevante, viável e que poucos pesquisadores na UFERSA trabalham com esse tema de pesquisa.

Sobre os principais aspectos da criação bovina, o quesito mais importante foi a higienização do ambiente, com sessenta e três virgula seis por cento (63,6%) das respostas, seguido pela exposição à temperatura, com cinquenta e um virgula cinco por cento (51,5%), identificação da temperatura corporal, com quarenta e oito virgula cinco por cento (48,5%), e, por fim, identificação da localização exata do animal e local com ventilação, com trinta e nove virgula quatro por cento (39,4%) das respostas.

O *survey* foi aplicado com o objetivo de avaliar os objetivos identificados no estudo de caso por meio da técnica análise de domínio. Para formalizar essa avaliação, um teste de hipótese foi executado. Segundo Wohlin *et al.* (2012) ele pode ser definido como uma

Figura 32 – Viabilidade da pesquisa



Fonte: Autoria Própria

regra específica que avalia, aceitando ou rejeitando, uma alegação sobre uma população de acordo com as provas fornecidas por uma amostra de dados. Além disso, para sua execução, algumas características devem ser compreendidas, são elas:

- Hipótese nula: afirmação assumida como verdadeira para a construção do teste;
- Hipótese alternativa: afirmação que deve ser verdadeira se a hipótese nula for falsa;
- Nível de significância: probabilidade de se aceitar ou rejeitar uma hipótese.

Para a realização do teste, os objetivos definidos na questão nove do questionário aplicado foram mapeados como uma variável X. As hipóteses e o nível de significância definidos são apresentados a seguir:

- Hipótese nula: o objetivo X é estritamente necessário para o sistema;
- Hipótese alternativa: o objetivo X não é estritamente necessário para o sistema;
- Nível de significância: $> 30\%$, ou seja, objetivos com mais de 30% das respostas definem como verdade a hipótese nula.

Na Figura 33, um resumo das respostas obtidas para a questão nove é apresentado, como pode ser observado, os objetivos que conseguiram um nível de significância acima de trinta por cento (30%) foram: *o sistema deve ser capaz de obter o tempo de pastejo, ruminação e descanso dos animais, O sistema deve permitir a consulta da localização exata dos animais dos animais, A localização deve ser obtida por meio de sensores que monitoram o animal, O sistema deve permitir a identificação do abigato, O sistema deve permitir a consulta da temperatura corporal dos animais, O sistema deve permitir o controle e monitoramento da temperatura ambiental e, por fim, O sistema deve informar dados sobre a exposição a radiação sofrida pelos animais.* Dos sete objetivos que alcançaram o nível de

significância, cinco deles foram modelados, o que representa setenta e um virgula quatro por cento (71,4%). Além disso, o objetivo com maior nível de significância, ou seja, a principal funcionalidade do sistema foi identificada e modelada. Esse dado oferece indícios de que a técnica análise de domínio recomendada pela MORE-SoS e a sua execução foram eficientes na identificação dos objetivos mais relevantes do SoS.

Figura 33 – Objetivos do sistema



Fonte: Autoria Própria

Sobre os outros objetivos que não obtiveram índice de significância satisfatório, pode-se inferir que eles não são prioritários, porém, isso não significa que eles não podem ser adicionados a solução final. O mesmo acontece com as funcionalidades sugeridas pelos respondentes na questão dez (10) do questionário. Algumas delas são apresentadas a seguir:

- "*Peso vivo (balança eletrônica), detector de cio e mudança de comportamento - por meio de coleira eletrônica*";
- "*Monitoração do consumo de ingestão de água e alimento*";
- "*Se o animal está em atividade ou ócio*";
- "*Explorar vocalização*".

5.4 Ameaças a Validade

Segundo Perry, Porter e Votta (2000), ameaças à validade são influências que podem limitar a habilidade de interpretar as conclusões provenientes dos dados extraídos. Na condução do estudo de caso, algumas ameaças foram percebidas, umas relacionadas a validade da abordagem MORE-SoS e outras relacionadas a validade do estudo de caso realizado.

Com relação a validade da abordagem MORE-SoS, as ameaças percebidas foram:

- Estudo de caso: o estudo de caso realizado garante que os artefatos e modelos preconizados pela abordagem MORE-SoS podem ser construídos, além de oferecer um exemplo de utilização. Porém, ele não garante a eficiência e eficácia da MORE-SoS, principalmente em comparação com outras abordagens;
- Validação: acredita-se que, para a real validação da abordagem MORE-SoS, faltou a utilização de uma técnica mais pragmática, em conjunto com o estudo de caso realizado. Uma das opções seria a utilização da técnica TAM (Modelo de Aceitação de Tecnologia). Ela poderia ser utilizada em conjunto com profissionais da área de engenharia de *software* e seriam avaliados aspectos como: importância, viabilidade e facilidade de uso da abordagem MORE-SoS.

Com relação a validade do estudo de caso realizado, as ameaças percebidas foram:

- População pequena: na condução do *survey*, foram obtidas trinta e três (33) respostas, dessas, vinte e cinco (25) pertenciam a pesquisadores da EMBRAPA e oito (8) pertenciam a discentes dos cursos de zootecnia da UFERSA. Apesar da qualidade dos respondentes; essa amostra ainda pode ser considerada pequena;
- Nível de significância: um nível de significância de apenas trinta por cento (30%) pode ser considerado baixo para garantir que uma funcionalidade deve, estritamente, pertencer ao sistema. Porém, esse valor foi escolhido analisando o mínimo de respostas necessárias para aferir a importância da funcionalidade para o SoS;
- Funcionalidades: acredita-se que, na elaboração do *survey*, uma questão discursiva sobre as funcionalidades do sistema seria o ideal, pois questões objetivas podem oferecer um viés considerável às respostas. Porém, o motivo da utilização de questões objetivas foi garantir uma quantidade de respostas razoável, visto que questionários extensos e com questões subjetivos limitam a quantidade de respostas;
- Elicitação dos objetivos do SoS: o processo de obtenção dos objetivos do SoS utilizou apenas a técnica de análise de domínio, acredita-se que os objetivos identificados

não foram comprometidos pela utilização de apenas uma técnica, porém, caso outras técnicas como entrevistas ou *brainstorm* tivessem sido utilizados, os objetivos teriam se tornado mais completos e robustos.

5.5 Comparativo entre a MORE-SoS e os Trabalhos Relacionados

Após apresentados os principais conceitos, fases, atividades e diretrizes da abordagem MORE-SoS, além de um estudo de caso exemplificando sua utilização, é possível efetuar uma análise mais profunda sobre essa abordagem em relação aos processos apresentados nos trabalhos relacionados. Assim, na Tabela 10, é apresentado um comparativo mais exaustivo entre essas três abordagens.

Tabela 10 – Comparativo entre a MORE-SoS e os trabalhos relacionados

Aspecto	SoS-ACRE	MORE-SoS	Ab. Baseada em SysML
Objetivo	Gerenciar todo o processo de ER de um SoS.	Derivar e modelar as missões e requisitos dos SCs de um SoS.	Definir os requisitos do SoS e modelar sua arquitetura.
Estrutura	Ontologia, <i>framework</i> , processo.	Definição do contexto, concepção e modelagem, <i>framework</i> .	Contexto do sistema, comportamento de caixa preta, projeto de arquitetura.
Fases da ER	Toda a ER.	Elicitação e especificação dos requisitos.	Elicitação e especificação dos requisitos.
Diretrizes para ex. das atividades	Não apresenta diretrizes para execução das fases do processo.	Apresenta diretrizes para execução de todas as atividades propostas.	Apresenta diretrizes para execução de todas as fases propostas.
Modelos utilizados	Diagrama de definição de bloco, diagrama de casos de uso e diagrama de requisitos.	Diagrama de definição de bloco, diagrama de requisitos, diagrama de atividade, diagrama de máquina de estados.	Diagrama de definição de bloco, diagrama de sequência, diagrama de casos de uso.
Tipo	Orientada a processos.	Orientada a modelos.	Orientada a modelos.
Domínio do sistema	Não apresenta atividades relacionadas ao domínio do sistema.	Apresenta atividades relacionadas ao domínio do sistema.	Não apresenta atividades relacionadas ao domínio do sistema.
Validação dos requisitos	Define atividades específicas para validação dos requisitos.	Não define atividades específicas para validação dos requisitos.	Não Define atividades específicas para validação dos requisitos.
Exemplo de utilização	Exemplifica por meio de um estudo de caso.	Exemplifica por meio de um estudo de caso.	Exemplifica por meio de um estudo de caso.

Como pode ser observado na tabela descrita anteriormente, a maior vantagem da abordagem SoS-ACRE sobre as demais é que ela propõem um processo geral que cobre todas as etapas da ER. Além disso, ela é a única abordagem que define atividades para as etapas de verificação e validação dos requisitos. Porém, ela tem a desvantagem de não especificar como cada etapa do seu processo deve ser executada. A abordagem MORE-SoS apresenta como vantagens a definição de diretrizes claras para execução das atividades propostas. Além disso, ela é orientada a modelos, o que proporciona diversas vantagens, como as apresentadas na seção 2.4.2. Por fim, a MORE-SoS é a única abordagem que dedica atividades específicas para definir o domínio do SoS a ser desenvolvido, porém, ela tem a desvantagem de não cobrir todas as atividades da ER.

Com relação a abordagem baseada em SysML, ela também tem como vantagens a definição de diretrizes para execução das suas atividades. Além disso, ela também é orientada a modelos. Como desvantagens dessa abordagem, é possível citar que ela não cobre todas as etapas da ER, não define atividades para compreensão do domínio do SoS e apresenta uma complexidade excessiva.

5.6 Considerações Finais

Neste capítulo, foram abordadas os principais aspectos da versão final da abordagem proposta por esse trabalho, a MORE-SoS. Foram apresentados seus conceitos, elementos, fases, diretrizes e atividades. Também foram apresentados os artefatos gerados pelo estudo de caso realizado para validar essa abordagem. Por fim, foi efetuada uma análise sobre o *survey* executado, foram discutidos aspectos da sua formulação, condução e resultados.

6 Conclusão

Este trabalho foi iniciado por meio de uma revisão bibliográfica com o objetivo de obter as principais informações que norteiam o estudo dos SoSs e da ER. Após essa etapa, foi planejada e conduzida uma revisão sistemática da literatura sobre engenharia de requisitos e sistema de sistemas. Do seu resultado, observou-se a falta de abordagens com objetivo de derivar e especificar os requisitos dos SCs de um SoS, pois muitas delas concentravam-se apenas na identificação e modelagem das missões do SoS. Outro fato observado foi a pouca quantidade de abordagens que cobriam todas as etapas da ER. Assim, sentiu-se a necessidade de propor, baseada em abordagens existentes na literatura, uma abordagem para desenvolver esse tipo de sistema que conseguisse suprir algumas das lacunas identificadas pela revisão sistemática.

Dessa forma, este trabalho propõe a abordagem orientada a modelos MORE-SoS, que tem como objetivo principal a especificação e modelagem das missões e requisitos do SoS. Para isso, estruturou-se a abordagem em três elementos principais, que são: (i) fase de definição do contexto, responsável pela definição do contexto e ambiente ao qual o SoS está inserido; (ii) *framework*, responsável pela definição dos modelos e artefatos a serem criados pela abordagem; (iii) fase de concepção e modelagem, responsável por desenvolver os elementos do *framework*.

A fase de concepção e modelagem pode ser considerada o principal elemento da abordagem, pois é nela que as missões, requisitos e outros aspectos dos sistemas são especificados e modelados. Ela é composta por um conjunto de atividades, que são: identificação e modelagem das missões do SoS; identificação e modelagem da estrutura do SoS; identificação, modelagem e especificação dos requisitos; modelagem das atividades do SoS e, por fim, modelagem dos estados dos SCs. É importante salientar que em cada atividade deve-se gerar um artefato e que a passagem de uma atividade para outra é feita por meio de diretrizes aplicadas nos modelos gerados.

Para auxiliar a condução deste trabalho, o paradigma *Design Science* foi utilizado. Ele é fundamentalmente uma técnica de resolução de problemas que procura criar inovações que definam ideias, práticas e produtos por meio dos quais a análise, o projeto e a implementação podem ser efetivamente e eficientemente realizados (HEVNER *et al.*, 2004). Com relação ao processo de criação da abordagem MORE-SoS, a metodologia pesquisa-ação foi utilizada. Nela, a partir de abordagens presentes na literatura, definiu-se uma versão inicial da abordagem, chamada de REAp-SoS. Dessa concepção inicial, iniciou-se um processo iterativo e incremental por meio da especificação e modelagem de um SoS de controle e monitoramento de tráfego urbano. No processo de desenvolvimento desse

estudo de caso, algumas fases, atividades e diretrizes foram modificadas de tal forma que, ao final do estudo, gerou-se a abordagem MORE-SoS, apresentada no Capítulo 5.

Por fim, um estudo de caso com a finalidade de validar a nova abordagem foi executado. Nele, um SoS para o monitoramento e controle da produção animal foi especificado e modelado. Para a obtenção dos objetivos e funcionalidades, a técnica de análise de domínio foi utilizada, e os requisitos foram derivados por meio da observação de outros sistemas desse tipo presentes no mercado, como os apresentados pelas empresas Tigrinus, Nortronic e ChipInside. Para avaliar a abordagem, a técnica utilizada foi o *survey*, que foi aplicado junto a comunidade acadêmica do curso de Zootecnia da UFERSA e com os pesquisadores da EMBRAPA.

6.1 Contribuições da Pesquisa

A principal contribuição deste trabalho foi a definição de uma abordagem específica de ER para SoS. Ela apresenta fases, atividades, modelos e estereótipos voltados especificamente para esse tipo de sistema. Assim, acredita-se que ela contribui para o seu desenvolvimento, principalmente com relação a especificação e modelagem dos seus requisitos. Além dessa, outras contribuições deste trabalho são:

- Planejamento e execução de uma RSL, que teve como objetivo mapear os conhecimentos na área de ER e SoS, identificando aspectos como: abordagens, desafios e modelos de ER aplicados especificamente aos SoSs, assim como a identificação das principais lacunas de pesquisa;
- Especificação e modelagem de um SoS para monitoramento e controle do tráfego urbano. Os conceitos modelados nesse estudo de caso podem servir de base para outros projetos na área;
- Especificação e modelagem de um SoS para monitoramento e controle da produção animal, a partir do qual as principais missões e requisitos foram identificados e modelados. Assim, os artefatos gerados pelo estudo de caso podem ser utilizados para um posterior projeto e implementação desse sistema;
- Definição, na abordagem MORE-SoS, de uma etapa para a análise do domínio do SoS. Essa atividade não está presente na maioria das abordagens identificadas na RSL. Ela é responsável pela definição de diversos aspectos do contexto do SoS, como; confirmação da existência do SoS, identificação das entidades e os seus níveis de influência e impacto, além da análise da viabilidade do sistema;
- Os conceitos presentes na abordagem MORE-SoS; como por exemplo: a análise do domínio e confirmação da existência do SoS, estereótipos específicos para a

modelagem de SoS, diretrizes formais para o desenvolvimento dos modelos sugeridos, dentre outros; podem servir de base para que sejam propostas outras técnicas e abordagens de ER para SoS. Além disso, outras abordagens de implementação, projeto e testes podem ser propostas para complementar o trabalho iniciado pela MORE-SoS.

6.2 Limitações da Abordagem

Apesar dos avanços que a abordagem MORE-SoS obteve em relação a sua versão inicial, ela ainda apresenta algumas limitações, dentre elas, é possível citar:

- A falta de diretrizes para a revisão dos requisitos, visto que devido a complexidade dos SoSs e a grande quantidade de requisitos, é importante executar um processo de verificação dos requisitos identificados e dos modelos construídos;
- A falta de diretrizes para a negociação dos requisitos entre os *stakeholders* dos SCs, visto que os requisitos pertencem a múltiplos domínios e provenientes de múltiplos *stakeholders*, que têm perspectivas diferentes sobre os sistemas;
- A falta de diretrizes para o gerenciamento dos requisitos, visto que os requisitos são instáveis, incompletos e em constante modificação. Assim, também é importante manter os requisitos dos SCs consistentes por meio de um processo sistemático de gerenciamento;
- A falta de diretrizes para definir como se dará a passagem da etapa de análise para a etapa de projeto e implementação, pois é importante definir critérios para facilitar o desenvolvimento dos modelos de projeto dos sistemas. Os modelos de atividade e de máquina de estados já auxiliam nessa tarefa, mas acredita-se que outros modelos também seriam necessários, como o modelo de casos de uso, sequência, dentre outros.

Os processos de validação e gerenciamento dos requisitos são atividades importantes da ER. Assim, o objetivo inicial deste trabalho era que a abordagem MORE-SoS englobasse essas atividades. Porém, na condução deste projeto, sentiu-se a necessidade de propor atividades mais robustas para a validação e gerenciamento dos requisitos de um SoS, e para isso, seria necessário um maior tempo de pesquisa, visto que o prazo para conclusão de um curso de mestrado é curto. Assim, essas atividades serão trabalhadas em futuros projetos.

6.3 Trabalhos Futuros

Como perspectivas de trabalhos futuros decorrentes deste trabalho, é possível citar:

- A implementação de uma ferramenta que automatize os processos de transformação entre os modelos criados na fase de concepção e modelagem da abordagem MORE-SoS;
- Proposta de uma abordagem formal para revisão, verificação e validação dos requisitos de sistemas complexos, a ser adicionada como fase da MORE-SoS;
- Proposta de uma abordagem formal para negociação e solução de conflitos entre os *stakeholders* do SoS, a ser adicionada como fase da MORE-SoS;
- Proposta de uma abordagem formal para o gerenciamento de requisitos de SoS, a ser adicionada como fase da MORE-SoS;
- Proposta de uma abordagem específica para o desenvolvimento de SoS, utilizando os conceitos da abordagem MORE-SoS como base;
- Desenvolvimento de um protótipo do sistema SisBovi.

6.4 Produções Científicas

Os resultados obtidos ao longo deste trabalho foram submetidos e/ou aprovados nas seguintes conferências e revistas:

- – Título: *Requirement Engineering Approach for System of System*.
– Veículo: *6th International Conference on Computational Science and Engineering (CSE 2018)*.
– Situação: Aceito.
- – Título: *Requirements Engineering for System of Systems: A Systematic Literature Review*.
– Veículo: *IET Software Journal*.
– Situação: Submetido.
- – Título: *MORE-SoS: Model Oriented Approach to Requirements Engineering for SoS*
– Veículo: *Software & Systems Modeling Journal*;
– Situação: A ser submetido.

Referências

- ABDALLA, G. *et al.* A Systematic Literature Review on Knowledge Representation Approaches for Systems-of-Systems. In: *2015 IX Brazilian Symposium on Components, Architectures and Reuse Software*. [S.l.: s.n.], 2015. p. 70–79.
- BABUR, Statistical Analysis of Large Sets of Models. In: . New York, NY, USA: ACM, 2016. (ASE 2016), p. 888–891. ISBN 9781450338455. Disponível em: <<http://doi.acm.org/10.1145/2970276.2975938>>.
- BAUDRY, B.; NEBUT, C.; TRAON, Y. L. Model-Driven Engineering for Requirements Analysis. In: *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*. [S.l.: s.n.], 2007. p. 459–459.
- BELLOIR, N. *et al.* Using Relax Operators into an MDE Security Requirement Elicitation Process for Systems of Systems. In: . New York, NY, USA: ACM, 2014. (ECSAW '14), p. 32:1–32:4. ISBN 9781450327787. Disponível em: <<http://doi.acm.org/10.1145/2642803.2642835>>.
- BENITTI, F. B. V. As a Teacher, I Want to Know What to Teach in Requirements Engineering So That Professionals Can Be Better Prepared. In: . New York, NY, USA: ACM, 2017. (SBES'17), p. 318–327. ISBN 9781450353267. Disponível em: <<http://doi.acm.org/10.1145/3131151.3131185>>.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistema com UML*. 3th. ed. [S.l.]: Elsevier Brasil, 2015. Google-Books-ID: elvjBwAAQBAJ.
- BIOLCHINI, J. *et al.* *Systematic Review in Software Engineering*. [S.l.], 2005.
- BUARQUE, S. C. *Construindo o desenvolvimento local sustentável: metodologia de planejamento*. [S.l.]: Editora Garamond, 2008. Google-Books-ID: UDzMIAM2ldwC. ISBN 9788586435768.
- CAVALCANTE, E. *et al.* Revisiting Goal-Oriented Models for Self-Aware Systems-of-Systems. In: *2015 IEEE International Conference on Autonomic Computing*. [S.l.: s.n.], 2015. p. 231–234.
- CAVALCANTE, E. *et al.* Thinking Smart Cities As Systems-of-Systems: A Perspective Study. In: . New York, NY, USA: ACM, 2016. (SmartCities '16), p. 9:1–9:4. ISBN 9781450346672. Disponível em: <<http://doi.acm.org/10.1145/3009912.3009918>>.
- CECCARELLI, A. *et al.* Introducing Meta-Requirements for Describing System of Systems. In: *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*. [S.l.: s.n.], 2015. p. 150–157.
- COLOMBO, A.; VECCHIO, D. D. Efficient Algorithms for Collision Avoidance at Intersections. In: . New York, NY, USA: ACM, 2012. (HSCC '12), p. 145–154. ISBN 9781450312202. Disponível em: <<http://doi.acm.org/10.1145/2185632.2185656>>.

- DAHMAN, J. *et al.* A model of systems engineering in a system of systems context. In: *Proceedings of the Conference on Systems Engineering Research*. [S.l.: s.n.], 2008. (CSER '08).
- DELLIGATTI, L. *SysML Distilled: A Brief Guide to the Systems Modeling Language*. [S.l.]: Addison-Wesley, 2013. Google-Books-ID: 3bMJAgAAQBAJ. ISBN 9780133430332.
- Department of Defense. Systems Engineering Guide for Systems of Systems. n. 01, Technical Report, Office of the Deputy Under Secretary of Defense for Acquisition and Technology, 01 2008.
- DUARTE, F. L.; CASTRO, A. F.; QUEIROZ, P. G. G. Reap-sos: A requirement engineering approach for system of systems. In: *The 6th International Conference on Computational Science and Engineering (CSE 2018)*. [S.l.: s.n.], 2018. p. 131 – 148. ISBN 9783642290435. ISSN 2231 - 5403.
- ESPERANÇA, V. N.; LUCRÉDIO, D. Late Decomposition of Applications into Services Through Model-Driven Engineering. In: . New York, NY, USA: ACM, 2017. (SBES'17), p. 164–173. ISBN 9781450353267. Disponível em: <<http://doi.acm.org/10.1145/3131151.3131165>>.
- FIRESMITH, D. Profiling systems using the defining characteristics of systems of systems. n. Technical Report, CMU/SEI-2010-TN-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2010.
- FITZGERALD, J.; BRYANS, J.; PAYNE, R. A Formal Model-Based Approach to Engineering Systems-of-Systems. In: . Springer, Berlin, Heidelberg, 2012. (IFIP Advances in Information and Communication Technology), p. 53–62. ISBN 9783642327742 9783642327759. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-32775-9_6>.
- FLANIGAN, D.; BROUSE, P. 11.4.2 Measuring the Requirements Allocation Capacity within a System of Systems. *INCOSE International Symposium*, v. 22, n. 1, p. 1671–1682, jul. 2012. ISSN 2334-5837. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2012.tb01429.x/abstract>>.
- FLANIGAN, D.; BROUSE, P. System of Systems Requirements Capacity Allocation. *Procedia Computer Science*, v. 8, p. 112–117, jan. 2012. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050912000233>>.
- FLANIGAN, D.; BROUSE, P. Evaluating the Allocation of Border Security System of Systems Requirements. *Procedia Computer Science*, v. 16, p. 631–638, jan. 2013. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050913000677>>.
- FRIEDENTHAL, S.; MOORE, A.; STEINER, R. *A Practical Guide to SysML: The Systems Modeling Language*. [S.l.]: Elsevier, 2011. Google-Books-ID: 8KY2YZliXv0C. ISBN 9780123852069.
- HALLERSTEDDE, S. *et al.* Technical challenges of SoS requirements engineering. In: *2012 7th International Conference on System of Systems Engineering (SoSE)*. [S.l.: s.n.], 2012. p. 573–578.

HEAVEN, W.; FINKELSTEIN, A. UML profile to support requirements engineering with KAOS. *IEE Proceedings - Software*, v. 151, n. 1, p. 10–27, fev. 2004. ISSN 1462-5970.

HEVNER, A. R. *et al.* Design Science in Information Systems Research. *MIS Q.*, v. 28, n. 1, p. 75–105, mar. 2004. ISSN 0276-7783. Disponível em: <<http://dl.acm.org/citation.cfm?id=2017212.2017217>>.

HOLT, J. *et al.* A Model-Based Approach for Requirements Engineering for Systems of Systems. *IEEE Systems Journal*, v. 9, n. 1, p. 252–262, mar. 2015. ISSN 1932-8184.

HORKOFF, J. *et al.* Goal-Oriented Requirements Engineering: A Systematic Literature Map. In: *2016 IEEE 24th International Requirements Engineering Conference (RE)*. [S.l.: s.n.], 2016. p. 106–115.

HULL, E.; JACKSON, K.; DICK, J. *Requirements Engineering*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2004. ISBN 9781852338794.

JOYCE, g. P. F. *et al.* 6.4.2 A Systems Assurance Perspective Towards Generic Systems Engineering. *INCOSE International Symposium*, v. 21, n. 1, p. 747–766, jun. 2011. ISSN 2334-5837. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2011.tb01240.x/abstract>>.

KATINA, P. F.; KEATING, C. B.; JARADAT, R. M. System Requirements Engineering in Complex Situations. *Requir. Eng.*, v. 19, n. 1, p. 45–62, mar. 2014. ISSN 0947-3602. Disponível em: <<http://dx.doi.org/10.1007/s00766-012-0157-0>>.

KENT, S. Model Driven Engineering. In: . London, UK, UK: Springer-Verlag, 2002. (IFM '02), p. 286–298. ISBN 9783540437031. Disponível em: <<http://dl.acm.org/citation.cfm?id=647983.743552>>.

KHLIF, I. *et al.* A Multi-scale Modelling Perspective for SoS Architectures. In: . New York, NY, USA: ACM, 2014. (ECSAW '14), p. 30:1–30:5. ISBN 9781450327787. Disponível em: <<http://doi.acm.org/10.1145/2642803.2642833>>.

KLAMBAUER, T.; HOLL, G.; GRÜNBAACHER, P. Monitoring System-of-Systems Requirements in Multi Product Lines. In: . Springer, Berlin, Heidelberg, 2013. (Lecture Notes in Computer Science), p. 379–385. ISBN 9783642374210 9783642374227. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-37422-7_27>.

KOTEJOSHYER, B.; SINGH, B.; TITTIBHA, S. Rule-based Requirements Management Methodology. *J. Softw. Evol. Process*, v. 26, n. 3, p. 321–328, mar. 2014. ISSN 2047-7473. Disponível em: <<http://dx.doi.org/10.1002/smr.1589>>.

LAHBOUBE, F. *et al.* *Systems of Systems Paradigm in a Hospital Environment: Benefits for Requirements Elicitation Process*. [S.l.: s.n.], 2014. v. 9. DOI: 10.15866/irecos.v9i10.3843.

LAMSWEERDE, A. v. Goal-oriented requirements engineering: a guided tour. In: . [S.l.: s.n.], 2001. p. 249–262.

LANE, J. A. System of systems capability to requirements engineering. In: *2014 9th International Conference on System of Systems Engineering (SOSE)*. [S.l.: s.n.], 2014. p. 91–96.

- LAPOUCHNIAN, A. *Goal-Oriented Requirements Engineering: An Overview of the Current Research*. Toronto, Canada, 2005.
- LEWIS, G. *et al.* Engineering Systems of Systems. In: *2008 2nd Annual IEEE Systems Conference*. [S.l.: s.n.], 2008. p. 1–6.
- LEWIS, G. A. *et al.* Requirements engineering for systems of systems. In: *2009 3rd Annual IEEE Systems Conference*. [S.l.: s.n.], 2009. p. 247–252.
- LISBOA, L. B. *et al.* A systematic review of domain analysis tools. *Information and Software Technology*, v. 52, n. 1, p. 1–13, jan. 2010. ISSN 0950-5849. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584909000834>>.
- LUO, J.; SAHRAOUI, A. E. K.; HESSAMI, A. G. Emergent properties and requirements evolution in engineering systems and a roadmap. In: *2015 Third World Conference on Complex Systems (WCCS)*. [S.l.: s.n.], 2015. p. 1–5.
- MAIA, P. *et al.* On the Development of Systems-of-Systems Based on the Internet of Things: A Systematic Mapping. In: . New York, NY, USA: ACM, 2014. (ECSAW '14), p. 23:1–23:8. ISBN 9781450327787. Disponível em: <<http://doi.acm.org/10.1145/2642803.2642828>>.
- MAIER, M. W. *Architecting Principles for System of Systems*. [S.l.: s.n.], 1998. v. 1. DOI: 10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D.
- MALL, R. *Fundamentals of Software Engineering*. 4th. ed. [S.l.]: Prentice-Hall of India Pvt.Ltd, 2014. ISBN 9788120348981.
- MOKHTARPOUR, B.; STRACENER, J. A Conceptual Methodology for Selecting the Preferred System of Systems. *IEEE Systems Journal*, v. 11, n. 4, p. 1928–1934, dez. 2017. ISSN 1932-8184.
- MONTEIRO, R. *et al.* Model-Driven Development for Requirements Engineering: The Case of Goal-Oriented Approaches. In: *2012 Eighth International Conference on the Quality of Information and Communications Technology*. [S.l.: s.n.], 2012. p. 75–84.
- MONZON, A. Smart cities concept and challenges: Bases for the assessment of smart city projects. In: *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*. [S.l.: s.n.], 2015. p. 1–11.
- NCUBE, C. On the engineering of systems of systems: Key challenges for the Requirements Engineering community. In: *2011 Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems*. [S.l.: s.n.], 2011. p. 70–73.
- NCUBE, C.; LIM, S. L.; DOGAN, H. Identifying top challenges for international research on requirements engineering for systems of systems engineering. In: *2013 21st IEEE International Requirements Engineering Conference (RE)*. [S.l.: s.n.], 2013. p. 342–344.
- PALDES, R. *et al.* A utilização da linguagem natural na especificação de requisitos: um estudo por meio das equações estruturais. In: *19o Workshop em Engenharia de Requisitos*. Quito, Pichincha, Ecuador: [s.n.], 2016. (WER '16).
- PARREIRAS, F. S. *Marrying model-driven engineering and ontology technologies: The TwoUse approach*. Tese (Doutorado) — University of Koblenz-Landau, 2011.

- PENZENSTADLER, B.; ECKHARDT, J. A Requirements Engineering content model for Cyber-Physical Systems. In: *2012 Second IEEE International Workshop on Requirements Engineering for Systems, Services, and Systems-of-Systems (RESS)*. [S.l.: s.n.], 2012. p. 20–29.
- PERRY, D. E.; PORTER, A. A.; VOTTA, L. G. Empirical Studies of Software Engineering: A Roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM, 2000. (ICSE '00), p. 345–355. ISBN 9781581132533. Disponível em: <<http://doi.acm.org/10.1145/336512.336586>>.
- PETRINCA, P.; GAMMALDI, M.; TIRONE, L. 5.5.1 A SysML-based Approach for the Specification of Complex Systems. *INCOSE International Symposium*, v. 22, n. 1, p. 713–786, jul. 2012. ISSN 2334-5837. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2012.tb01370.x/abstract>>.
- POHL, K.; RUPP, C. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant*. [S.l.]: Rocky Nook, 2011. Google-Books-ID: GbsiTwEACAAJ. ISBN 9781933952819.
- RUNESON, P. *et al.* *Case Study Research in Software Engineering: Guidelines and Examples*. 1st. ed. [S.l.]: Wiley Publishing, 2012. ISBN 9781118104354.
- SAFWAT, A.; SENOUSY, M. B. Addressing Challenges of Ultra Large Scale System on Requirements Engineering. *Procedia Computer Science*, v. 65, p. 442–449, jan. 2015. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050915029464>>.
- SANTOS, W. A. D.; LEONOR, B. B. F.; STEPHANY, S. A Knowledge-Based and Model-Driven Requirements Engineering Approach to Conceptual Satellite Design. In: . Springer, Berlin, Heidelberg, 2009. (Lecture Notes in Computer Science), p. 487–500. ISBN 9783642048395 9783642048401. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-04840-1_36>.
- SAVIO, D.; ANITHA, P. C.; IYER, P. P. Considerations for a requirements engineering process model for the development of systems of systems. In: *2011 Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems*. [S.l.: s.n.], 2011. p. 74–76.
- SAYAMA, H. *Introduction to the Modeling and Analysis of Complex Systems*. Print edition. [S.l.]: Open SUNY Textbooks, 2015. ISBN 9781942341086.
- SILVA, E.; BATISTA, T.; OQUENDO, F. A mission-oriented approach for designing system-of-systems. In: . [S.l.: s.n.], 2015. p. 346–351.
- SILVA, E. *et al.* On the Characterization of Missions of Systems-of-Systems. In: . New York, NY, USA: ACM, 2014. (ECSAW '14), p. 26:1–26:8. ISBN 9781450327787. Disponível em: <<http://doi.acm.org/10.1145/2642803.2642829>>.
- SOMMERVILLE, I. *Software Engineering*. 9th. ed. USA: Addison-Wesley Publishing Company, 2010. ISBN 9780137035151.

- SOMMERVILLE, I.; LOCK, R.; STORER, T. Information Requirements for Enterprise Systems. In: . Springer, Berlin, Heidelberg, 2012. (Lecture Notes in Computer Science), p. 266–282. ISBN 9783642340581 9783642340598. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-34059-8_14>.
- STAAB, S. *et al.* Model Driven Engineering with Ontology Technologies. In: . Berlin, Heidelberg: Springer-Verlag, 2010. (ReasoningWeb'10), p. 62–98. ISBN 9783642155420. Disponível em: <<http://dl.acm.org/citation.cfm?id=1886135.1886138>>.
- TECH, A. R. B. Monitoramento de bovino utilizando redes de sensores sem fio. In: *XXI Congresso Brasileiro de Zootecnia - Inovações Tecnológicas e Mercado Consumidor*. Alagoas, Brasil: IEEE, 2012.
- VARGAS, I. G.; GOTTARDI, T.; BRAGA, R. T. V. Approaches for Integration in System of Systems: A Systematic Review. In: . New York, NY, USA: ACM, 2016. (SESoS '16), p. 32–38. ISBN 9781450341721. Disponível em: <<http://doi.acm.org/10.1145/2897829.2897835>>.
- VIERHAUSER, M. A Requirements Monitoring Infrastructure for Systems of Systems. In: . New York, NY, USA: ACM, 2014. (ASE '14), p. 887–890. ISBN 9781450330138. Disponível em: <<http://doi.acm.org/10.1145/2642937.2653468>>.
- VIERHAUSER, M.; RABISER, R.; GRÜNBAACHER, P. A Requirements Monitoring Infrastructure for Very-Large-Scale Software Systems. In: . Springer, Cham, 2014. (Lecture Notes in Computer Science), p. 88–94. ISBN 9783319058429 9783319058436. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-05843-6_7>.
- VIERHAUSER, M. *et al.* A requirements monitoring model for systems of systems. In: *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. [S.l.: s.n.], 2015. p. 96–105.
- VÖLTER, M. *et al.* *Model-Driven Software Development: Technology, Engineering, Management*. [S.l.]: John Wiley & Sons, 2013. Google-Books-ID: 9ww_D9fAKncC. ISBN 9781118725764.
- WAINER, J. Métodos de pesquisa quantitativa e qualitativa para a ciência da computação. In: *T. Kowaltowski K. Breitman (Eds.), Jornada de atualização em informática - csbc*. Rio de Janeiro: SBC PUC-Rio, Brasil: [s.n.], 2007. p. 221 – 262.
- WALKER, R. G.; KEATING, C. *Defining SoS requirements: An early glimpse at a methodology*. [S.l.: s.n.], 2012. v. 3. DOI: 10.1504/IJSSE.2012.052685.
- WEILKIENS, T. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. [S.l.]: Elsevier, 2011. Google-Books-ID: LKC9F5gQt3AC. ISBN 9780080558318.
- WIERINGA, R. J. *Design Science Methodology for Information Systems and Software Engineering*. [S.l.: s.n.], 2014. DOI: 10.1007/978-3-662-43839-8.
- WOHLIN, C. *et al.* *Experimentation in Software Engineering*. Berlin Heidelberg: Springer-Verlag, 2012. ISBN 9783642290435. Disponível em: <<http://www.springer.com/in/book/9783642290435>>.
- WOHLIN, C.; RUNESON, P.; ST, M. H. *Experimentation in Software Engineering*. Edição: 2012. New York: Springer, 2012. ISBN 9783642290435.

- WOODCOCK, J. *et al.* *The COMPASS Modelling Language: Timed Semantics in UTP*. [S.l.]: Communicating Process Architectures 2014, 2014.
- WOODCOCK, J.; CAVALCANTI, A. The Semantics of Circus. In: *ZB 2002: Formal Specification and Development in Z and B*. Springer, Berlin, Heidelberg, 2002. p. 184–203. Disponível em: <https://link.springer.com/chapter/10.1007/3-540-45648-1_10>.
- WOODCOCK, J. *et al.* Features of CML: A formal modelling language for Systems of Systems. In: *2012 7th International Conference on System of Systems Engineering (SoSE)*. [S.l.: s.n.], 2012. p. 1–6.
- YANG, K. w. *et al.* The Research of System of Systems Requirement Modeling and Toolkits. In: *2009 11th International Conference on Computer Modelling and Simulation*. [S.l.: s.n.], 2009. p. 107–110.
- YANG-TURNER, F.; LAU, L. A pragmatic strategy for creative requirements elicitation: From current work practice to future work practice. In: *2011 Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems*. [S.l.: s.n.], 2011. p. 28–31.
- ZINK, T. Strategies for planning complex systems development. In: *PMI® Global Congress*. [S.l.: s.n.], 2015. (PMI '15).
- ZOPPI, T.; CECCARELLI, A.; BONDAVALLI, A. Exploring Anomaly Detection in Systems of Systems. In: . New York, NY, USA: ACM, 2017. (SAC '17), p. 1139–1146. ISBN 9781450344869. Disponível em: <<http://doi.acm.org/10.1145/3019612.3019765>>.
- ZOWGHI, D.; COULIN, C. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In: . Springer, Berlin, Heidelberg, 2005. p. 19–46. ISBN 9783540250432 9783540282440. DOI: 10.1007/3-540-28244-0_2. Disponível em: <https://link.springer.com/chapter/10.1007/3-540-28244-0_2>.

Apêndices

APÊNDICE A – Questionário Aplicado

Questionário para obtenção das funcionalidades de um sistema de controle e monitoramento da produção animal.

Prezado pesquisador ou profissional da área de Zootecnia,

Estamos executando uma pesquisa de opinião (survey) com o objetivo de identificar as principais funcionalidades de um sistema capaz de controlar e monitorar a produção animal de forma automatizada. Esse sistema teria como objetivo diminuir os custos da produção, além de aumentar a qualidade de vida dos produtores rurais.

Sua participação ocorrerá por meio das respostas informadas neste questionário, elaborado pelo estudante de mestrado Felipe Lima Duarte. Garantimos o mais rigoroso sigilo das informações aqui declaradas, através da omissão total de quaisquer informações que permitam sua identificação.

Qualquer orientação ou dúvida sobre o questionário deve ser enviado para o discente responsável pela pesquisa através do seguinte email: felipe.duarte@ufersa.edu.br.

O resultado da pesquisa será utilizado como insumo para a dissertação de mestrado do referido discente, podendo também ser apresentado em encontros ou em revistas científicas.

Perfil Profissional

- 1. Informe o seu nível de escolaridade:

- Bacharel
- Especialista
- Mestre
- Doutor
- Outro

- 2. Faixa Etária:

- 17 a 24 Anos
- 25 a 35 Anos
- 36 a 50 Anos
- A partir de 51 Anos

- 3. Qual a sua experiência com projetos para informatização de processos de produção animal?
 - Profissional
 - Acadêmica
 - Outra

- 4. Qual o seu grau de conhecimento sobre a Zootecnia de Precisão?
 - Muito Baixo
 - Baixo
 - Médio
 - Alto
 - Muito Alto

Viabilidade da Pesquisa

- 5. Na sua opinião, qual a importância da Zootecnia de Precisão na produção animal ?
 - Muito Baixa
 - Baixa
 - Médio
 - Alta
 - Muito Alta

- 6. Dado o seu conhecimento, há projetos relacionados a Zootecnia de Precisão na UFERSA?
 - Sim
 - Não
 - Outra:

- 7. Na sua opinião, qual seria a viabilidade de se implementar projetos relacionados a Zootecnia de Precisão na UFERSA?
 - Muito Baixa
 - Baixa

- Média
- Alta
- Muito Alta

Funcionalidades do Sistema

- 8. Na criação e produção bovina, quais desses aspectos merecem maior atenção por parte dos produtores?
 - Localização exata do Animal
 - Exposição à radiação
 - Exposição à temperatura
 - Exposição à umidade
 - Temperatura corporal
 - Local com ventilação
 - Higienização do ambiente
 - Outro:

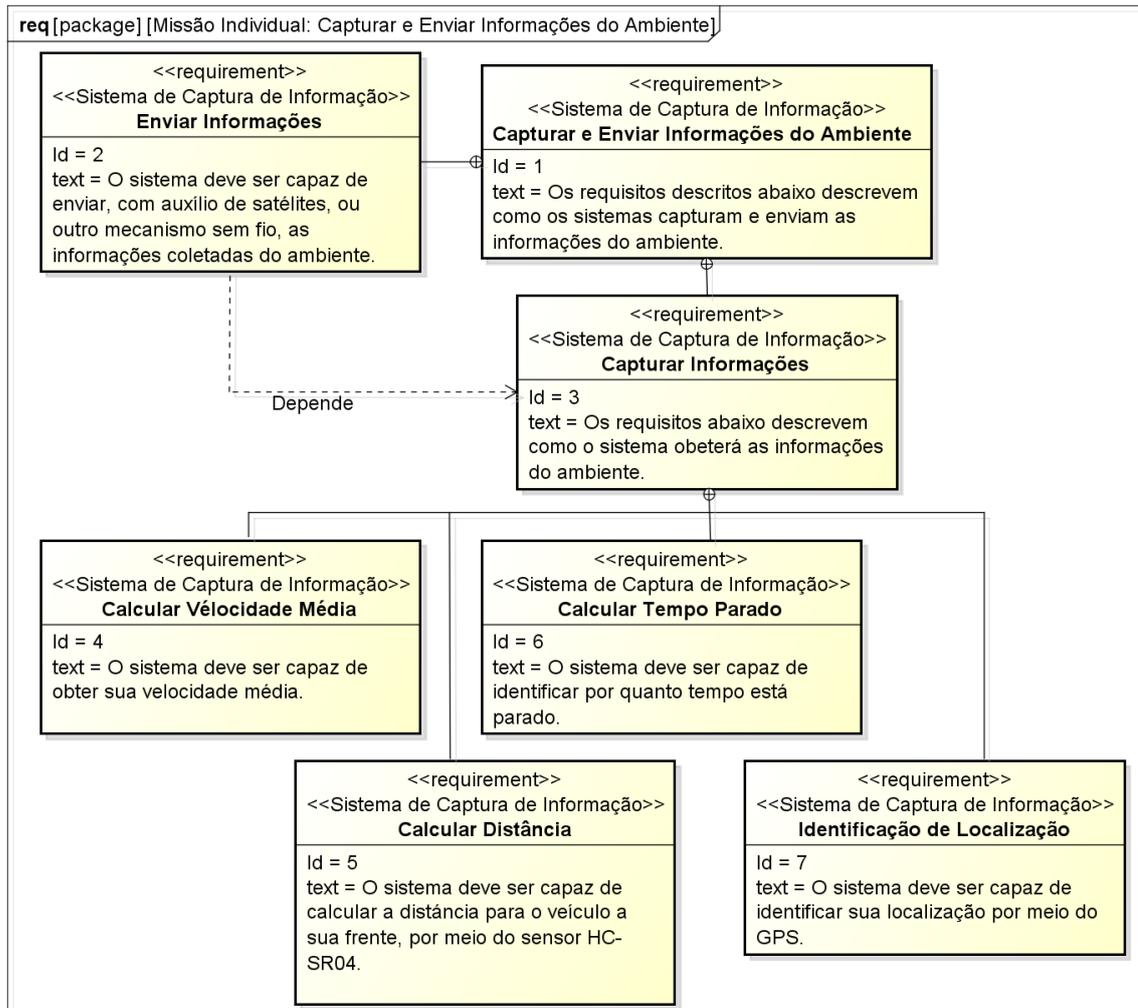
- 9. Dado um sistema para controle e monitoramento de produção animal, quais dessas funcionalidades são estritamente necessários para o sistema?
 - O sistema deve permitir a consulta da localização exata dos animais.
 - A localização deve ser obtida por meio de drones que sobrevoam a área.
 - A localização deve ser obtida por meio de sensores que monitoram o animal.
 - o sistema deve ser capaz de obter o tempo de pastejo, ruminção e descanso dos animais.
 - O sistema deve permitir a identificação do Abigeato.
 - O sistema deve permitir a consulta da temperatura corporal dos animais.
 - O sistema deve permitir o controle e monitoramento da temperatura ambiental.
 - O sistema deve permitir o controle e monitoramento da umidade relativa do ar.
 - O sistema deve informar dados sobre a exposição a radiação sofrida pelos animais.

- 10. Quais outras funcionalidades não listadas na pergunta anterior seriam importantes para esse sistema?

APÊNDICE B – Modelos do Sistema

SisBovi

Figura 34 – Requisitos da missão capturar e enviar informações do ambiente (completo)



Fonte: Autoria Própria

Figura 35 – Requisitos da missão calcular anomalia (completo)

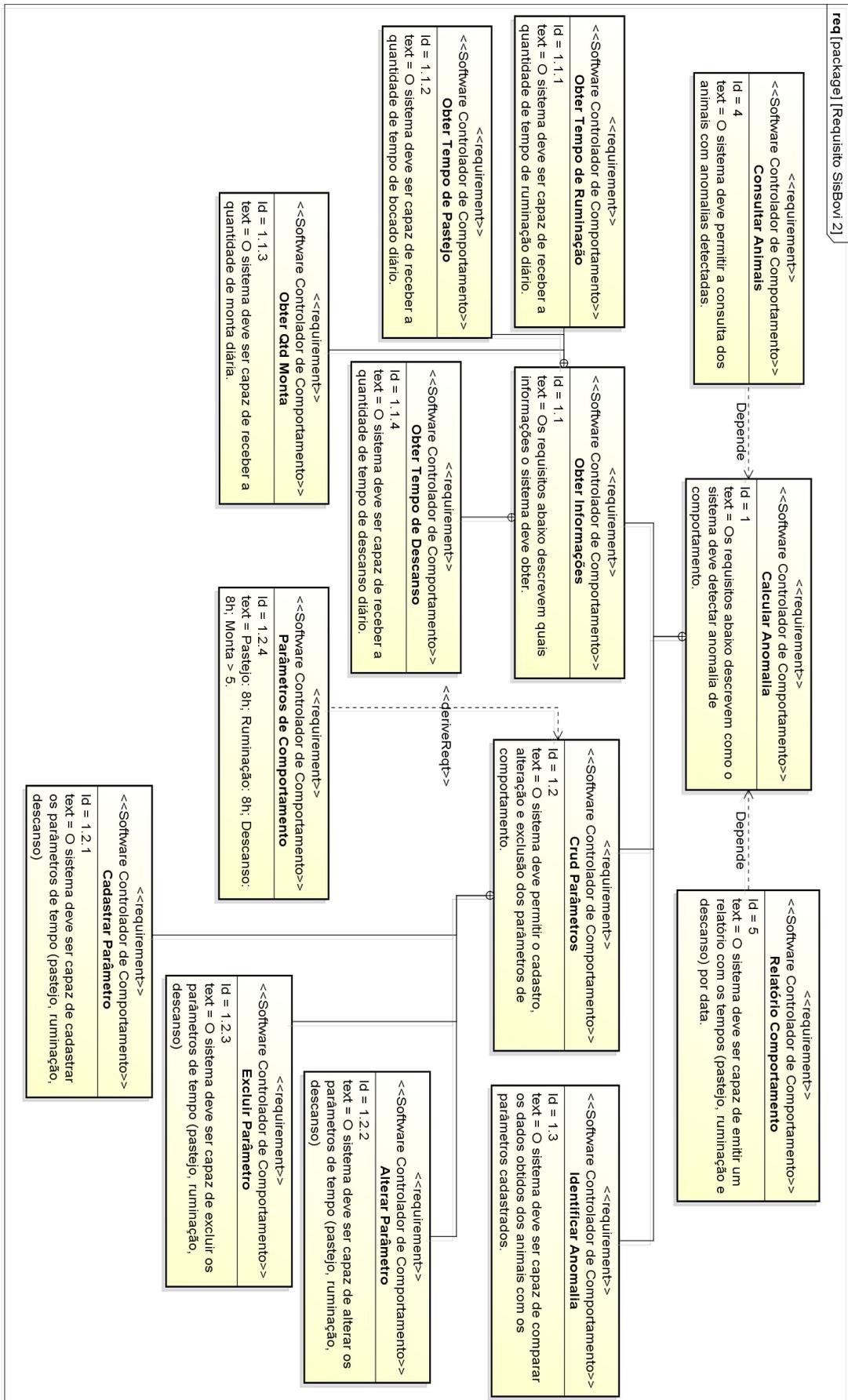


Figura 36 – Requisitos da missão identificar saída do perímetro (completo)

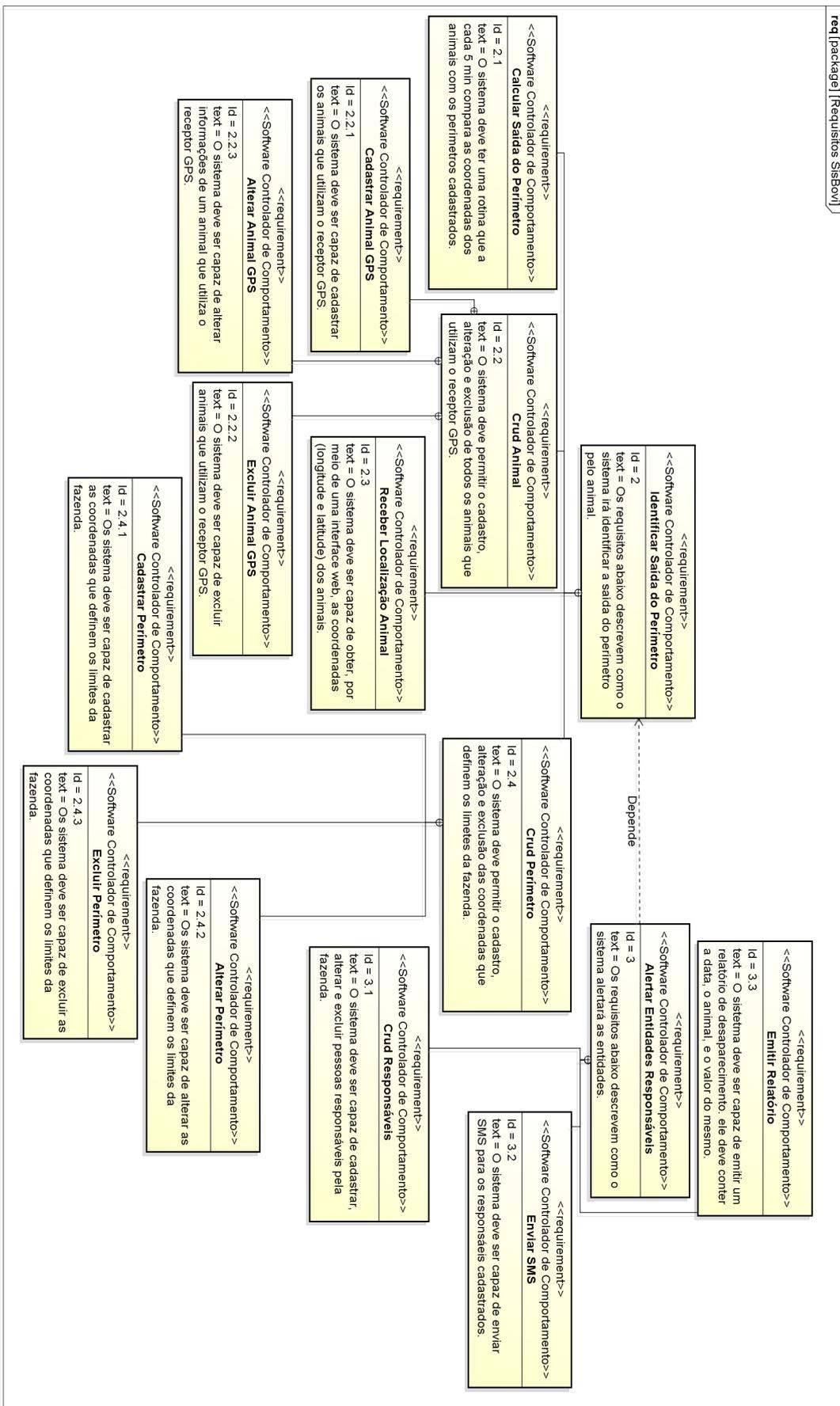
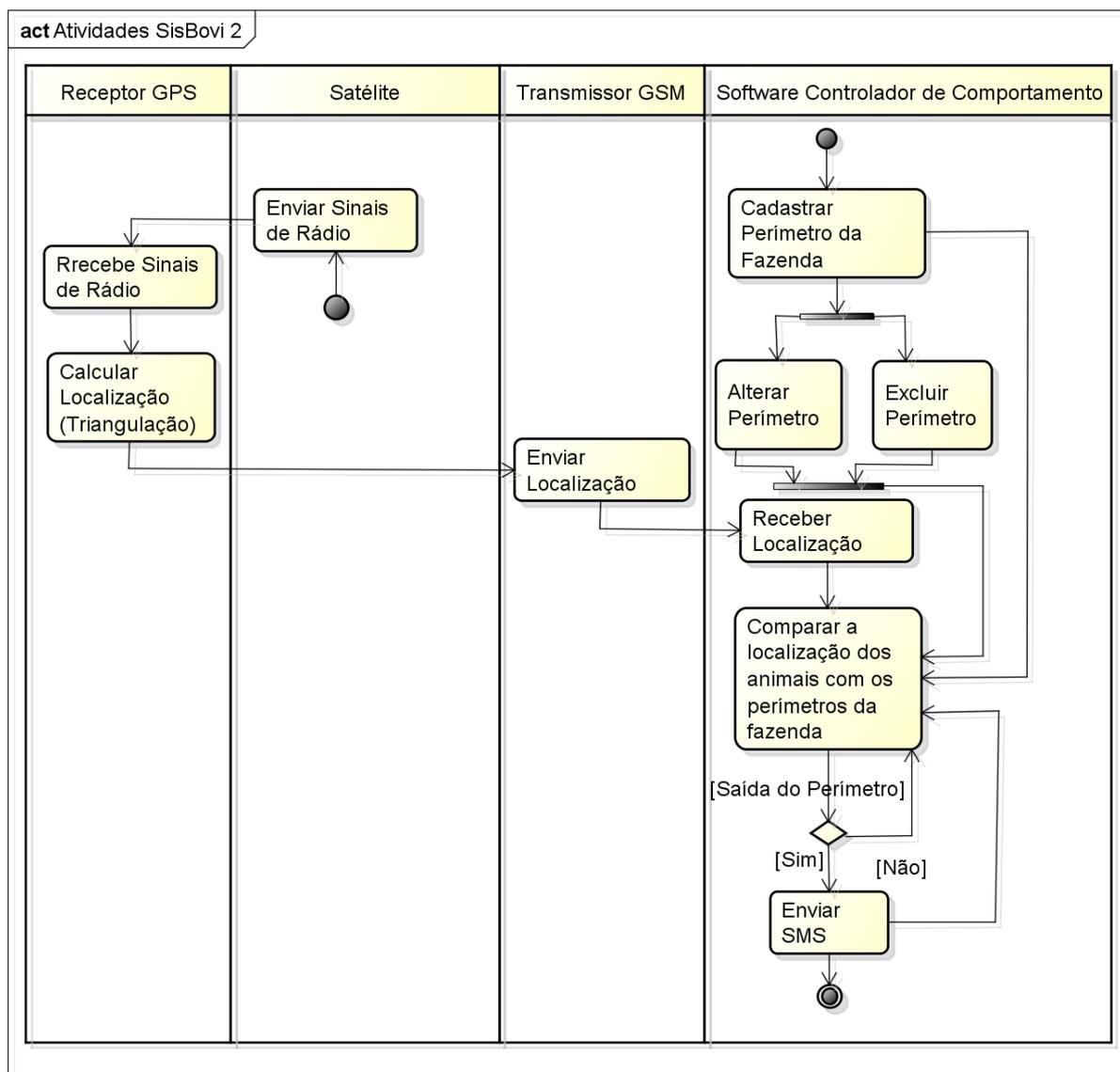


Figura 37 – Atividades do fluxo detectar saída do perímetro



Fonte: Autoria Própria