

**MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO  
E EM ENGENHARIA DE DEFESA**

**EQUIPE IME PROJETO S2C2**

**ESPECIFICAÇÃO DE PROCEDIMENTOS SISTEMÁTICOS PARA APOIAR A  
INTEROPERABILIDADE PARA SOS DE  $C^2$  (NÍVEL CONCEITUAL)**

**RIO DE JANEIRO  
JULHO/2025**

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>2</b>
<b>2</b>	<b>PROCEDIMENTO SISTEMÁTICO PARA DESENVOLVIMENTO DE ONTOLOGIAS . . . . .</b>	<b>4</b>
2.1	A METODOLOGIA NEON . . . . .	6
2.2	A ABORDAGEM SABIO . . . . .	8
<b>3</b>	<b>PROCEDIMENTO SISTEMÁTICO PARA CONCEPÇÃO DE SOS .</b>	<b>11</b>
3.1	LEVANTAMENTO DOS TRABALHOS RELACIONADOS . . . . .	11
3.2	PROCESSO PARA MODELAGEM DE UM SOSC2 . . . . .	15
3.2.1	LEVANTAMENTO DE REQUISITOS DO SOS . . . . .	15
3.2.2	PROJETO SOS . . . . .	18
3.2.3	CONSIDERAÇÕES ADICIONAIS . . . . .	21
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>23</b>
4.1	CONSTRUÇÃO DE ONTOLOGIAS DE REFERÊNCIA . . . . .	23
4.2	APOIO À CONSTRUÇÃO DE SOSC2 . . . . .	25
4.3	CONSIDERAÇÕES FINAIS . . . . .	26
	<b>REFERÊNCIAS . . . . .</b>	<b>28</b>

# 1 INTRODUÇÃO

Os trabalhos desenvolvidos no contexto do Projeto **Sistema de Sistemas de Comando e Controle (S2C2)** responderam às questões de pesquisa levantadas nos relatórios anteriores (RELATÓRIOS DE LEVANTAMENTO DE ABORDAGENS EXISTENTES PARA INTEROPERABILIDADE (NÍVEL CONCEITUAL E LÓGICO) DE SOS DE C2). A seguir listamos as questões de pesquisa mencionadas:

- É possível construir uma ontologia de referência que possa servir de base para a representação e intercâmbio de dados nos SoSC2, facilitando a interoperabilidade entre estes?
- É possível construir uma ontologia de referência que possa servir de base para a representação de redes de comunicação heterogêneas de modo a manter os sistemas constituintes de um SoSC2 bem conectados e com isso assegurar o intercâmbio de dados e a interoperabilidade entre eles?

Além dessas questões de pesquisa, ao longo do projeto, surgiu uma outra questão:

- Como apoiar o desenvolvimento de um SoS para aumentar a interoperabilidade de sistemas de C2?

Para responder tais questões, os trabalhos realizados utilizaram e propuseram procedimentos sistemáticos, aplicando-os ao contexto do presente projeto.

Este relatório é um indicador físico de execução da atividade de *Especificação da Sistemática<sup>1</sup> para o nível **conceitual***, prevista no Plano de Trabalho do projeto S2C2, para atingir a Meta física 5: *Interop - Conceber linhas de ação para a interoperabilidade de dados com relação à modelagem nos níveis conceitual e lógico*.

Além da introdução, o presente relatório é estruturado como se segue. O capítulo 2 apresenta o procedimento sistemático adotado para a concepção das ontologias desenvolvidas no contexto deste projeto. A seguir, o capítulo 3 apresenta a proposta de um metaprocesso para apoiar o desenvolvimento de um SoS a partir de sistemas constituintes já existentes. Por fim, o capítulo 4 apresenta uma descrição resumida dos resultados obtidos até então com a aplicação dos procedimentos sistemáticos apresentados. Ao final desse último capítulo apresentamos uma análise sobre os resultados alcançados até então tanto

---

<sup>1</sup> Termo originalmente usado para o planejamento de atividades do projeto, mas que foi ajustado para refletir melhor o significado pretendido. Assim, pelo restante do documento passaremos a nos referir a *Procedimento Sistemático*.

em termos de artefatos e publicações (35 ao todo) quanto com relação à formação de recursos humanos (13 ao todo).

Gostaríamos de destacar que os resultados obtidos ao longo da execução deste projeto representam um retorno significativo para o investimento realizado pelo órgão de fomento, contribuindo de forma efetiva para o avanço do conhecimento na área e para a formação de recursos humanos qualificados.

## 2 PROCEDIMENTO SISTEMÁTICO PARA DESENVOLVIMENTO DE ONTOLOGIAS

Ruy *et al.* (1) definem ontologia como uma representação formal de uma conceitualização compartilhada de um universo de discurso. Caracterizadas principalmente por uma taxonomia e um conjunto de regras de inferência, elas podem ser usadas em um aplicativo específico para delinear possíveis relacionamentos e restrições no uso desses conceitos. De acordo com os mesmos autores, a literatura acadêmica classifica as ontologias usando diversas perspectivas, incluindo níveis de generalidade. Nesse sentido, as ontologias podem ser classificadas em domínio, núcleo e fundacional (ou de fundamentação). Enquanto as ontologias de domínio descrevem conceitos de um domínio específico (nível de generalidade mais baixo), como a ontologia de sensores semânticos (2) ou a ontologia de redes heterogêneas (3), as ontologias fundacionais abrangem muitos campos, fornecendo conceitos gerais e transformando-se em uma ontologia independente de domínio, como DOLCE (4) e UFO (5). Por fim, as ontologias de núcleo ficam entre as ontologias de fundamentação e de domínio, fornecendo conhecimento estrutural em um campo específico que pode abranger diferentes áreas de aplicação no mesmo campo. Portanto, as ontologias de nível superior podem ser usadas para apoiar o desenvolvimento de ontologias de baixo nível.

As ontologias têm sido cada vez mais aplicadas em várias áreas a fim de se criar representações fidedignas com o objetivo de interoperar e facilitar as análises de domínios, ajudando na resolução de problemas, na organização do conhecimento e na tomada de decisões. Na área de desenvolvimento de sistemas, a representação da realidade acerca de determinado domínio ou sistema implica diretamente na capacidade de transferência de informações e de interoperabilidade com outros sistemas. Dentro desse contexto, a modelagem conceitual tem sido usada como parte de metodologias de desenvolvimento de sistemas e bancos de dados para representar porções do mundo real dentro de um determinado domínio ou sistema, representando o estado atual de compreensão e fornecendo uma base sólida para testar e melhorar nossa compreensão (6). A prática da modelagem conceitual baseada em ontologia é denominada como *Ontology-driven conceptual modeling* (ODCM) (7).

Uma linguagem de modelagem conceitual deve possuir a capacidade de fornecer um conjunto de primitivas de modelagem que possam expressar diretamente conceitos relevantes do domínio, compreendendo o que é chamado de conceitualização de domínio (8). Ao se criar sistemas que irão compartilhar informações ou interoperar modelos, os conceitos acerca do domínio precisam ser bem definidos. Em (9), Guizzardi argumenta que a fidelidade à realidade de um determinado sistema depende muito da disponibilidade de linguagens de modelagem conceitual bem fundamentadas. Dessa forma, o exercício da modelagem

conceitual também se torna um desafio quando se propõe a realizar uma representação fidedigna do mundo real. Dentre as linguagens conhecidas para a realização dessa tarefa destacam-se a UML (Unified Modeling Language) (10), OCL (Object Constraint Language) (11), ERM (Entity-relationship model) (12), dentre outras.

Não obstante, essas linguagens foram desenvolvidas de forma a atender requisitos de desenvolvimento de sistemas, sem se ater aos princípios de representação da realidade já estudados em outras áreas do conhecimento. Dessa forma, essas linguagens trouxeram certa limitação na capacidade de representação e problemas de falta de expressividade, incompletude de linguagem, incompletude ontológica, interoperabilidade semântica e de ambiguidade, por utilizar um pensamento mecanicista que usa como base as experiências dos desenvolvedores de sistemas.

Por sua vez, um modelo conceitual bem fundamentado, propõe justamente o contrário, utilizando como base conceitos de representação da realidade bem fundamentados advindos da ontologia, aumentando a expressividade semântica nas representações, tornando possível a criação de ontologias de referência por meio da modelagem conceitual. Em (13) os autores realizam uma comparação entre modelos conceituais tradicionais e modelos conceituais dirigidos à ontologia, demonstrando que podem ser criados modelos com maior qualidade usando ODCM do que modelos conceituais tradicionais. Um exemplo de linguagem de modelagem conceitual voltada para ODCM é a OntoUML (14), que é baseada na ontologia de fundamentação denominada *Unified Foundational Ontology* (UFO) (9), que utiliza conceitos formais da ontologia para a criação de modelos conceituais bem fundamentados.

Independentemente do tipo da ontologia, deve-se seguir um processo de desenvolvimento conhecido como engenharia de ontologia. Entre as metodologias disponíveis para orientar o desenvolvimento de uma ontologia, estão a **NeOn** (15) e a **SABiO** (16). A primeira é frequentemente referenciada na academia, inspirando outras metodologias de desenvolvimento. A segunda materializa os conceitos do ODCM ao distinguir claramente dois artefatos: a ontologia de referência, apoiada por uma ontologia de fundamentação, e uma ontologia operacional que as máquinas podem interpretar. Ambas serão descritas mais detalhadamente a seguir.

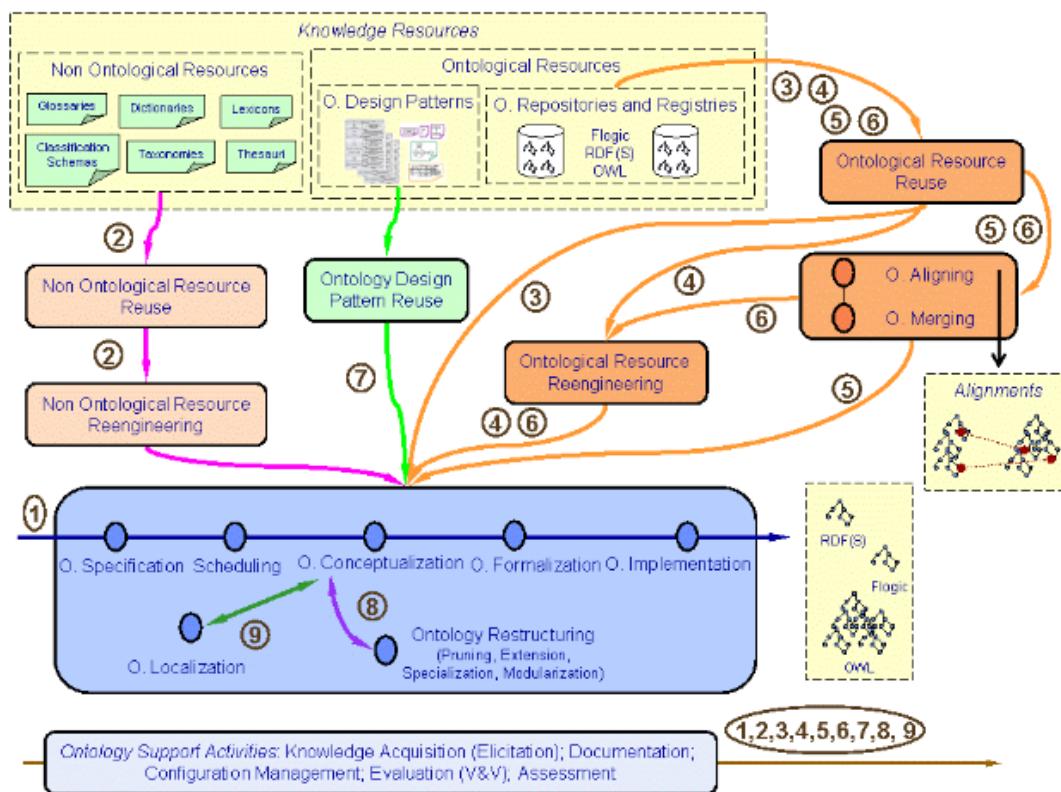
É importante salientar que a construção de ontologias pode ser inspirada e apoiada pelas duas metodologias, utilizando seus aspectos mais interessantes. Por exemplo, enquanto a SABiO propõe um abordagem de construção mais pragmática, suas fases iniciais podem ser enriquecidas pelos processos de levantamento de recursos ontológicos e não-ontológicos e de definição de requisitos sugeridos pela NeOn.

## 2.1 A Metodologia NeOn

A metodologia NeOn, proposta por Gómez-Pérez *et. al* (17, 15), tem como objetivo fornecer uma estrutura de desenvolvimento para sistemas baseados em ontologias como resposta à crescente complexidade da Web Semântica, representada pelo grande número de ontologias colaborativas construídas por equipes distribuídas que se encontram inseridas em redes de ontologias. Estas redes podem incluir ontologias existentes, mapeamentos de ontologias e ontologias em contínua evolução, ou ontologias cujo desenvolvimento se baseou na reutilização de recursos ontológicos e/ou não ontológicos disponíveis, tais como glossários, léxicos, esquemas de classificação e dicionário de sinônimos, entre outros.

O desenvolvimento da metodologia foi orientado por quatro princípios: generalidade do âmbito, independência tecnológica, definição precisa de processos e atividades, e fácil assimilação por programadores de software e profissionais de ontologias. Neste sentido, a metodologia engloba nove cenários diferentes como resultado da utilização de uma abordagem de dividir para conquistar, decompondo o problema geral do desenvolvimento de ontologias em diferentes subproblemas a resolver. A Figura 1 mostra cada cenário decomposto em diferentes processos e atividades definidos no Glossário de Processos e Atividades da NeoOn (18). Normalmente, estes cenários são combinados para resolver uma necessidade específica de desenvolvimento de ontologias.

Figura 1 – Scenarios for building ontology networks



O cenário *Da Especificação à Implementação* é o primeiro e mais importante cenário

de desenvolvimento, uma vez que é a espinha dorsal da construção de ontologias utilizando a metodologia. Neste cenário, os desenvolvedores constroem a ontologia pretendida a partir do zero, sem reutilizar os recursos de conhecimento existentes. Primeiro, empregam simultaneamente duas atividades: a aquisição de conhecimento, que deve durar todo o desenvolvimento, e a especificação dos requisitos da ontologia, onde os desenvolvedores devem especificar os requisitos da ontologia. O resultado desta última atividade é um documento designado por documento de especificação dos requisitos da ontologia (ORSD em inglês), onde devem ser descritos os seguintes itens: o objetivo, o âmbito e a linguagem de implementação da rede de ontologias, o grupo-alvo, as utilizações pretendidas da rede de ontologias e o conjunto de requisitos que a rede de ontologias deve cumprir sob a forma de questões de competência (CQ). Uma vez concluída esta fase, recomenda-se que sejam investigados os termos que aparecem no ORSD nos recursos de conhecimento disponíveis. O resultado da investigação permite aos desenvolvedores saber quais os recursos podem ser eleitos para reutilização. Os autores sugerem que, após a execução destas atividades, os desenvolvedores realizem as atividades de conceitualização, formalização e implementação recorrendo a outros *frameworks* bem conhecidos como METHONTOLOGY(19) ou On-To-Knowledge(20).

O segundo cenário (*Reutilização e Reengenharia de Recursos Não-Ontológicos*) prescreve a realização do processo de reutilização de recursos não ontológicos (NOR em inglês). Neste caso, os desenvolvedores devem decidir, de acordo com os requisitos do ORSD, quais as NORs que podem ser reutilizadas na ontologia que está a ser desenvolvida. Neste caso, um processo de reengenharia deve transformar as NORs selecionadas em ontologias. O cenário *Reutilização de Recursos Ontológicos* é o terceiro cenário proposto. Neste caso, os desenvolvedores reutilizam recursos ontológicos existentes para construir redes de ontologias. Estes recursos podem ser utilizados na totalidade, ou apenas um subconjunto dos mesmos, ou apenas as suas declarações ontológicas, dependendo da frequência com que os termos recolhidos no ORSD aparecem. Por exemplo, se os recursos necessários tiverem de ser implementados noutra linguagem, os desenvolvedores podem fazer a reengenharia seguindo o quarto cenário (*Reutilização e Reengenharia de Recursos Ontológicos*). Neste caso, o processo de reengenharia pode envolver a engenharia inversa de recursos ontológicos, a reestruturação de recursos e/ou a engenharia avançada de recursos. Estas atividades podem ocorrer a diferentes níveis (especificação, conceitualização, formalização e/ou implementação) e os seus resultados devem ser integrados na atividade correspondente do primeiro cenário.

O quinto cenário (*Reutilização e Fusão de Recursos Ontológicos*) trata da reutilização e fusão de recursos ontológicos existentes pelos desenvolvedores para desenvolver a ontologia prevista. Este cenário surge quando os recursos ontológicos no mesmo domínio são selecionados para reutilização, na esperança de criar um novo recurso ontológico a partir de duas ou mais ontologias, enquanto os recursos ontológicos de origem possivelmente

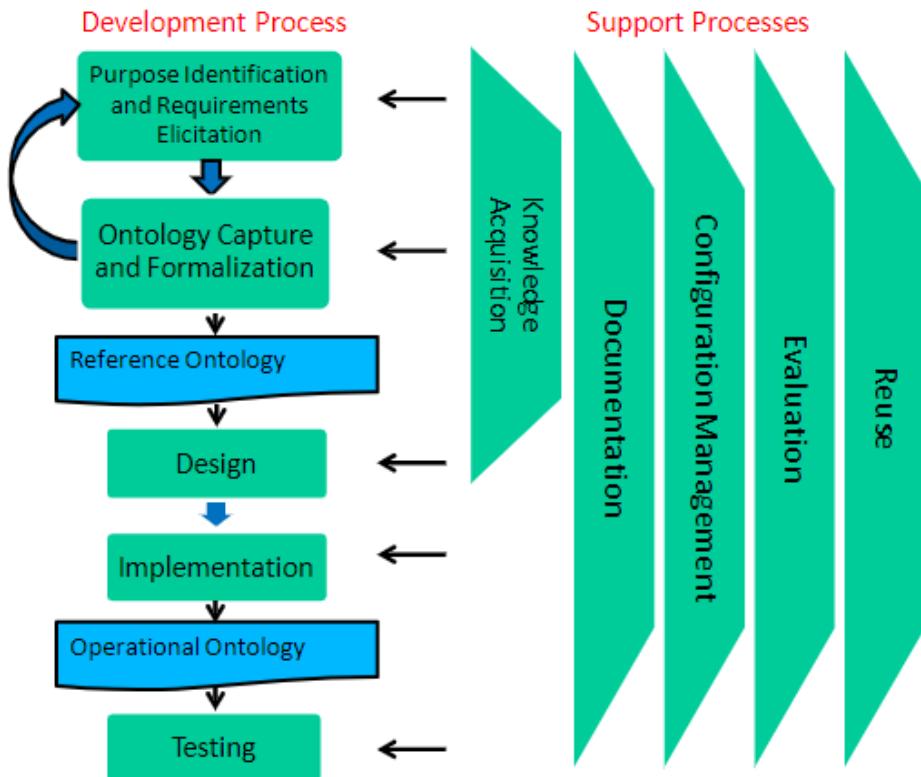
se sobrepõem. No cenário *Reutilização, Fusão e Reengenharia de Recursos Ontológicos*, os desenvolvedores seguem a mesma sequência de atividades do quinto cenário. No entanto, podem decidir fazer a reengenharia dos recursos ontológicos em vez de os fundir tal como estão. O sétimo cenário (*Reutilização de Padrões de Concepção de Ontologias*) trata da reutilização de padrões de concepção de ontologias para reduzir as complexidades de modelagem, com o objetivo de acelerar o processo ou de verificar a adequação das decisões de modelagem. O cenário *Reestruturação de Recursos Ontológicos* é o oitavo cenário descrito na NeOn. Neste cenário, os desenvolvedores devem reestruturar os recursos ontológicos para os integrar na rede de ontologias construída. Isto pode ser feito através da modularização da ontologia em diferentes módulos, podando ramos de ontologia desnecessários, estendendo a ontologia base com novos conceitos e relações, ou especializando os ramos que requerem mais granularidade e incluindo conceitos e relações de domínio mais especializados. Finalmente, o nono cenário (*Localização de Recursos Ontológicos*) trata da localização de ontologias. Os desenvolvedores devem adaptar uma ontologia existente a uma ou várias línguas e comunidades culturais para obter uma ontologia multilíngue.

## 2.2 A Abordagem SABiO

A Abordagem Sistemática para Construção de Ontologias (Systematic Approach for Building Ontologies, em inglês)(SABiO) é uma estrutura de engenharia de ontologias proposta por Ricardo Falbo (21). Fortemente influenciado pela metodologia NeOn, Falbo propõe uma estrutura de desenvolvimento de ontologias mais concisa com cinco fases: *Identificação de Propósito e Elicitação de Requisitos, Captura e Formalização, Projeto, Implementação e Teste*. Cada fase é composta por várias atividades suportadas por vários processos, como apresentado na Figura 2. Uma vez que Falbo descreve adequadamente as atividades das fases e o processo de apoio no seu trabalho, concentrarmo-nos apenas em descrever brevemente as fases de desenvolvimento.

A característica distinta da abordagem é a diferenciação entre ontologias de referência e operacional de um domínio. Uma ontologia de referência do domínio (ou simplesmente **ontologia de referência**) é definida como um modelo conceitual específico do domínio. Em contrapartida, uma ontologia operacional do domínio (ou simplesmente **ontologia operacional**) é uma versão implementável em linguagem de máquina da ontologia de referência. Em termos gerais, enquanto a ontologia de referência é o foco das duas primeiras fases de desenvolvimento, a fase de *Projeto* transforma-a numa ontologia operacional. Finalmente, a implementação e o teste da ontologia operacional é o foco das duas últimas fases. É importante notar que o autor sugere um desenvolvimento incremental e iterativo em vez de uma abordagem sequencial que uma apresentação passo-a-passo das fases de desenvolvimento pode sugerir.

Figura 2 – SABiO's processes



Na fase *Identificação do Propósito e Elicitação de Requisitos*, o desenvolvedor de ontologias deve começar por identificar o objetivo da construção da ontologia e onde esta será utilizada. Logo em seguida, começa a eliciar os seus requisitos. Estes requisitos podem ser classificados em requisitos funcionais e não funcionais. Os requisitos não funcionais referem-se a aspectos gerais não relacionados com o conteúdo da ontologia, tais como o desempenho e a disponibilidade de raciocínio semântico, a capacidade de manutenção e os requisitos de implementação, entre outros. Por outro lado, os requisitos funcionais referem-se ao conteúdo a ser representado pela ontologia. Durante o desenvolvimento de uma ontologia, a utilização das questões de competência é uma das ferramentas utilizadas para gerar os requisitos funcionais. Gruninger (22) define as questões de competência (competence questions em inglês - CQs) como perguntas que a ontologia deve ser capaz de responder, determinando o que é relevante para a ontologia ou, em outras palavras, determinando o seu escopo. Existem diferentes estratégias para gerar tais questões. O desenvolvedor da ontologia pode começar a decompor questões complexas em questões mais simples (abordagem *top-down*) ou, de forma inversa, criar questões simples que serão posteriormente compostas por questões complexas (abordagem *bottom-up*). Além disso, o desenvolvedor pode começar a escrever questões relevantes para o domínio que serão decompostas ou compostas mais tarde em questões simples ou abstratas (abordagem *middle-out*).

Em seguida, na fase *Captura e Formalização*, o principal objetivo é capturar a

conceitualização do domínio por meio das questões de competência. É apoiado pelo processo de aquisição de conhecimento, onde são consultados especialistas do domínio e fontes de conhecimento consolidados (e.g., livros, artigos, normas internacionais, modelos de referência, entre outros). Em seguida, inicia-se a atividade de modelagem conceitual, onde são identificados conceitos e relações no domínio, combinando as questões de competência com o conhecimento adquirido de especialistas do domínio e outras fontes relacionadas. Após organizar os conceitos do domínio em taxonomias, o desenvolvedor está pronto para construir a ontologia de referência. Neste ponto, Falbo sugere que uma ontologia de fundamentação deve apoiar a ontologia de referência, fornecendo uma sólida identificação e organização dos elementos do domínio. Nesse sentido, ele sugere o uso da OntoUML, um perfil do diagrama de classes da UML que incorpora as micro-teorias ontológicas da Unified Foundational Ontology (UFO), uma ontologia de fundamentação proposta por Giancarlo Guizzardi(5). Assim, uma ontologia de referência é entregue no final desta fase.

Em alguns casos, a entrega da ontologia de referência é suficiente e a abordagem não deve mais ser seguida. No entanto, há casos em que uma ontologia operacional também deve ser entregue com base na ontologia de referência. Assim, o desenvolvedor entregará a ontologia operacional na fase de *Projeto* da SABiO. Ao contrário da ontologia de referência, a ontologia operacional é concebida e implementada numa linguagem de máquina, concentrando-se em garantir propriedades computacionais desejáveis. Inicialmente, o desenvolvedor tem de complementar a lista de requisitos tecnológicos não funcionais para a ontologia operacional e depois definir o ambiente em que esta será implementada. Uma vez definido o ambiente, é necessário rever a modularização da ontologia elaborada no início do projeto e definir a arquitetura da ontologia com base nos requisitos tecnológicos não funcionais e no ambiente construído.

Nas últimas fases (*Implementação* e *Teste*), a ontologia operacional desenvolvida é implementada na linguagem operacional escolhida e testada em relação a vários casos. Mais uma vez, as questões de competência desempenham um papel importante na criação global da ontologia. Embora sejam essenciais para a coleta de conceitos e relações do domínio na fase *Captura e Formalização*, aqui são a espinha dorsal dos testes da ontologia. De acordo com Falbo, um caso de teste compreende uma consulta no ambiente de implementação, uma instanciação de dados de acordo com a ontologia sendo testada (entrada) e um resultado esperado (saída). Neste cenário, uma consulta é a materialização de uma questão de competência. O desenvolvedor da ontologia deve executar os casos de teste em diferentes fragmentos, tais como sub-ontologias e partes integradas, até chegar ao teste da ontologia completa.

### 3 PROCEDIMENTO SISTEMÁTICO PARA CONCEPÇÃO DE SOS

Quanto à questão de pesquisa “*Como apoiar o desenvolvimento de um SoS para aumentar a interoperabilidade de sistemas de C2?*”, foi feito um levantamento na literatura, descrito na Seção 3.1, em que foram selecionadas abordagens existentes para a concepção de SoS. Algumas abordagens apontam para a necessidade de especificar os processos, mas não chegam a especificar os dados de entrada e saída das atividades do processo, e consequentemente, não especificam o mapeamento sintático e semântico de dados, nem os protocolos de comunicação entre os sistemas. No sentido de preencher essa lacuna, este trabalho apresenta uma proposta de processo de modelagem de SoSC2 que visa contemplar uma especificação mais detalhada, a partir da qual seja possível gerar módulos integradores entre os sistemas formando um SoS.

Cabe enfatizar que o processo de modelagem de um SoS de C2 depende da colaboração de especialistas militares e/ou responsáveis pelo gerenciamento dos sistemas de C2, em especial da Família de Aplicativos de Comando e Controle da Força Terrestre (FAC2FTer). A interação com tais especialistas visa identificar os processos e sistemas envolvidos nos cenários de operação de C2. Na literatura militar existem muitos documentos e doutrinas que detalham processos e operações de C2, mas a leitura de tais manuais não é suficiente para o entendimento desses processos, e consequentemente, para a modelagem de um SoS de C2. É nesse contexto que este trabalho pretende contribuir, concebendo um método que apoie o desenvolvimento de um SoS para cenários específicos de C2. O método proposto será descrito na Seção 3.2

#### 3.1 Levantamento dos Trabalhos Relacionados

Entre os trabalhos relacionados mais recentes voltados para a concepção de SoS, foram selecionados 9 trabalhos. A seguir apresentamos brevemente do que trata cada trabalho e as lacunas deixadas por cada um deles, justificando dessa forma, a proposta de um método que atenda melhor os requisitos para a concepção de um SoS.

O método proposto por Fernandes et al.(23) foca especificamente em Sistemas de Sistemas de Informação (SoIS), que são um tipo específico de SoS com ênfase em sistemas de informação, e não a qualquer tipo de SoS. Se concentra na fase inicial de identificação de links de interoperabilidade, não abrangendo as etapas de desenvolvimento, implementação e validação de um SoS robusto. O artigo não detalha as atividades específicas dos processos de negócio, nem explicita suas entradas e saídas. Essa falta de detalhamento pode dificultar a identificação da interoperabilidade em um nível mais granular, que é essencial para a

modelagem de um SoS. Embora proponha um método, ele não oferece uma metodologia detalhada que guie o desenvolvimento do SoS em todas as suas fases.

Em seu artigo, Fernandes, Neto e Santos(24) oferecem uma visão abrangente sobre a interoperabilidade em Sistemas de Sistemas de Informação (SoIS), identificando dez fatores influentes. No entanto, não propõem uma metodologia detalhada para a concepção, desenvolvimento e implementação de um SoS. O artigo traz uma proposta de um modelo conceitual, porém esse modelo é abstrato e não detalha os processos operacionais, os fluxos de informação e as interações específicas entre os sistemas constituintes. Os autores discutem a importância da interoperabilidade, mas não detalham como alcançar diferentes níveis (técnico, semântico, organizacional) em um SoS. Essas lacunas trazem a necessidade de um método mais completo, prático e específico para a modelagem e concepção de um SoS.

No trabalho de Vaneman e Budka(25), os autores definem o modelo de processo “Vee” do SoSE&I e discute como ele é usado para projetar o SoS ao longo de seu ciclo de vida. Trata-se de uma abordagem estruturada para a integração de sistemas no contexto militar, especialmente na Marinha dos EUA. Apesar de sua contribuição, o modelo apresenta algumas limitações que impactam sua aplicabilidade em um contexto mais genérico de SoS. Não é detalhado um método específico para a modelagem dos processos e interações, e também não são detalhadas as atividades específicas de cada fase, como levantamento de requisitos, projeto detalhado, implementação e teste. O artigo reconhece a importância da interoperabilidade, mas não detalha como ela é alcançada ou como os sistemas constituintes interagem entre si de forma específica. Embora seja mencionado que o SoS é formado por sistemas independentes, eles não detalham como a autonomia dos constituintes é mantida e gerenciada.

A metodologia proposta por Baek et al.(26), por exemplo, propõe uma abordagem mais sistêmica, com foco no desenvolvimento iterativo e em um processo contínuo de coleta e análise de dados. Esse enfoque é uma melhoria significativa em relação ao modelo “Vee”, pois oferece um *framework* que pode ser adaptado de acordo com as necessidades específicas de cada fase do ciclo de vida do SoS, ao mesmo tempo em que garante a adaptação contínua a novas informações e requisitos. No entanto, não detalha a modelagem dos processos de negócio e fluxos de informação entre os sistemas constituintes. Também não detalha as atividades específicas de cada fase do ciclo de vida de um SoS e nem especifica um método para lidar com a complexidade da interação entre sistemas heterogêneos e autônomos em um SoS.

O metamodelo MeMSoS, desenvolvido por Dridi, Benzadri e Belala(27), apresenta uma abordagem robusta para modelagem de SoSs ao oferecer um *framework* que facilita a interoperabilidade e a reutilização de capacidades, funções e objetivos entre subsistemas. Embora o MeMSoS seja vantajoso por promover a interoperabilidade e a reutilização de

componentes, ele apresenta algumas limitações relevantes quando comparado às necessidades específicas para a concepção de um SoS abrangente. Uma de suas limitações é a falta de um processo detalhado de desenvolvimento que inclua as etapas de análise de requisitos, design e implementação de um SoS. O MeMSoS concentra-se na descrição e organização dos elementos constitutivos do SoS, mas não oferece diretrizes metodológicas claras para conduzir a concepção de um SoS desde o levantamento dos requisitos até a implementação final. Além disso, o MeMSoS foca principalmente na reutilização de subsistemas e em uma visão abstrata da interoperabilidade.

A abordagem de metamodelagem para SoS baseados em IoT proposta por Smaali e Boucebsi(28) apresenta algumas vantagens, especialmente em termos de flexibilidade, escalabilidade e interoperabilidade, que são essenciais para a integração de múltiplos sistemas heterogêneos. No entanto, a complexidade da configuração, a dependência de padrões e as limitações para gerenciar segurança e tempo real são desafios que devem ser considerados na escolha dessa metodologia para modelagem de um SoS. Apesar de fornecer uma estrutura modular e orientada a serviços, essa abordagem não apresenta um processo completo que cubra todas as etapas do ciclo de vida do sistema e os requisitos específicos de um SoS.

O estudo de Xue et al.(29) traz uma importante aplicação do AERSoS na gestão de emergências em incêndios florestais, utilizando modelos baseados no DoDAF para representar os pontos de vista operacional (OV) e de capacidade (CV) do sistema. No entanto, essa abordagem apresenta algumas limitações que dificultam sua aplicação em cenários mais abrangentes. O artigo não propõe um método para a concepção ou desenvolvimento do SoS, ele utiliza o DoDAF para representar o sistema já existente e documentar suas interações, hierarquias e dependências entre sistemas constituintes no cenário de combate a incêndios. Embora o uso de modelos OV permita visualizar aspectos operacionais e fluxos de recursos, esses modelos são focados na descrição do sistema e não na integração efetiva dos constituintes. Outra limitação é a ausência de um processo detalhado de entrada e saída de informações e requisitos, que são essenciais para a integração de sistemas constituintes em um SoS. Ele não detalha as especificidades da interoperabilidade entre os diferentes sistemas, nem as atividades específicas desses processos.

Embora o estudo de Paes et al.(30) apresente importantes desafios, há uma limitação significativa na falta um método consolidado para apoiar o desenvolvimento e evolução do SoS. O foco estava na interoperabilidade de sistemas militares existentes, com uma abordagem inicial de integração baseada em modelagem de processos e especificação de requisitos. Apesar de usar ferramentas como BPMN e UML, o trabalho enfrentou desafios com a falta de uma visão do SoS como um todo, limitando sua eficácia para capturar a complexidade de um SoS completo. Essa proposta detalha as interações entre os sistemas utilizando linguagens de especificação, para capturar a complexidade e a dinâmica

do SoS. O planejamento das operações e a validação dos sistemas no SisGAAz foram realizados de forma parcial, sem uma metodologia sistemática que cobrisse todas as etapas do desenvolvimento.

O plano de pesquisa proposto por Cook e Pratt(31) apresenta uma abordagem interessante e estruturada para investigar desafios e metodologias no contexto de defesa, particularmente com relação ao desenvolvimento de capacidades de SoS para a defesa australiana. Embora apresente uma estrutura abrangente, possui algumas limitações. O trabalho não apresenta um processo sistemático de desenvolvimento de SoS que abranja desde sua especificação até a sua implementação. Além disso, o método se concentra na análise de abordagens globais e na identificação de desafios através de entrevistas e *workshops*. Embora essa fase permita uma compreensão aprofundada dos desafios de interoperabilidade e integração, não oferece um modelo claro e replicável para a concepção e desenvolvimento prático de um SoS.

Esta seção trouxe uma análise das abordagens e metodologias existentes para o desenvolvimento e integração de SoS, incluindo iniciativas de interoperabilidade e resiliência em contextos militares e de defesa. São apresentados métodos como o “Vee” para Engenharia e Integração de SoS (25), MBSoSE para apoiar a engenharia de SoS em diferentes domínios (26), o modelo conceitual de interoperabilidade para Sistemas de Sistemas de Informação (SoIS) (24), criação de links de interoperabilidade para compor um Sistema de Sistemas de Informação (SoIS) (23) e o metamodelo MeMSoS (27), que abordam, respectivamente, a organização do processo de desenvolvimento, modelagem baseada em simulação, a identificação de fatores críticos para integração, abordagem prática para identificação e padronização de pontos de interoperabilidade e a reutilização de subsistemas para reduzir custos. Adicionalmente, a seção incluiu abordagens voltadas à defesa nacional, uma que detalha um plano de adaptação de metodologias globais para as particularidades australianas (31), o SisGAAz, que integra sistemas de comando e controle da Marinha do Brasil com foco em interoperabilidade (30) e os modelos operacionais de AERSoS aplicados à gestão de emergências, enfatizando a visualização de interações e hierarquias de sistemas em diferentes pontos de vista (29). Em um contexto de interoperabilidade orientada a IoT, é apresentada uma metamodelagem para SoS, focando na flexibilidade e escalabilidade para incorporar novos dispositivos (28). Cada abordagem revisada foi analisada em relação às limitações e desafios que apresenta, especialmente no que se refere à falta de uma metodologia que consiga abranger todo o ciclo de vida da concepção do SoS (modelagem completa), desde a especificação até a implementação e testes, e aos requisitos essenciais para os sistemas.

Assim, a revisão conduzida fornece uma base sólida para a justificativa da metodologia proposta neste trabalho, que busca suprir essas lacunas desenvolvendo um método completo e estruturado que abranja todas as etapas do processo de concepção, considerando

Ref.	Cobertura dos trabalhos relacionados								
	(23)	(24)	(29)	(30)	(26)	(31)	(27)	(28)	(25)
Mapeamento entradas e saídas	X								
Fluxos com interações			X	X				X	
Modelagem completa do SoS									
Estudo de caso detalhado			X	X				X	

Tabela 1 – Cobertura dos Trabalhos Relacionados

as particularidades de um SoS, como a necessidade de interoperabilidade. Na sequência será apresentada a nossa proposta de método para esta modelagem, que busca cobrir as características apresentadas na Tabela 1, que evidencia as lacunas existentes nos trabalhos relacionados apresentados neste capítulo.

## 3.2 Processo para Modelagem de um SoSC2

O problema de pesquisa está centrado na necessidade de integrar diferentes sistemas, garantindo que eles possam operar de maneira coordenada, compartilhando informações em tempo real e adaptando-se a diferentes situações. O método proposto, apresentado na Figura 3, busca permitir a comunicação e a troca de informações em tempo real entre diferentes constituintes, buscando permitir uma melhoria na comunicação em processos de tomada de decisão.

O método contempla três etapas principais: (i) levantamento de requisitos do SoS, (ii) Projeto do SoS, e (iii) construção do SoS. As duas primeiras etapas contemplam procedimentos que envolvem a modelagem de processos em nível conceitual. Essas etapas serão detalhadas a seguir. A terceira etapa, no entanto, encontra-se em desenvolvimento e será apresentada no próximo relatório, cujo foco é na especificação de procedimentos que contemplam atividades no nível lógico e operacional.

### 3.2.1 Levantamento de Requisitos do SoS

O levantamento de requisitos em Sistemas de Sistemas (SoS) envolve uma sequência de atividades que garantem a coleta, análise, definição e validação de requisitos de maneira eficiente, considerando a complexidade e a natureza interdependente dos sistemas envolvidos.

- **Definir cenário alvo:** Estabelecer um cenário alvo envolve algumas tarefas que buscam garantir que a comunicação, a coordenação e a gestão dos recursos atendam as expectativas de maneira adequada. É necessário definir claramente os objetivos,

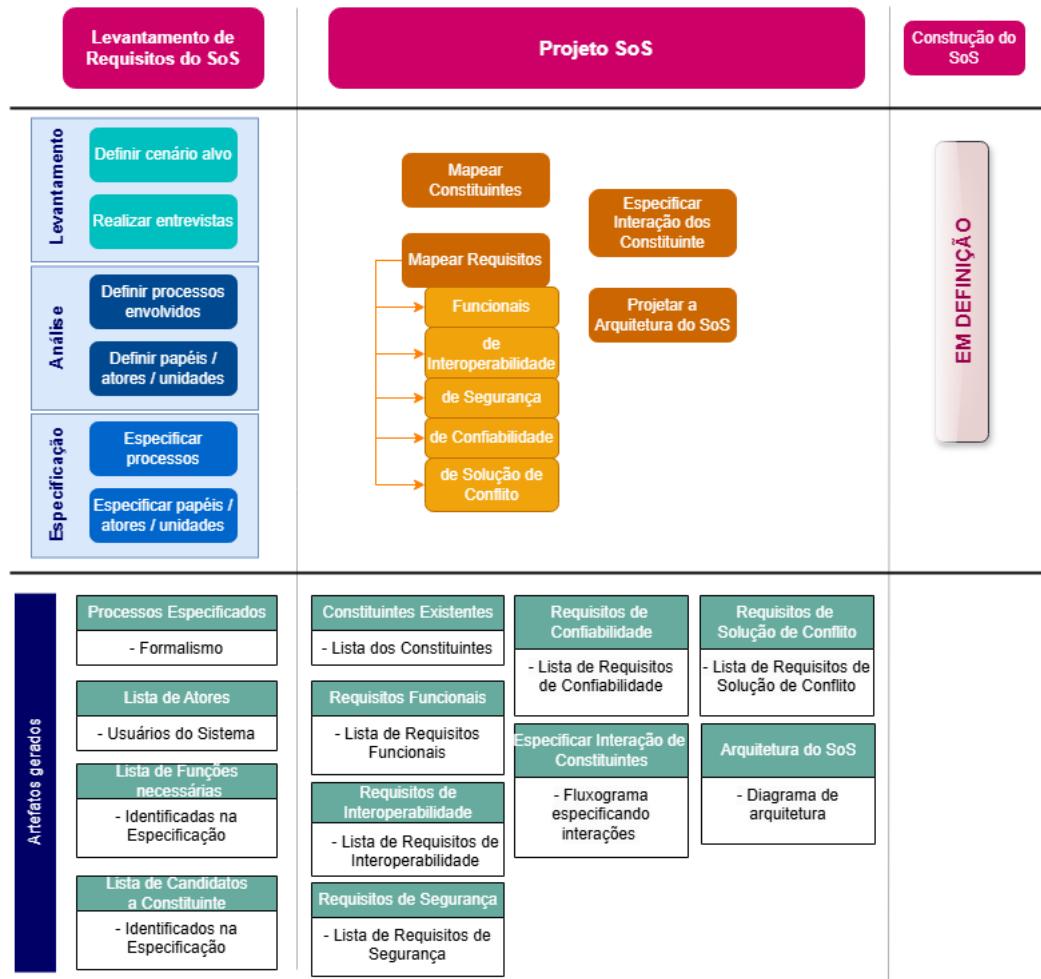


Figura 3 – Método para modelagem e concepção de um SoS

determinando as partes envolvidas assim como quais recursos serão necessários para a missão.

- Estabelecer a finalidade do SoS no contexto do cenário (definir o objetivo do cenário alvo)
- Delimitar o escopo do cenário (sua abrangência e limitações)
- **Realizar entrevistas:** No processo de concepção de um SoS, os entrevistados devem incluir uma variedade de perfis. Em particular, é fundamental envolver a equipe de desenvolvimento dos sistemas candidatos a integrar o SoS, assim como os diferentes tipos de usuários que utilizam esses sistemas. Muitas informações importantes são obtidas através da experiência desses profissionais, e dificilmente são encontradas nas leituras dos documentos. Duas tarefas precisam ser mapeadas nesta atividade:
  - Definir os especialistas (escolher os mais relevantes, em diferentes perspectivas, indo da parte técnica até a organizacional)

- Elaborar roteiro de entrevista (pode ser através de um questionário). O roteiro é necessário para que a entrevista seja objetiva. Preferencialmente, o entrevistador precisa ter um conhecimento considerável sobre o material que descreve o cenário e sobre os manuais dos sistemas em uso, para poder ser objetivo com os entrevistados.
- **Definir processos envolvidos:** É uma tarefa em que identificam-se os processos que devem ser detalhados, buscando ficar dentro do escopo definido anteriormente.
  - Analisar processos atuais nos sistemas individuais (como cada sistema opera atualmente)
  - Elencar o material específico sobre cada processo
  - Realizar entrevistas e coletar experiências com a equipe que executa ou participa do processo
- **Definir papéis/atores/unidades:** Para cada processo identificado é necessário definir o papel de cada ator, estabelecendo sua função e responsabilidade.
  - Listar os principais atores e unidades envolvidas (identificando todos os sistemas, organizações e indivíduos que interagem direta ou indiretamente com o SoS)
  - Definir os papéis (mapear as funções principais que cada papel/ator ou unidade desempenha no contexto do SoS)
- **Especificiar processos:** busca detalhar as atividades e a interdependência entre elas.
  - Mapear as interações (detalhar como os atores/papéis se conectam e colaboram entre si dentro do SoS, gerando um fluxo de troca de informações)
- **Especificiar papéis/atores/unidades:** atividade que compreende uma definição de hierarquias e responsabilidades necessárias ao funcionamento do SoS.
  - Descrever responsabilidades de cada papel (detalhar as atribuições específicas de cada ator ou unidade)
  - Definir hierarquias ou níveis de autoridade (estabelecer uma estrutura hierárquica ou funcional para indicar como os papéis se relacionam em termos de supervisão e execução)

Os artefatos gerados nesta fase incluem:

- Processos especificados expressos em algum formalismo;
- Lista de atores (usuários do sistema);

- Lista de funções necessárias;
- Lista de candidatos a constituintes.

### 3.2.2 Projeto SoS

Fase fundamental para garantir que o sistema seja projetado de modo a atender aos requisitos do SoS. Consiste em um processo complexo, que envolve a criação de uma arquitetura capaz de integrar diferentes sistemas independentes, atendendo a requisitos técnicos e operacionais, além de garantir a interoperabilidade e a capacidade de evoluir conforme as necessidades.

- **Mapear Constituintes:** Tem como objetivo determinar quais sistemas individuais (ou subsistemas) farão parte do SoS, classificando os possíveis constituintes de acordo com suas funcionalidades e como eles se inter-relacionam para atingir os objetivos do SoS.
  - Identificar constituintes (todos os sistemas ou subsistemas disponíveis que podem ser considerados constituintes do SoS)
  - Mapear as funcionalidades principais de cada sistema (descrever brevemente as capacidades e limitações de cada um)
  - Definir o papel de cada constituinte no contexto do SoS e como ele contribui para os objetivos gerais.
  - Estabelecer fluxos de comunicação e troca de informações (criar diagramas ou fluxogramas que representem as interações entre os sistemas)
  - Representar graficamente o SoS (desenvolver um diagrama de alto nível que mostre todos os constituintes e as interações entre eles)
- **Mapear Requisitos Funcionais:** São especificações que descrevem as funcionalidades ou capacidades essenciais que o SoS deve realizar para cumprir sua missão ou atender aos objetivos esperados. Definem o que o SoS precisa fazer para que os sistemas constituintes, quando integrados, possam colaborar e realizar as operações desejadas de maneira coordenada. Esses requisitos vão além das funcionalidades individuais de cada sistema constituinte, focando nas ações que o SoS como um todo deve realizar.
  - Listar funcionalidades gerais do SoS (criar uma lista das capacidades esperadas para o SoS como um todo, incluindo ações que exigem a colaboração entre os constituintes)
  - Detalhar os requisitos funcionais, respondendo a questões como: (i) O que o SoS deve fazer? (ii) Como as funcionalidades serão realizadas em colaboração

entre os constituintes? (iii) Quais entradas e saídas são necessárias para cada funcionalidade?

- **Mapear Requisitos de Interoperabilidade:** Buscar a interoperabilidade significa permitir que os sistemas constituintes do SoS possam interagir de maneira eficiente e sem falhas, independentemente de suas diferenças tecnológicas.
  - Revisar as interações esperadas no SoS (mapear os pontos de integração onde a interoperabilidade será essencial)
  - Analisar as características dos sistemas constituintes (listar protocolos, arquiteturas e formatos de dados usados por cada constituinte, assim como identificar possíveis incompatibilidades técnicas entre eles)
  - Especificar os padrões de interoperabilidade (ex. protocolos de comunicação, formatos de dados comuns).
  - Definir mecanismos de compatibilidade técnica (propor métodos para converter dados entre sistemas com tecnologias ou protocolos diferentes)
  - Criar uma matriz de interoperabilidade (organizar os requisitos em uma tabela que relate: (i) constituintes envolvidos; (ii) padrões e protocolos utilizados; (iii) pontos críticos de interoperabilidade)
  - Desenvolver diagramas de interação (representando visualmente como os constituintes se conectam e trocam informações)
- **Mapear Requisitos de Segurança:** Definir como os dados serão protegidos, visando garantir sua integridade.
  - Analisar os objetivos e cenários do SoS relacionados à segurança (identificar os tipos de informações sensíveis trocadas no SoS)
  - Definir requisitos de confidencialidade (estabelecer permissão de acesso e mecanismos de autenticação/autorização, assegurando que apenas os usuários autorizados possam acessar determinadas informações ou executar algumas ações)
- **Mapear Requisitos de Confiabilidade:** Planejar como o SoS responderá a falhas, assegurando que seja robusto e confiável, e também atendendo aos requisitos de disponibilidade e resiliência.
  - Estabelecer requisitos de disponibilidade (constituintes devem ter acesso contínuo e confiável às informações, mesmo em cenários adversos)
  - Definir requisitos de resiliência (propor medidas de contenção para evitar propagação de falhas que comprometam o SoS como um todo)

- **Mapear Requisitos de Solução de Conflito:** Resolver conflitos que possam surgir entre os sistemas constituintes, seja no nível de dados, processos ou interação. Envolve especificar como conflitos serão resolvidos, seja em relação ao compartilhamento de recursos ou prioridades de execução.
  - Identificar os tipos de conflitos no SoS (que podem ser: (i) Conflitos de dados: inconsistências, redundâncias ou falta de padronização; (ii) Conflitos de recursos: disputas por acesso a recursos compartilhados, por exemplo, rede, energia, armazenamento; (iii) Conflitos de processos: sobreposição de funcionalidades)
  - Estabelecer regras para resolução de conflitos de dados (definindo critérios de prioridade para dados conflitantes, como por exemplo, fonte mais confiável, *timestamp* mais recente).
  - Estabelecer políticas para alocação de recursos (propor métodos de escalonamento ou redistribuição de recursos quando necessário)
  - Definir mecanismos para resolução de conflitos de processos (determinando um fluxo de decisão para situações em que processos sobrepostos precisam ser priorizados)
  - Definir tratamento de exceção, para garantir a continuidade e a confiabilidade das operações (exceções podem ocorrer em várias formas, como falhas de comunicação, erros de dados, falhas de componentes, ou violações de segurança. O tratamento de exceção em SoS refere-se ao conjunto de procedimentos e mecanismos implementados para detectar, isolar, resolver e, se necessário, mitigar os impactos de falhas e erros. Estes procedimentos são necessários para que o SoS possa continuar a operar mesmo na presença de problemas, minimizando a interrupção dos serviços)
- **Especificiar Interação dos Constituintes:** Detalhar como os sistemas constituintes irão interagir uns com os outros para alcançar os objetivos do SoS.
  - Identificar os pontos de interação entre os constituintes (mapear quais sistemas precisam se comunicar diretamente, especificando dados a serem enviados e recebidos)
  - Desenvolver diagramas de fluxo de comunicação (representar visualmente os fluxos de dados entre os constituintes, quem envia o quê e para quem)
  - Especificar protocolos de comunicação adequados (definir quais protocolos serão usados nas interações)
  - Definir formatos padrão para mensagens (garantir que os formatos atendam às necessidades dos constituintes e do SoS como um todo)

- Definir requisitos de desempenho para interações (especificar limites de tempo para envio e resposta de mensagens, assim como determinar requisitos de largura de banda e capacidade de processamento)
- Criar uma matriz de interação (onde são listados os constituintes e detalhados: (i) Quem interage com quem; (ii) Tipos de mensagens trocadas; (iii) Protocolos e formatos usados)
- **Projetar a Arquitetura do SoS:** Projetar uma arquitetura que assegure o atendimento de todos os requisitos de segurança, interoperabilidade, confiabilidade e interação necessários para o funcionamento do SoS.
  - Criar um diagrama inicial da arquitetura (representando os sistemas constituintes, suas conexões e interações principais)

Os artefatos gerados são:

- Lista de Constituintes;
- Lista de Requisitos Funcionais;
- Lista de Requisitos de Interoperabilidade;
- Lista de Requisitos de Segurança;
- Fluxograma especificando interações entre constituintes;
- Diagrama inicial da arquitetura.

### 3.2.3 Considerações adicionais

Cada constituinte em um SoS pode ser projetado para atender a requisitos específicos e operacionais distintos, de acordo com seu contexto ou domínio. A autonomia permite que esses sistemas sejam mantidos e evoluídos de forma independente, sem que mudanças em um constituinte necessariamente afete os outros. Ter autonomia contribui para a resiliência de todo o SoS, pois se um constituinte falhar ou estiver indisponível, os outros podem continuar a operar sem grandes interrupções, garantindo a continuidade das operações. A autonomia também permite que cada constituinte evolua e se atualize sem depender de outros, o que facilita, por exemplo, a incorporação de novos recursos sem causar impacto direto nos demais dentro do SoS.

A autonomia dos constituintes permite sua flexibilidade, resiliência e escalabilidade, mesmo em cenários complexos e dinâmicos. Ela permite que cada sistema funcione de maneira independente, mantendo suas características e capacidades, enquanto ainda contribui para os objetivos globais do SoS. Nesse contexto, a abordagem federada se

destaca como uma alternativa adequada, permitindo que cada constituinte mantenha sua autonomia e flexibilidade, operando de forma independente, mas colaborando dentro do SoS.

Como já foi dito anteriormente, no contexto da Família de Aplicativos de C2 da Força Terrestre (FAC2FTer) e de outros conjuntos de sistemas de C2 das FFAA, os sistemas candidatos a constituintes já existem e funcionam autonomamente. Porém, ainda não formam um SoS, i.e., precisam interoperar. A fase seguinte do método proposto está em desenvolvimento, e leva em consideração a implementação de sistemas constituintes que fazem um papel integrador entre os sistemas autônomos existentes, de modo a interferir o mínimo possível na sua autonomia, e assim, atender a demanda por flexibilidade, resiliência e escalabilidade. A especificação desse procedimento deve ser o foco do próximo relatório.

## 4 RESULTADOS

Neste capítulo final, as questões de pesquisa mencionadas no Capítulo 1 são reapresentadas a seguir com suas respectivas respostas e resultados.

### 4.1 Construção de Ontologias de Referência

**Questão de Pesquisa:** *É possível construir uma ontologia de referência que possa servir de base para a representação e intercâmbio de dados nos SoSC2, facilitando a interoperabilidade entre estes?*

Sim. Através do procedimento sistemático para desenvolvimento de ontologias apresentado no Capítulo 2, foi possível desenvolver as ontologias CROMO, MISCON e ATOP.

No trabalho de Silva (32) foi desenvolvida a CROMO, uma ontologia de referência para representar cenários militares de operação de C2 utilizando rádios cognitivos. Como resultados deste trabalho, destacam-se:

- As definições formais dos conceitos de *situação* e *cena*, bem como suas relações mereológicas envolvendo relações de parte e todo. Além da harmonização desses conceitos no contexto da UFO, apontando as relações existentes entre cenas e eventos, disposições e proposições, além de sua influência nas relações de causalidade entre eventos e ações;
- O desenvolvimento da ontologia *CROMO*, cobrindo o universo de discurso associado ao processo decisório em três porções de conhecimentos, a saber: (i) a descrição do processo de planejamento de comunicações alinhado com o planejamento das operações; (ii) a descrição de um cenário operacional composto por cenas e situações detectáveis por um rádio cognitivo sob pelo menos um aspecto técnico e aspectos táticos relacionados a pelo menos uma operação militar; e (iii) a descrição dos elementos que compõem o Cenário Operacional e as ações envolvidas.
- Publicações: (33, 34, 35, 36)

Já o trabalho de Demori (37) apresentou o desenvolvimento da ontologia MISCON, uma ontologia de referência para representar cenários militares levando em conta dispositivos de telecomunicações e as características de seus portadores. A ontologia MISCON foi utilizada como base para representar dados de duas aplicações de simulação de cenários de comunicação em operações de C2. Entre as contribuições deste trabalho, destacam-se:

- As definições formais dos conceitos de Organização Militar, seus subtipos e as relações de subordinação entre estes.
- A representação dos agentes militares, suas filiações e as plataformas militares.
- A representação dos equipamentos de comunicação e as regras de comunicação conforme a alocação dos mesmos aos agentes militares.
- Publicações: (38, 39, 40, 41, 42, 43)

O mesmo procedimento sistemático foi utilizado para o desenvolvimento de uma terceira ontologia no contexto do projeto S2C2, fruto do trabalho de Jesus (44). Este trabalho apresenta a ontologia de referência chamada ATOp-NavalOntology, cujo objetivo é representar recursos necessários às operações de C2 envolvendo meios operativos navais. Entre as contribuições deste trabalho, destacam-se:

- As definições formais dos conceitos como Plataforma Naval militar, Compartimentos de Armazenamento, Suprimentos e seus subtipos.
- A representação de eventos de operações navais típicas envolvendo plataformas navais, como a atracação e a desatracação.

**Questão de Pesquisa:** *É possível construir uma ontologia de referência que possa servir de base para a representação de redes de comunicação heterogêneas de modo a manter os sistemas constituintes de um SoSC2 bem conectados e com isso assegurar o intercâmbio de dados e a interoperabilidade entre eles?*

Sim. Através do procedimento sistemático para desenvolvimento de ontologias apresentado no Capítulo 2, foi possível desenvolver as ontologias HINT e QVAS. O trabalho de Tesolin (45) apresentou o desenvolvimento dessas duas ontologias de referência. A HINT tem como foco a representação de redes de comunicação sem fio, visando sua melhor gerência e aproveitamento. Já a QVAS abarca conceitos necessários ao monitoramento e registro dos eventos na rede em um cenário de comunicações de uma operação de C2 em andamento. Dentre as contribuições deste trabalho, destacam-se:

- A representação de um novo conceito para aprimorar a descrição ontológica de um ambiente sem fio móvel - o MEIO (Medium), que não existe nas ontologias de rede disponíveis.
- A revisão da representação de alguns conceitos, como Servidor (Server) e Vizinho (Neighbor), que estão mal descritos em algumas ontologias relacionadas à rede e são fundamentais para dar suporte ao processo de transferência (*handover*).

- A desambiguação dos conceitos de **Link** e **Connection**, fornecendo uma compreensão clara do que são, quais entidades vinculam e como estão relacionados.
- A representação dos níveis de rede.
- A representação de medições de sinal, potência, ruído, etc.
- Publicações: (33, 46, 42, 47)

Todas as ontologias mencionadas acima foram desenvolvidas seguindo a recomendação da abordagem SABiO, i.e., utilizando como base uma ontologia de fundamentação. O resultado deste processo são ontologias bem fundamentadas, que tornam explícito seu compromisso ontológico com uma ontologia de fundamentação, e podem portanto, ser classificadas como ontologias de referência. Os artefatos gerados na etapa de captura e formalização das ontologias foram criados utilizando a ferramenta Visual Paradigm<sup>1</sup> habilitada com a linguagem OntoUML<sup>2</sup>, que incorpora a ontologia de fundamentação UFO (9). Os arquivos com essas ontologias de referência em formato VPP estão disponíveis no repositório<sup>3</sup> ou na página<sup>4</sup> do Projeto S2C2.

## 4.2 Apoio à Construção de SoSC2

**Questão de Pesquisa:** *Como apoiar o desenvolvimento de um SoS para aumentar a interoperabilidade de sistemas de C2?*

A Capítulo 3 procurou responder a esta pergunta através de um levantamento bibliográfico e da proposta de um procedimento sistemático, ou um método para apoiar a construção de um SoS. O detalhamento de tal método encontra-se em (48), e entre suas contribuições principais, destacam-se:

- O mapeamento de processos e identificação de falhas nas interações e dependências entre os sistemas candidatos a constituintes.
- O detalhamento de quais artefatos que devem ser gerados, para explicitar os dados e as novas funcionalidades que requer um SoS.
- A definição de uma arquitetura de SoS incluindo os sistemas existentes e novos sistemas constituintes integradores. Estes últimos atendendo as novas funcionalidades que surgem da interoperabilidade com os sistemas existentes.

<sup>1</sup> <https://www.visual-paradigm.com/>

<sup>2</sup> <https://dev.ontouml.org/>

<sup>3</sup> <https://github.com/comp-ime-eb-br/S2C2-IME/tree/main/diagrams>

<sup>4</sup> <https://comp-ime-eb-br.github.io/S2C2-IME/ontodesc/>

- A concepção de um método de construção de SoS minimizando a interferência nos sistemas já existentes, e adaptável a diferentes cenários operacionais, além de ser escalável para incorporar novos sistemas ou tecnologias conforme necessário, contribuindo para a evolução contínua da arquitetura do SoS.

### 4.3 Considerações Finais

O presente relatório apresenta os resultados alcançados até então no contexto do projeto S2C2, em desenvolvimento pelo IME. Como relatado nas seções anteriores, o projeto atingiu suas metas, respondendo às questões de pesquisa colocadas nas etapas iniciais do projeto, tendo gerado contribuições relevantes, não só em termos de artefatos (softwares e ontologias), mas também em termos de publicação (artigos em conferências e em revistas). Os artefatos e publicações gerados nesse contexto estão disponíveis através do site<sup>5</sup> do projeto, e do repositório<sup>6</sup> associado, e seguem totalizados conforme Tabela 2.

Além das publicações já mencionadas nas seções 4.1 e 4.2, o conjunto de publicações inclui também artigos relacionados com foco em C2 na linha de mineração de texto com uso de modelos de linguagem (IA) para extração de conhecimento (49, 50) e para enriquecimento semântico (51, 52, 53), e inclui ainda, outros trabalhos complementares dos membros da equipe (54, 55, 56).

Resultados	Total	Referências
Artigos em Conferência Nacional	8	(33, 39, 40, 51, 54, 53, 49, 56)
Artigos em Conferência Internacional	5	(34, 41, 42, 52, 50)
Artigos em Revistas	3	(47, 36, 55)
Capítulos de Livro	1	(43)
Ontologias de Referência	5	(57)
Ontologias Operacionais	3	(58, 59, 60)
Registros de Software	3	(61, 62, 63)
Teses de doutorado	2	(32, 45)
Dissertações de Mestrado	2	(37, 44)
Projetos de Final de Curso	3	(35, 46, 38)
Teses de Doutorado em andamento	4	
Dissertações de Mestrado em andamento	2	
Cursos de Curta Duração Oferecidos	4	Visitantes Especialistas

Tabela 2 – Sumarização dos resultados obtidos

Analizando os resultados, em 4 anos de projeto, foi possível alcançar resultados significativos, tanto em publicações e artefatos, como também com relação à formação de recursos humanos. A Tabela 2 revela que foram gerados 35 publicações/artefatos, e que o

<sup>5</sup> <https://comp-ime-eb-br.github.io/S2C2-IME/>

<sup>6</sup> <https://github.com/comp-ime-eb-br/S2C2-IME>

projeto contribuiu para a formação de 13 alunos até o momento. Vale ainda destacar que alguns dos alunos envolvidos no projeto integraram, ou ainda, poderão integrar as equipes do Centro de Desenvolvimento de Sistemas do EB, justificando desta forma, o investimento encomendado à financiadora do projeto. Além disso, outros 2 alunos que foram formados no contexto do projeto, hoje atuam junto à MB, justamente na área de C2. Esse resultado se deve a uma equipe de professores do IME e da UFRJ, e seus colaboradores, atuantes nas áreas de interoperabilidade de dados e ontologias. Tais professores formaram um time coeso e atuaram colaborativamente na orientação dos alunos envolvidos, guiando-os em temas dentro dos interesses das FFAA e da sociedade.

Uma outra contribuição para a formação da equipe do projeto, que beneficiou professores e alunos, foi a organização de cursos de curta duração ministrados por pesquisadores especialistas reconhecidos na área de Semântica e Serviços, da equipe da Universidade de Twente (UT) da Holanda. Foram ao todo 4 visitas e os cursos foram oferecidos a professores e alunos além da equipe do IME, atingindo um número aproximado de 70 participantes ao todo.

Assim, além dos produtos técnicos e científicos gerados, os impactos positivos extrapolam o âmbito acadêmico, alcançando também a sociedade por meio de ações de divulgação, inovação e aplicação prática dos resultados. Dessa forma, reafirma-se a relevância do apoio recebido, evidenciando que os recursos foram empregados com responsabilidade, planejamento e foco em resultados concretos. Por fim, considerando que alguns trabalhos ainda estão em andamento, espera-se que sejam gerados ainda mais resultados a serem reportados no próximo relatório.

## REFERÊNCIAS

- 1 RUY, F. B.; GUIZZARDI, G.; FALBO, R. A.; REGINATO, C. C.; SANTOS, V. A. From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering*, Elsevier BV, v. 109, p. 41–69, 5 2017. ISSN 0169-023X. Disponível em: <<http://dx.doi.org/10.1016/j.datak.2017.03.004>>.
- 2 HALLER, A.; JANOWICZ, K.; COX, S. J. D.; LEFRANÇOIS, M.; TAYLOR, K. L.; LE-PHUOC, D.; LIEBERMAN, J.; GARCÍA-CASTRO, R.; ATKINSON, R.; STADLER, C. The sosa / ssn ontology : A joint w3c and ogc standard specifying the semantics of sensors , observations , actuation , and sampling. In: . [s.n.], 2018. Disponível em: <<https://api.semanticscholar.org/CorpusID:204767879>>.
- 3 ZHOU, Q.; GRAY, A. J. G.; MCLAUGHLIN, S. ToCo: An Ontology for Representing Hybrid Telecommunication Networks. In: *The Semantic Web*. [S.l.]: Springer International Publishing, 2019. p. 507–522. ISBN 9783030213473.
- 4 BORGO, S.; FERRARIO, R.; GANGEMI, A.; GUARINO, N.; MASOLO, C.; PORRELLO, D.; SANFILIPPO, E. M.; VIEU, L. DOLCE: A descriptive ontology for linguistic and cognitive engineering1. *Applied Ontology*, IOS Press, v. 17, n. 1, p. 45–69, 3 2022. ISSN 1875-8533. Disponível em: <<http://dx.doi.org/10.3233/ao-210259>>.
- 5 GUIZZARDI, G. *Ontological foundations for structural conceptual models*. Tese (PhD Thesis) — University of Twente, 01 2005.
- 6 ALBERTS, D.; HAYES, R. *Understanding command and control*. [S.l.: s.n.], 2006.
- 7 VERDONCK, M.; GAILLY, F.; CESARE, S. D.; POELS, G. Ontology-driven conceptual modeling: A systematic literature mapping and review. *Applied Ontology*, IOS press, v. 10, n. 3-4, p. 197–227, 2015.
- 8 GUIZZARDI, G. On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In: IOS PRESS. *Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference, DB&IS'2006*. Vilnius, Lituânia, 2007. v. 155, p. 18.
- 9 GUIZZARDI, G. *Ontological foundations for structural conceptual models*. Tese (PhD Thesis) — University of Twente, The Netherlands, out. 2005. ISBN: 90-75176-81-3 Issue: 05-74 Series: CTIT PhD Thesis Series.
- 10 BOOCHE, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. Brasil: Elsevier Brasil, 2006.
- 11 RICHTERS, M.; GOGOLLA, M. Ocl: Syntax, semantics, and tools. In: CLARK, T.; WARMER, J. (Ed.). *Object Modeling with the OCL: The Rationale behind the Object Constraint Language*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 42–68. ISBN 978-3-540-45669-8.
- 12 CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. *ACM transactions on database systems (TODS)*, Acm Nova York, NY, EUA, v. 1, n. 1, p. 9–36, 1976.

- 13 VERDONCK, M.; GAILLY, F.; PERGL, R.; GUIZZARDI, G.; MARTINS, B.; PASTOR, O. Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study. *Information Systems*, v. 81, p. 92–103, 2019. ISSN 0306-4379.
- 14 GUERSON, J.; SALES, T. P.; GUIZZARDI, G.; ALMEIDA, J. P. A. Ontouml lightweight editor: a model-based environment to build, evaluate and implement reference ontologies. In: IEEE. *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*. Adelaide, SA, Austrália, 2015. p. 144–147.
- 15 SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M. The neon methodology framework: A scenario-based methodology for ontology development. *Applied Ontology*, IOS Press, v. 10, n. 2, p. 107–145, 9 2015. ISSN 1875-8533. Disponível em: <<http://dx.doi.org/10.3233/ao-150145>>.
- 16 FALBO, R. d. A. Sabio: Systematic approach for building ontologies. In: GUIZZARDI, G.; PASTOR, O.; WAND, Y.; CESARE, S. de; GAILLY, F.; LYCETT, M.; PARTRIDGE, C. (Ed.). *Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering 2014*. CEUR-WS.org, 2014. (CEUR, 1). March, 2019. Disponível em: <[http://ceur-ws.org/Vol-1301/ontocomodise2014\\\_2.pdf](http://ceur-ws.org/Vol-1301/ontocomodise2014\_2.pdf)>.
- 17 SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M. "the neon methodology for ontology engineering". In: \_\_\_\_\_. *"Ontology Engineering in a Networked World"*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 9–34. ISBN 978-3-642-24794-1. Disponível em: <[https://doi.org/10.1007/978-3-642-24794-1\\_2](https://doi.org/10.1007/978-3-642-24794-1_2)>.
- 18 SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A. Towards a glossary of activities in the ontology engineering field. In: CALZOLARI, N.; CHOUKRI, K.; MAEGAARD, B.; MARIANI, J.; ODIJK, J.; PIPERIDIS, S.; TAPIAS, D. (Ed.). *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), 2008. Disponível em: <[http://www.lrec-conf.org/proceedings/lrec2008/pdf/219\\_paper.pdf](http://www.lrec-conf.org/proceedings/lrec2008/pdf/219_paper.pdf)>.
- 19 LOPEZ, M.; GOMEZ-PEREZ, A.; SIERRA, J.; SIERRA, A. Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems and their Applications*, v. 14, n. 1, p. 37–46, 1999.
- 20 STAAB, S.; STUDER, R.; SCHNURR, H.-P.; SURE, Y. Knowledge processes and ontologies. *IEEE Intelligent Systems*, v. 16, n. 1, p. 26–34, 2001.
- 21 FALBO, R. de A. Sabio: Systematic approach for building ontologies. In: *ONTO.COM/ODISE@FOIS*. [s.n.], 2014. Disponível em: <<https://api.semanticscholar.org/CorpusID:17310637>>.
- 22 GRUNINGER, M. Methodology for the design and evaluation of ontologies. In: *International Joint Conference on Artificial Intelligence*. [s.n.], 1995. Disponível em: <<https://api.semanticscholar.org/CorpusID:16641142>>.
- 23 FERNANDES, J.; CORDEIRO, F.; FERREIRA, F.; NETO, V. V. G.; SANTOS, R. P. dos. A method for identification of potential interoperability links between information systems towards system-of-information systems. *iSys-Brazilian Journal of Information Systems*, v. 15, n. 1, p. 2–1, 2022.

- 24 FERNANDES, J.; NETO, V. V. G.; SANTOS, R. P. d. An approach based on conceptual modeling to understand factors that influence interoperability in systems-of-information systems. In: *XX Brazilian Symposium on Software Quality*. [S.l.: s.n.], 2021. p. 1–10.
- 25 VANEMAN, W.; BUDKA, R. Defining a system of systems engineering and integration approach to address the navy's information technology technical authority. *INCOSE International Symposium*, v. 23, p. 1202–1214, 06 2013.
- 26 BAEK, Y.-M.; MIHRET, Z.; SHIN, Y.-J.; BAE, D.-H. A modeling method for model-based analysis and design of a system-of-systems. In: *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. [S.l.: s.n.], 2020. p. 336–345.
- 27 DRIDI, C. E.; BENZADRI, Z.; BELALA, F. System of systems engineering: Meta-modelling perspective. In: *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. [S.l.: s.n.], 2020. p. 000135–000144.
- 28 SMAALI, S.; BOUCEBSI, R. Towards a metamodeling approach for iot based system of systems. 2022.
- 29 XUE, Y.; LIU, H.; TIAN, Y.; WANG, X. Study on the aviation emergency rescue system for forest firefighting based on sose. In: *IEEE. 2022 17th Annual System of Systems Engineering Conference (SOSE)*. [S.l.], 2022. p. 81–86.
- 30 PAES, C. E. d. B.; NETO, V. V. G.; MOREIRA, T.; NAKAGAWA, E. Y. Conceptualization of a system-of-systems in the defense domain: An experience report in the brazilian scenario. *IEEE Systems Journal*, IEEE, v. 13, n. 3, p. 2098–2107, 2018.
- 31 COOK, S. C.; PRATT, J. M. Towards designing innovative sose approaches for the australian defence force. In: *2014 9th International Conference on System of Systems Engineering (SOSE)*. [S.l.: s.n.], 2014. p. 295–300.
- 32 SILVA, M. A. A. da. *Combinando Técnica e Doutrina por Meio de Conceitos Ontológicos para Representar Cenários Operacionais Militares em Sistemas de Comunicações Cognitivos*. Tese (Doutorado) — Instituto Militar de Engenharia, 02 2023.
- 33 TESOLIN, J.; SILVA, M.; CAMPOS, M.; MOURA, D.; CAVALCANTI, M. C. Critical Communications Scenarios Description Based on Ontological Analysis. In: *ONTOBRAS*. [S.l.: s.n.], 2020.
- 34 MOURA, L. d. A. L.; SILVA, M. A. A. d.; CORDEIRO, K. d. F.; CAVALCANTI, M. C. A Well-founded Ontology to Support the Preparation of Training and Test Datasets. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 2*. SCITEPRESS, 2021. p. 99–110. Disponível em: <<https://doi.org/10.5220/0010460000990110>>.
- 35 BOZZA, G.; RUY, M. M. *Desenvolvimento De Um Ambiente Para Testes De Raciocínio Em Um Rádio Cognitivo*. 2021. Projeto de Final de Curso de Graduação IME. Disponível em: <[http://www.comp.ime.eb.br/graduacao/?pc=p\\_detp&num=479](http://www.comp.ime.eb.br/graduacao/?pc=p_detp&num=479)>.
- 36 SILVA, M. A. A. d.; BOZZA, G.; RUY, M. M.; CAVALCANTI, M. C. R.; MOURA, D. F.; PIRES, L. F. Engenharia dirigida a modelos aplicada à cognição de rádios em operações militares. *RMCT*, v. 40, n. 1, p. 53–66, 2023.

- 37 DEMORI, A. M. *UMA ABORDAGEM BASEADA EM ONTOLOGIA PARA REPRODUÇÃO DE CENÁRIOS DE OPERAÇÕES MILITARES*. Dissertação (Mestrado) — Instituto Militar de Engenharia (IME), 12 2023.
- 38 PONTES, D. H. M.; ROCHA, C. C. B. *Um Simulador Para Redes Heterogêneas sob o Paradigma "Always Best Connected*. 2022. Projeto de Final de Curso de Graduação IME - Mimic. Disponível em: <[http://www.comp.ime.eb.br/graduacao/?pc=p\\_detp&num=494](http://www.comp.ime.eb.br/graduacao/?pc=p_detp&num=494)>.
- 39 DEMORI, A. M.; TESOLIN, J. C. C.; CAVALCANTI, M. C. R.; MOURA, D. F. C. Supporting simulation of military communication systems using well-founded modeling. In: *Proc of the XV Seminar on Ontology Research in Brazil (ONTOBRAS)*. [s.n.], 2022. v. 3346, p. 73–84. Disponível em: <<https://ceur-ws.org/Vol-3346/Paper6.pdf>>.
- 40 DEMORI, A. M.; SILVA, R. R. M. R. da; ALBUQUERQUE, P. P. L. de; MOURA, D. F. C.; TESOLIN, J. C. C.; CAVALCANTI, M. C. R. Minimanager: Uma proposta de plataforma web para emulação de redes heterogêneas de comunicação móvel. In: SBC. *Anais Estendidos do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.I.], 2022. p. 17–24.
- 41 BARONE, D. A. C.; WICKBOLDT, J. A.; CAVALCANTI, M. C. R.; MOURA, D. F. C.; TESOLIN, J. C. C.; DEMORI, A. M.; ANJOS, J. C. S. dos; CARVALHO, L. F. B. S. de; GOMES, J. E. C.; FREITAS, E. P. de. Integrating a multi-agent system simulator and a network emulator to realistically exercise military network scenarios. In: WAGNER, G.; WERNER, F.; RANGO, F. D. (Ed.). *Proceedings of the 13th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2023, Rome, Italy, July 12-14, 2023*. SCITEPRESS, 2023. p. 194–201. Disponível em: <<https://doi.org/10.5220/0012051600003546>>.
- 42 DEMORI, A. M.; TESOLIN, J. C. C.; MOURA, D. F. C.; GOMES, J. E. C.; PEDROSO, G.; CARVALHO, L. F. B. S. de; FREITAS, E. P. de; CAVALCANTI, M. C. R. A semantic web approach for military operation scenarios development for simulation. In: GUSIKHIN, O.; HAMMOUDI, S.; CUZZOCREA, A. (Ed.). *Proceedings of the 12th International Conference on Data Science, Technology and Applications, DATA 2023, Rome, Italy, July 11-13, 2023*. SCITEPRESS, 2023. p. 390–397. Disponível em: <<https://doi.org/10.5220/0012088600003541>>.
- 43 BARONE, D. A. C.; WICKBOLDT, J. A.; CAVALCANTI, M. C. R.; MOURA, D.; TESOLIN, J. C. C.; DEMORI, A. M.; CARVALHO, L. F. B. S. de; GOMES, J. E. C.; ANJOS, J. C. S. dos; FREITAS, E. P. de. Integrated multi-agent system simulator and network emulator framework to realistically exercise networked command and control application scenarios. In: WAGNER, G.; WERNER, F.; RANGO, F. D. (Ed.). *Simulation and Modeling Methodologies, Technologies and Applications*. Cham: Springer Nature Switzerland, 2024. p. 9–28. ISBN 978-3-031-77603-8.
- 44 JESUS, V. da Silva de. *AUTONOMIA DE MEIOS OPERATIVOS PARA APOIO À DECISÃO EM SISTEMAS DE COMANDO E CONTROLE: UMA ABORDAGEM PREDITIVA E CONCEITUAL UTILIZANDO APRENDIZADO DE MÁQUINA*. Dissertação (Mestrado) — UFRJ, 05 2025.

- 45 TESOLIN, J. C. C. *Towards A Mobile Wireless Network Ontology For Radio Access Points Selection Supported By Semantic Reasoning*. Tese (Doutorado) — Instituto Militar de Engenharia, 07 2024.
- 46 ALBUQUERQUE, P. P. L. D.; SILVA, R. R. M. R. D. *Um Simulador Para Redes Heterogêneas Sob O Paradigma "Always Best Connected*. 2021. Projeto de Final de Curso de Graduação IME - MiniManager. Disponível em: <[http://www.comp.ime.eb.br/graduacao/?pc=p\\_detp&num=484](http://www.comp.ime.eb.br/graduacao/?pc=p_detp&num=484)>.
- 47 TESOLIN, J.; DEMORI, A. M.; MOURA, D. F. C.; CAVALCANTI, M. C. Enhancing heterogeneous mobile network management based on a well-founded reference ontology. 17 2022. Disponível em: <<http://dx.doi.org/10.2139/ssrn.4368533>>.
- 48 VIGNOLI, L. E. *Abordagem para a modelagem e concepção de um Sistema de Sistemas: Um estudo de caso em Comando e Controle*. Tese (Doutorado) — Proposta em desenvolvimento - Instituto Militar de Engenharia, 03 2025.
- 49 AVELINO, J. O.; ROSA, G. F.; DANON, G. R.; CORDEIRO, K. F.; CAVALCANTI, M. C. Preanote: Uma abordagem de anotação de corpus para o ajuste fino de large language model pré-treinado. In: PIRES, C. E. S.; RAZENTE, H. L.; OGASAWARA, E. S. (Ed.). *Proceedings of the 39th Brazilian Symposium on Databases, SBBD 2024, Florianópolis, SC, Brazil, October 14-17, 2024*. SBC, 2024. p. 806–812. Disponível em: <<https://doi.org/10.5753/sbbd.2024.242494>>.
- 50 AVELINO, J. O.; ROSA, G. F.; DANON, G. R.; CORDEIRO, K. F.; CAVALCANTI, M. C. Knowledge graph generation from text using supervised approach supported by a relation metamodel: An application in C2 domain. In: FILIPE, J.; SMIALEK, M.; BRODSKY, A.; HAMMOUDI, S. (Ed.). *Proceedings of the 26th International Conference on Enterprise Information Systems, ICEIS 2024, Angers, France, April 28-30, 2024, Volume 1*. SCITEPRESS, 2024. p. 281–288. Disponível em: <<https://doi.org/10.5220/0012629300003690>>.
- 51 FERREIRA, F.; DUARTE, J.; UGULINO, W. Automated statistics extraction of public security events reported through microtexts on social networks. In: *Proceedings of the XVIII Brazilian Symposium on Information Systems*. New York, NY, USA: Association for Computing Machinery, 2022. (SBSI '22). ISBN 9781450396981. Disponível em: <<https://doi.org/10.1145/3535511.3535513>>.
- 52 MOSAFI, F. F. D. S.; PIRES, L. F.; DUARTE, J. C.; CAVALCANTI, M. C. Towards a microservices architecture to support communication in c2 applications. In: *2024 19th Annual System of Systems Engineering Conference (SoSE)*. [S.l.: s.n.], 2024. p. 227–232.
- 53 VANZAN, M.; DUARTE, J. Malware classification using transfer learning through the gpt-2 model. In: *Anais do XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Porto Alegre, RS, Brasil: SBC, 2023. p. 167–180. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/sbseg/article/view/27205>>.
- 54 SILVA, C.; MOREIRA, T.; FERNANDES, I.; PASSOS, C.; DUARTE, J.; GOLDSCHMIDT, R. Sistemas tutores inteligentes na aprendizagem por competências: Uma revisão sistemática da literatura. In: *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2023. p. 1120–1132. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/sbie/article/view/26740>>.

- 55 OLIVEIRA, R. L. de; DUARTE, J. C.; CORDEIRO, K. de F. Machine learning model explainability supported by data explainability: a provenance-based approach. *Journal of Information and Data Management*, v. 15, n. 1, p. 93–102, Feb. 2024. Disponível em: <<https://journals-sol.sbc.org.br/index.php/jidm/article/view/3337>>.
- 56 RAMELLA, I.; JESUS, V. da Silva de. Integração de sistemas heterogêneos com mdsp: Uma avaliação arquitetural. In: *Anais do 57º Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2025.
- 57 S2C2, E. *Ontologias de Referência*. 2025. Artefatos resultantes de modelagem conceitual dirigida a ontologias. Disponível em: <<https://comp-ime-eb-br.github.io/S2C2-IME/ontodesc/>>.
- 58 DEMORI, A. M.; MOURA, D. F. C.; CAVALCANTI, M. C. R. *Ontologia Operacional MISCON*. 2024. Artefato resultante da operacionalização da ontologia de referência MISCON. Disponível em: <<https://github.com/comp-ime-eb-br/S2C2-IME/blob/main/onto/miscon.ttl>>.
- 59 TESOLIN, J. C. C.; MOURA, D. F. C.; CAVALCANTI, M. C. R. *Ontologia Operacional HINT*. 2024. Artefato resultante da operacionalização da ontologia de referência HINT. Disponível em: <<https://github.com/comp-ime-eb-br/S2C2-IME/blob/main/onto/hint.ttl>>.
- 60 TESOLIN, J. C. C.; MOURA, D. F. C.; CAVALCANTI, M. C. R. *Ontologia Operacional QVAS*. 2024. Artefato resultante da operacionalização da ontologia de referência QVAS. Disponível em: <<https://github.com/comp-ime-eb-br/S2C2-IME/blob/main/onto/qvas.ttl>>.
- 61 DEMORI, A. M.; MOURA, D. F. C.; CAVALCANTI, M. C. R. *Ontologia Operacional MISCON*. 2024. Registro da ontologia operacional MISCON. Disponível em: <[https://revistas.inpi.gov.br/pdf/Programa\\_de\\_computador2810.pdf](https://revistas.inpi.gov.br/pdf/Programa_de_computador2810.pdf)>.
- 62 TESOLIN, J. C. C.; MOURA, D. F. C.; CAVALCANTI, M. C. R. *Registro da Ontologia Operacional HINT*. 2024. Registro da ontologia operacional HINT. Disponível em: <[https://revistas.inpi.gov.br/pdf/Programa\\_de\\_computador2809.pdf](https://revistas.inpi.gov.br/pdf/Programa_de_computador2809.pdf)>.
- 63 AVELINO, J. de O.; CORDEIRO, K. de F.; CAVALCANTI, M. C. R. *Registro do Software PREAnoTeTool*. 2024. Registro do Software PREAnoTeTool. Disponível em: <[https://revistas.inpi.gov.br/pdf/Programa\\_de\\_computador2832.pdf](https://revistas.inpi.gov.br/pdf/Programa_de_computador2832.pdf)>.