

# **Uma Abordagem Baseada em Cenários para Extração de Requisitos de Tolerância a Falhas de Sistemas-de-Sistemas a partir de Processos-de-Processos de Negócios**

**Sidny de Almeida Molina Pereira<sup>1</sup> e Maria Istela Cagnin (orientadora)<sup>1</sup>**

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Universidade Federal de Mato Grosso do Sul (UFMS)

Cidade Universitária, Campo Grande, MS – Brasil

sidny.molina@gmail.com, istela.machado@ufms.br

**Resumo.** *Falhas podem ocorrer durante a interoperabilidade de Sistemas-de-Sistemas (SoS) afetando o seu funcionamento e confiabilidade. Este trabalho define uma abordagem baseada em cenários na notação BPMN (Business Process Model and Notation) para extrair, de maneira sistemática e automática, requisitos de tolerância a falhas de SoS durante a interoperabilidade entre os seus sistemas constituintes, a partir de informação útil de Processos-de-Processos de Negócio (PoP) de alianças de organizações. Com isso, é possível alcançar alinhamento entre os níveis técnico e de negócio, podendo favorecer a competitividade e lucratividade de alianças de organizações.*

## **1. Visão Geral do Tema**

Sistemas de Sistemas (do inglês, *Systems-of-Systems* ou SoS) referem-se a um conjunto de sistemas de software individuais (denominados sistemas constituintes) que interoperaem entre eles para trocar informações e oferecer funcionalidades e capacidades não presentes em um único sistema de software [Maier 1998]. SoS são amplamente aplicados em áreas como cidades inteligentes, saúde, agronegócio, e possuem características únicas como comportamento emergente, desenvolvimento evolucionário, e independência operacional e gerencial dos seus sistemas constituintes [Maier 1998, Dersin 2014]. No contexto contemporâneo de negócios, os sistemas constituintes de um SoS podem pertencer a organizações distintas com seus próprios processos de negócios. Quando organizações se unem em alianças (como parcerias, fusões e aquisições), os seus processos de negócio formam processos de negócios complexos, normalmente automatizados pelo SoS e denominados Processos-de-Processos de Negócio (do inglês, *Processes-of-Business Processes* ou PoP) [Cagnin and Nakagawa 2021].

Nesse contexto, SoS adapta-se continuamente para atender as missões (ou seja, os objetivos estratégicos de negócio) das alianças de organizações, cujos processos de negócio são representados por meio do PoP. Contudo, falhas podem ocorrer durante a interoperabilidade entre os sistemas constituintes [Ferreira et al. 2021] ou nos próprios constituintes, em razão da reorganização dinâmica do SoS quando esses sistemas são alterados ou removidos [Batista 2013]. Essas modificações, resultantes de ajustes dos negócios das alianças, impactam diretamente a confiabilidade do SoS como um todo. Assim, para aumentar a confiabilidade e a estabilidade do SoS, é crucial não apenas compreender as falhas ou comportamentos inesperados que podem afetar suas operações [Ncube and Lim 2018], mas também assegurar que quaisquer situações

adversas ocorridas durante a interoperabilidade sejam tratadas por meio de requisitos de tolerância a falhas, a fim de manter a estabilidade e a confiabilidade do SoS [Cook 2008, Sanduka and Obermaisser 2015].

Embora os requisitos de confiabilidade, como de tolerância a falhas, sejam fundamentais, eles ainda são pouco explorados na literatura [Ferreira et al. 2021, Molina et al. 2022]. Os trabalhos existentes geralmente consideram esses requisitos apenas durante o *design* arquitetural, negligenciando a importância de tratá-los ao longo de todo o ciclo de vida do SoS, bem como o alinhamento entre os níveis técnico e de negócio [Molina et al. 2022], visto que informações de modelos de negócio são fontes valiosas para a extração de requisitos funcionais [Nogueira and De Oliveira 2017] e não funcionais [Mohammadi and Heisel 2016]. Ademais, a Engenharia de Requisitos em SoS enfrenta desafios complexos devido a múltiplas interações entre sistemas individuais, diferenças de domínio, ambiguidades conceituais e necessidade de adaptações rápidas [Lewis et al. 2009].

Portanto, é crucial adotar técnicas de gerenciamento de requisitos durante o ciclo de vida dos SoS [Ncube and Lim 2018] considerando o negócio das organizações envolvidas. Para isso, é primordial iniciar pelas etapas de extração e especificação de requisitos, como é caso de requisitos de tolerância a falhas durante a interoperabilidade para suprir os desafios inerentes a eles, propiciando o alinhamento entre os níveis técnico e de negócio [Pereira 2024]. Salienta-se que os desafios mencionados também são aplicáveis no contexto de SoIS (*Systems-of-Information Systems*), que é uma categoria de SoS composto por um ou mais sistemas de informação (SI) intensivos em software, inter-relacionados de forma dinâmica para coletar, armazenar e processar dados, fornecendo informações e conhecimento que apoiam tomadas de decisões organizacionais [Neto et al. 2016]. Por fim, o gerenciamento de requisitos de tolerância a falhas durante a interoperabilidade de SoS e de SoIS, alinhado ao negócio de alianças de organizações, colabora com a resolução de um dos *Grandes Desafios de Pesquisa em Sistemas de Informação no Brasil para 2016 a 2026*, cujo objetivo é alcançar a interoperabilidade plena, desde o nível organizacional até o nível técnico.

## 2. Estado da Arte e Trabalhos Relacionados

Para descobrir como requisitos de tolerância a falhas de SoS têm sido extraídos, especificados e validados, conduzimos um mapeamento sistemático da literatura (MSL) [Molina et al. 2022] seguindo o processo estabelecido em [Kitchenham and Charters 2007] e visando responder a seguinte questão de pesquisa: **“Como os requisitos de confiabilidade de SoS estão sendo elicitados, especificados e validados?”**.

Foram realizadas buscas no período de 27/09/2021 a 17/11/2021 em cada biblioteca digital selecionada (IEEE Xplore, ACM e Scopus), adaptando a *string* de busca padrão definida. As buscas retornaram um total de 603 estudos e, após a remoção dos estudos duplicados e daqueles eliminados por critérios de exclusão nas etapas de seleção, foram selecionados 12 estudos primários [Prasad and Milks 2008, Despotou and Kelly 2009, Andrews et al. 2013, Garro and Tundis 2014, Ingram et al. 2014, Tianjian and Xin 2014, Damm and Vincentelli 2015, El Hachem et al. 2016, Rambikur et al. 2017,

Van Bossuyt et al. 2018, Hyun et al. 2020, Imamura et al. 2021].

A partir deste MSL constatou-se que os requisitos de confiabilidade estão focados principalmente em tolerância a falhas e recuperabilidade, sendo identificados e especificados após a fase de requisitos (ou seja, no *design* arquitetural), negligenciando a importância do tratamento de requisitos ao longo de todo o ciclo de vida do SoS, bem como o alinhamento entre os níveis técnico e de negócio. Devido à preocupação desse tipo de requisitos apenas no *design* arquitetural, as especificações são predominantemente baseadas em modelos e diagramas, como os da SysML (*Systems Modeling Language*). A identificação dos requisitos de confiabilidade é geralmente centrada em comportamentos inesperados que podem ocorrer durante a comunicação entre os constituintes [Tianjian and Xin 2014, Van Bossuyt et al. 2018, Hyun et al. 2020], devido à importância de garantir a interoperabilidade entre os sistemas para alcançar as missões do SoS, que ainda é um desafio nas comunidades de SoS e SoIS. Os resultados também indicaram que nenhum dos estudos primários selecionados considera os complexos e dinâmicos processos de negócios das alianças de organizações (PoP) como fonte para extrair os requisitos de confiabilidade de SoS. No entanto, é importante abordar potenciais falhas desde o nível de processos de negócio, como comportamentos imprevistos ou interrupções inesperadas, que podem afetar a execução e a estabilidade do negócio. Utilizando essa fonte de informação, é possível alinhar os níveis técnico e de negócio, podendo auxiliar as alianças a alcançarem seus objetivos estratégicos e aumentar a sua eficiência.

Trabalhos mais recentes no contexto de Engenharia de Requisitos de SoS foram encontrados, porém não consideram o nível de negócio. [Svenson et al. 2022] investigam a Engenharia de Requisitos voltada para a qualidade no desenvolvimento de SoS, destacando a necessidade de serviços mediadores e acordos entre desenvolvedores de sistemas constituintes para assegurar a qualidade na troca de dados entre esses sistemas. Em [Laliberte et al. 2022], são apresentados e testados métodos automatizados de rastreabilidade baseados em técnicas de Processamento de Linguagem Natural (PLN), para garantir a precisão dos links entre os requisitos não funcionais de um SoS e seus constituintes, prevenindo problemas comuns de rastreabilidade.

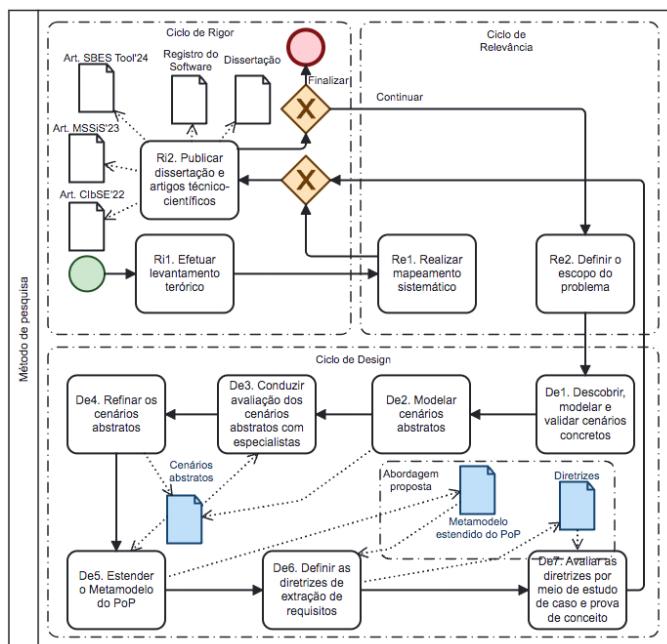
### **3. Questão de Pesquisa e Objetivos**

Após a condução do MSL, observou-se uma lacuna de estudos que explorem sistematicamente a extração de requisitos de tolerância a falhas de SoS a partir de modelos de processos de negócio complexos e dinâmicos, como o PoP. Diante dos desafios da Engenharia de Requisitos de SoS e da importância de identificar corretamente os requisitos de tolerância a falhas para manter a estabilidade do SoS, ao mesmo tempo em que contribuem para atingir os objetivos estratégicos das alianças de organizações, definimos a seguinte questão de pesquisa [Pereira 2024]: “*Como facilitar a extração sistemática de requisitos de tolerância a falhas de SoS, com base em informações relevantes do nível de negócio de alianças, para assegurar a estabilidade dos sistemas constituintes durante a interoperabilidade?*”. Para responder essa questão, o objetivo geral deste trabalho foi definir uma abordagem que facilite a extração sistemática e automatizada de requisitos de tolerância a falhas de SoS, usando informações sobre o tratamento de exceções na comunicação entre processos constituintes do PoP [Pereira 2024]. Essa proposta visa alinhar os níveis técnico e de negócio. Para alcançar o objetivo geral, três objetivos específicos foram estabelecidos: (1) Identificar os elementos da notação BPMN (*Business*

*Process Model and Notation*) [Object Management Group 2013], adotada para modelar o PoP [Cagnin and Nakagawa 2021], que podem ser comumente utilizados para representar o tratamento de exceções na comunicação entre processos de negócio do PoP; (2) Facilitar a extração sistemática de requisitos de tolerância a falhas de SoS em fases recentes do ciclo de vida do SoS a partir de informações do nível de negócio, levando em considerações os elementos BPMN identificados no objetivo anterior; e (3) Garantir que os requisitos de tolerância a falhas extraídos sejam completos e alinhados ao negócio com o intuito de manter a estabilidade do SoS.

#### 4. Método de Pesquisa

A metodologia adotada neste trabalho é qualitativa, exploratória e baseada na *Design Science Research* (DSR), que foca em resolver problemas práticos em contextos específicos por meio de artefatos, gerando novos conhecimentos científicos [Wieringa 2014]. A DSR envolve três ciclos principais: (i) Ciclo de Relevância, (ii) Ciclo de *Design*, e (iii) Ciclo de Rigor. A Figura 1 ilustra esses ciclos no contexto do trabalho desenvolvido e as respectivas etapas realizadas, bem com os artefatos desenvolvidos (destacados em azul). Os resultados da etapa *Ri1* são mencionados na Seção 1. A etapa *Re1* é descrita sucintamente na Seção 2, enquanto a etapa *Re2* é apresentada na Seção 3. Os principais resultados das etapas *De1* a *De4* são descritos na Seção 5. A Seção 6 apresenta os principais resultados da etapa *De5* e a Seção 7 descreve os resultados das etapas *De6* e *De7*. Por fim, a Seção 8 apresenta os resultados da etapa *Ri2*.



**Figura 1. Ciclos e as respectivas etapas e artefatos do método de pesquisa adotado (adaptado de [Pereira 2024])**

#### 5. Cenários Abstratos de Tratamento de Exceções na Interoperabilidade de PoP

Para atender o primeiro objetivo específico do trabalho (Seção 3), foram desenvolvidos dois cenários abstratos em BPMN. Esses cenários foram criados para representar o trata-

mento de exceções, abrangendo o envio de requisições (ou mensagens) e o recebimento de suas respostas, com o intuito de ilustrar situações comuns de exceções na interoperabilidade entre processos constituintes distintos que compõem o PoP [Molina et al. 2023]. A finalidade é utilizar esses cenários abstratos como base para facilitar a modelagem de PoP reais.

A elaboração dos cenários abstratos seguiu exemplos de elementos relacionados ao tratamento de exceções da especificação BPMN, considerando o contexto de um PoP dirigido (ou seja, processos constituintes automatizados por sistemas que integram o SoS são coordenados por uma autoridade central, ou seja, um processo dominante responsável por alcançar os objetivos do PoP e, consequentemente, do SoS). Além disso, os cenários abstratos foram inspirados no conhecimento do grupo de pesquisa dos autores ao modelar cenários concretos de PoP reais em diversos domínios, como educação (PoP de repositório institucional da UFMS), agronegócio (PoP de monitoramento de produtividade e bem-estar animal da Embrapa Gado de Corte), saúde (PoP de saúde pública [Cagnin and Nakagawa 2021]) e de emergência (resgate) [Andrews et al. 2013]. Esse último cenário concreto foi abstraído a partir de situações de falhas de um SoS [Andrews et al. 2013], que automatiza o PoP correspondente.

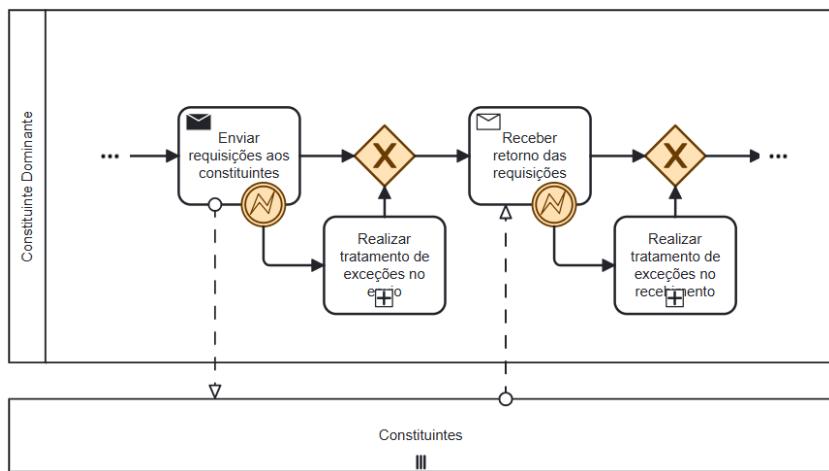
Os cenários abstratos foram avaliados por sete especialistas em BPMN, por meio de um questionário<sup>1</sup> [Molina et al. 2023]. Os resultados permitiram refinar os elementos BPMN utilizados para representar de maneira adequada o tratamento de exceções no envio e no recebimento de retorno de mensagens no contexto de PoP, oferecendo *insights* valiosos para aprimorar o tratamento de exceções no contexto da interoperabilidade entre processos constituintes.

Para representar falhas em tarefas de envio e de recebimento de mensagens nos cenários abstratos refinados (Figura 2), utiliza-se o elemento BPMN de evento intermediário de erro (raio em amarelo) anexado na borda dessas tarefas. Além disso, os cenários propõem dois subprocessos: um para “Realizar tratamento de exceções no envio” de requisições (Figura 3) e outro para “Realizar tratamento de exceções no recebimento” de retorno de requisições [Molina et al. 2023]. Cada subprocesso está vinculado ao fluxo de sequência que sai de cada evento intermediário de erro, responsável pelo tratamento das exceções, conforme ilustrado na Figura 2.

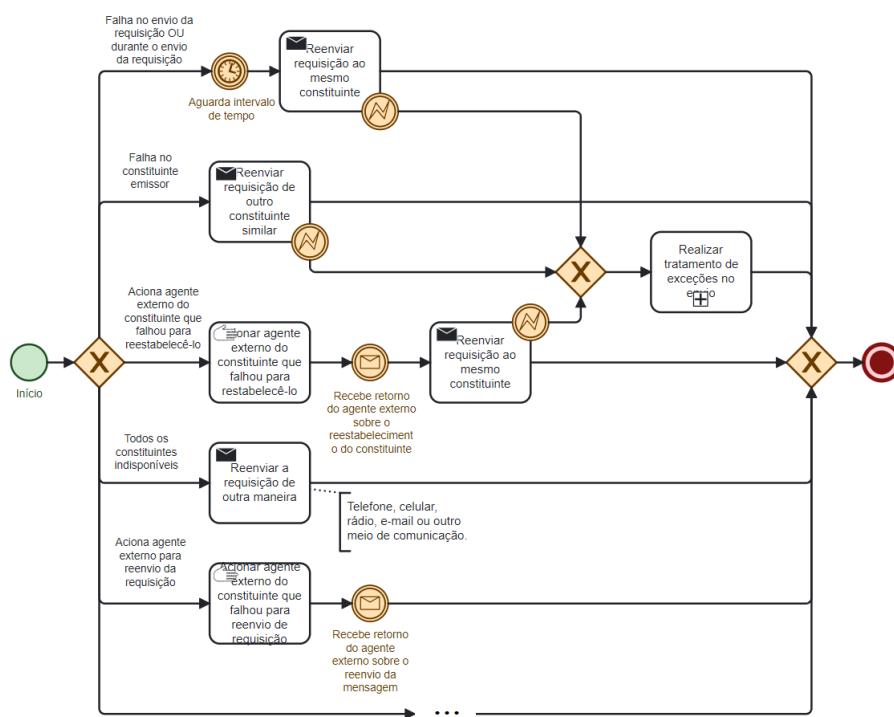
## 5.1. Elementos BPMN utilizados para representar tratamento de exceções em PoP

Após a avaliação e os refinamentos dos cenários abstratos [Molina et al. 2023], foi possível identificar os elementos BPMN mais apropriados para representar as falhas comuns durante a interoperabilidade em PoP e seus respectivos tratamentos de exceções, sendo eles: (i) tarefas que realizam o envio de requisições (`sendTask`); (ii) tarefas que realizam o recebimento de retorno de requisições (`receiveTask`) por meio de fluxos de mensagens (`messageFlow`) que indicam a interoperabilidade entre processos de negócio; (iii) evento intermediário de erro (`errorEventDefinition`); (iv) elementos dos subprocessos dos cenários: evento de ínicio (`startEvent`), desvio exclusivo (`exclusiveGateway`), fluxo de sequência (que representa cada possível falha) que sai do desvio exclusivo (`sequenceFlow`) e os demais elementos que são considerados para contornar as falhas, ou seja, evento intermediário temporal

<sup>1</sup>Disponível em: <https://link.ufms.br/DyYc7>



**Figura 2. Exemplo de interoperabilidade entre processos constituintes durante o envio de requisições e recebimento de retorno de requisições [Molina et al. 2023]**



**Figura 3. Cenário abstrato do subprocesso “Realizar tratamento de exceções no envio” [Molina et al. 2023]**

(timerEventDefinition), tarefa de envio (sendTask), tarefa de recebimento (receiveTask), tarefa manual (manualTask) e evento intermediário de recebimento de mensagem do tipo ‘captura’ (intermediateCatchEvent). Por fim, o elemento de fim (endEvent) do tipo ‘terminate’ (terminateEventDefinition) especificado no final de ambos cenários abstratos.

## 6. Metamodelo estendido do PoP

Para atender o segundo objetivo específico do trabalho (Seção 3), que se refere à definição da abordagem proposta, primeiramente, o metamodelo do PoP [Cagnin and Nakagawa 2021] foi estendido com base nos elementos BPMN relacionados à falhas e seus respectivos tratamentos de exceções em PoP direcionados (Seção 5.1). Essa extensão teve como intuito documentar a fonte de informação para realizar a extração sistemática dos requisitos de tolerância a falhas de SoS a partir de PoP, por meio da abordagem proposta. Originalmente, o metamodelo do PoP foi definido para facilitar o entendimento dos diversos elementos e conceitos envolvidos em PoP [Cagnin and Nakagawa 2021].

A extensão do metamodelo foi realizada em dois passos principais, sendo eles: (i) identificação de elementos BPMN; e (ii) incorporação desses elementos ao metamodelo. No **primeiro passo** foram analisados os arquivos XML gerados pela ferramenta bpmn.io, que é *opensource* e segue rigorosamente a especificação BPMN, para identificar as classes correspondentes aos elementos BPMN utilizados nos cenários abstratos (Seção 5.1). No **segundo passo**, as classes BPMN identificadas foram incorporadas ao metamodelo do PoP, representando seus relacionamentos de dependência com classes já existentes no modelo. Esses relacionamentos indicam que uma classe do PoP é representada por uma classe BPMN, reutilizando a semântica da sua especificação. Os elementos foram organizados em pacotes temáticos (*Gateway*, *Event* e *Task*), para melhorar a organização e legibilidade do modelo. A interoperabilidade entre processos constituintes (*Constituent-Process*) é representada por fluxos de mensagem (*messageFlow*), e cada processo constituinte é modelado como um participante (*participant*) associado a um processo público (*PublicCP*), conforme apresentado na Figura 4 e destacado em laranja.

As falhas podem estar relacionadas à fluxos de mensagens (*messageFlow*) que indicam a interoperabilidade entre processos de negócio, conectando tarefas de envio (*sendTask*) e recebimento (*receiveTask*) do pacote *Task*. O evento intermediário de erro (*errorEventDefinition*<sup>2</sup>) é anexado a essas tarefas para identificar falhas, enquanto os elementos dos subprocessos correspondentes aos cenários abstratos tratam as exceções no envio e recebimento. Esses elementos incluem: evento de início (*startEvent*<sup>1</sup>), desvio exclusivo (*exclusiveGateway*<sup>3</sup>), fluxo de sequência (*sequenceFlow*), que representa possíveis falhas, e elementos para contorná-las, como eventos temporais (*timerEventDefinition*<sup>1</sup>), tarefas manuais (*manualTask*<sup>4</sup>) e eventos intermediários de captura de mensagem (*intermediateCatchEvent*<sup>1</sup>). Detalhes dos elementos BPMN incorporados e das classes dos pacotes *Gateway*, *Event* e *Task* do metamodelo estendido do PoP são obtidos em [Pereira 2024].

## 7. Diretrizes para Extração dos Requisitos

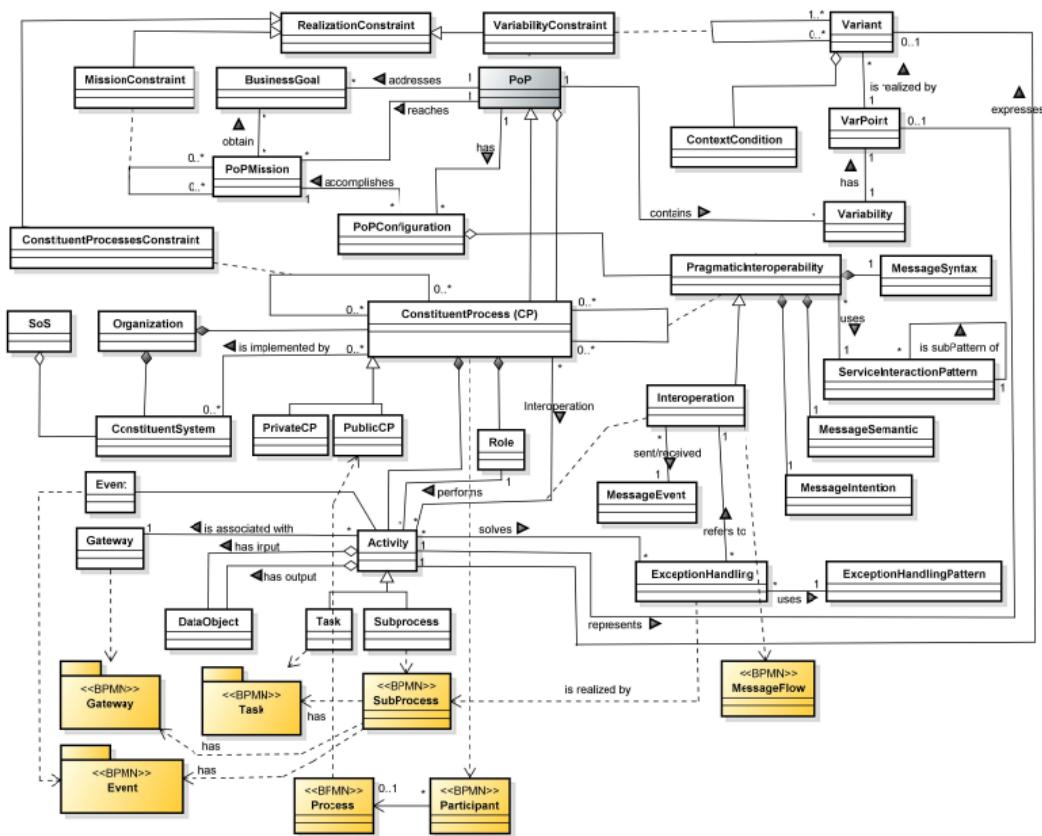
Para compor a abordagem proposta, além do metamodelo estendido, também foram elaboradas diretrizes para extração de requisitos de tolerância a falhas de SoS durante a interoperabilidade. Essas diretrizes foram motivadas pelo estudo [Cagnin and Nakagawa 2024], no qual os autores apresentam o PoP como fonte de

---

<sup>2</sup>Pacote *Event*

<sup>3</sup>Pacote *Gateway*

<sup>4</sup>Pacote *Task*



**Figura 4. Metamodelo estendido do PoP [Pereira 2024]**

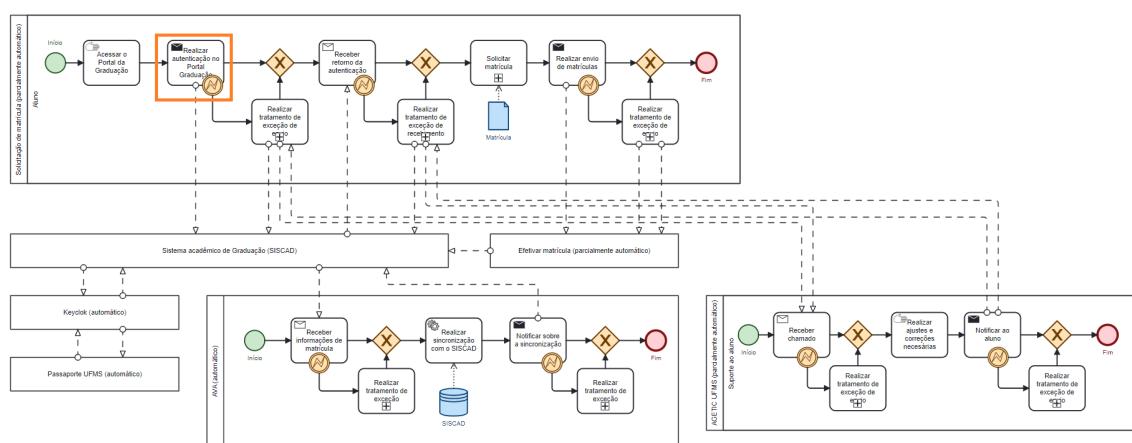
informação valiosa para identificar três classes de requisitos de SoS (funcionais, não funcionais e de comportamentos emergentes de SoS) a partir de elementos do metamodelo de PoP [Cagnin and Nakagawa 2022]; mas, sem considerar elementos específicos da notação BPMN utilizada para representar os modelos de PoP. Como benefícios de extrair requisitos de SoS a partir do PoP, têm-se o alinhamento dos requisitos às necessidades organizacionais e o mapeamento direto entre os elementos do PoP e os elementos que compõem a especificação dos requisitos de SoS [Cagnin and Nakagawa 2024]. Diante disso, observou-se a importância de definir as diretrizes de extração a partir de elementos da notação utilizada na modelagem de PoP, em particular o *Modelo Detalhado de Missão do PoP*, com o intuito de sistematizar o processo de extração dos requisitos e potencializar a sua automação. Esse modelo é um dos modelos utilizados para modelar PoP [Cagnin and Nakagawa 2022] e representa detalhadamente todos os processos constituintes necessários para atingir uma determinada missão do PoP, bem como a interoperabilidade entre eles.

Sucintamente, as diretrizes foram definidas a partir da execução de quatro passos: (i) identificação das fontes de informação dos requisitos a partir da análise dos elementos BPMN utilizados nos cenários abstratos (Seção 5.1) e representados no metamodelo estendido do PoP, resultando na Tabela 1; (ii) definição de campos relevantes para especificar requisitos, combinando elementos reutilizados de [Cagnin and Nakagawa 2024], adaptados e criados especificamente para o contexto de requisitos de tolerância a falhas durante a interoperabilidade (ver Tabelas 10 e 11 de [Pereira 2024]); (iii) elaboração de

*templates* textuais para documentar os requisitos extraídos (apresentados nas Tabelas 2 e 3), utilizando variáveis extraídas dos elementos BPMN (primeira coluna da Tabela 1); e (iv) criação do algoritmo para sistematizar a extração e especificação dos requisitos. Como premissa para utilizar o algoritmo é necessário que o *Modelo Detalhado de Missão do PoP* esteja em conformidade com os cenários abstratos (Seção 5). Basicamente, o algoritmo percorre cada *Modelo Detalhado de Missão do PoP* e elicta os requisitos a partir de informação dos elementos BPMN envolvidos em cada troca de mensagem entre dois constituintes que possuem tarefa de envio ou recebimento com evento de erro na borda e com subprocesso associado, com base nas Tabelas 1, 2 e 3. Em paralelo, os conteúdos dos campos dos *templates* são registrados, à medida que são identificados. Ressalta-se que durante a extração manual é essencial que o engenheiro de requisitos de SoS atribua um identificador único para o “ID” do requisito e indique o nome do SoS em questão como conteúdo do campo “Sujeito”.

## 7.1. Estudo de Caso

Para avaliar as diretrizes de extração de requisitos definidas, primeiramente foi realizado um estudo de caso descritivo [Wholin et al. 2000], considerando evidências obtidas a partir das especificações textuais dos requisitos extraídos. O estudo de caso teve como **objetivo** analisar as diretrizes definidas, com o **propósito** de observar a sua aplicação **com respeito ao** apoio na extração sistemática de requisitos de tolerância a falhas **no contexto de** um SoS real a partir de um *Modelo Detalhado de Missão* (em particular, da missão “Adicionar turmas e os respectivos alunos matriculados no Ambiente Virtual de Aprendizagem (AVA)”) de um PoP dirigido (no caso, PoP Educacional-UFMS) **sob o ponto de vista de** um especialista em BPMN. A Tabela 4 (colunas 1 e 2) apresenta os requisitos extraídos e especificados manualmente (com apoio da abordagem definida) do subprocesso de tratamento de exceções no envio associado à tarefa de envio “Realizar autenticação no Portal Graduação”, destacada em laranja na Figura 5. O subprocesso associado ao evento de erro dessa tarefa é apresentado na Figura 6.



**Figura 5. Modelo Detalhado de Missão “Adicionar turmas e os respectivos alunos matriculados no Ambiente Virtual de Aprendizagem (AVA)” [Pereira 2024]**

Devido à limitação de espaço, mais informações sobre os requisitos extraídos e especificados manualmente, bem como sobre as ameaças à validade associadas ao estudo de caso são obtidas em [Pereira 2024].

**Tabela 1. Elementos BPMN para extração dos requisitos [Pereira 2024]**

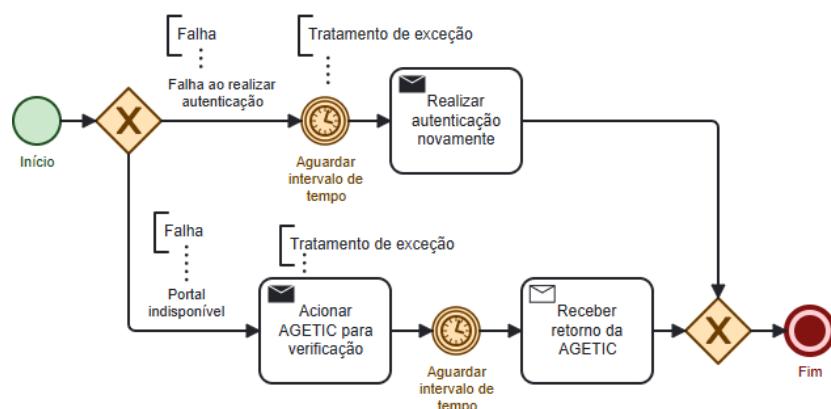
Variáveis	Elementos BPMN	Fonte de informação
Constituinte_Origem	(Participante/Lane) participant: id, name, processRef	Piscina que dá origem ao envio da mensagem
Constituinte_Destino	(Participante/Lane) participant: id, name, processRef	Piscina que dá destino da mensagem
Tarefa_envio_msg_com_ev_erro	(Tarefa de envio) sendTask: id, name + boundaryEvent: id, attachedToRef (refere-se ao ID da sendTask) + errorEventDefinition: id, outgoing	Tarefa que realiza o envio de mensagem
Tarefa_recebimento_msg_com_ev_erro	(Tarefa de recebimento) receiveTask: id, name + boundaryEvent: id, attachedToRef (refere-se ao ID da receiveTask) + errorEventDefinition: id, outgoing	Tarefa que realiza o recebimento de mensagem
Ev_interm_envio_msg	(Evento intermediário de envio) intermediateThrowEvent: id, name, incoming, outgoing + messageEventDefinition: id	Evento intermediário que realiza o envio de mensagem
Ev_interm_receb_msg	(Evento intermediário de recebimento) intermediateCatchEvent: id, name, incoming, outgoing + messageEventDefinition: id	Evento intermediário que realiza o recebimento de mensagem
Ev_interm_temporal	(Evento intermediário temporal de interromper) intermediateCatchEvent: id, name, incoming, outgoing + timerEventDefinition: id	Evento intermediário temporal do tipo que interrompe o envio ou o recebimento de mensagem conforme descrição
Realizar_tratamento_de_exceção_de_envio	(Subprocesso) subProcess: id, name, incoming, outgoing	Atividade que irá realizar o tratamento de exceção da tarefa de envio de mensagem que falhou
Momento_Falha_Envio	Composta pela tarefa de envio (sendTask = id, name) com o evento intermediário de erro (errorEventDefinition) anexo em sua borda (boundaryEvent)	Observada pela tarefa de envio com evento de erro anexado na borda da tarefa
Falha_envio	Representada no rótulo do primeiro fluxo de sequência (sequenceFlow = id, name) que sairá do desvio exclusivo (exclusiveGateway) associado (sourceRef e targetRef)	Representada no rótulo do primeiro fluxo de sequência que sairá do desvio exclusivo associado
Solução_falha_envio	Composta por evento intermediário de tempo (intemediateCatchEvent = id, name), tarefa de envio (sendTask = id, name), tarefa de recebimento (receiveTask = id, name), tarefa de serviço (serviceTask = id, name), tarefa manual (manualTask: id), desvio exclusivo (exclusiveGateway: id)	Composta por todos os elementos identificados após cada falha até o final do segundo desvio exclusivo final (junção XOR) do tratamento de exceções no envio
Momento_Falha_Recebimento	Composta pela tarefa de recebimento (receiveTask = id, name) com o evento intermediário de erro (errorEventDefinition) anexo em sua borda (boundaryEvent)	Observada pela tarefa de recebimento com evento de erro anexo na borda da tarefa
Realizar_tratamento_de_exceção_de_recebimento	(Subprocesso) subProcess: id, name, incoming, outgoing	Atividade que irá realizar o tratamento de exceção da tarefa de recebimento de mensagem que falhou
Falha_recebimento	Representada no rótulo do primeiro fluxo de sequência (sequenceFlow = id, name) que sairá do desvio exclusivo exclusiveGateway associado (sourceRef e targetRef)	Representada no rótulo do primeiro fluxo de sequência que sairá do desvio exclusivo associado
Solução_falha_recebimento	Composta por evento intermediário de tempo (intemediateCatchEvent = id, name), tarefa de envio (sendTask = id, name), tarefa de recebimento (receiveTask = id, name), tarefa de serviço (serviceTask = id, name), tarefa manual (manualTask: id), desvio exclusivo (exclusiveGateway: id)	Composta por todos os elementos identificados após cada falha até o final do segundo desvio exclusivo final (junção XOR) do tratamento de exceções no recebimento
Rotulo_fluxo_sequencia	(Fluxo de sequência) sequenceFlow: id, sourceRef, targetRef	Fluxo de sequência saindo de um gateway ou de uma atividade
Desvio.exclusivo	(Desvio) exclusiveGateway: id	Desvio de sequência a ser percorrido

**Tabela 2. Template detalhado para especificação dos requisitos de tolerância a falha no envio [Pereira 2024]**

Campo	Conteúdo
ID	ID da tarefa de envio com falha
Classe	Tolerância a falhas
Sujeito	[Sujeito do requisito, no caso é o próprio SoS]
Constituinte de origem	constituinte_origem
Constituinte de destino	constituinte_destino
Momento para ocorrência da falha durante o envio da mensagem	momento_falha_envio é observado pela tarefa_envio_msg_com_ev_erro
Quais falhas que ocorrem durante o envio da mensagem	Cada falha_envio é observada por cada rotulo_fluxo_sequencia que sai do desvio_exclusivo do subprocesso cujo rótulo é “Realizar_Tratamento_de_Exceção_de_Envio”
Como resolver as falhas durante o envio da mensagem	Cada solução_falha_envio é observada por cada ev_interm_temporal, por cada ev_interm_envio_msg e por cada task após cada falha_envio
Ação	Durante o envio de mensagem do constituinte_origem para o constituinte_destino, quando momento_falha_envio *[ao ocorrer falha_envio, então solução_falha_envio]
Rastreabilidade	Participant constituinte_origem para Participant constituinte_destino na SendTask tarefa_envio_msg_com.ev_erro

**Tabela 3. Template detalhado para especificação dos requisitos de tolerância a falha no recebimento [Pereira 2024]**

Campo	Conteúdo
ID	ID da tarefa de recebimento com falha
Classe	Tolerância a falhas
Sujeito	[Sujeito do requisito, no caso é o próprio SoS]
Constituinte de origem	constituinte_origem
Constituinte de destino	constituinte_destino
Momento para ocorrência da falha durante o recebimento da mensagem	momento_falha_recebimento é observada pela tarefa_recebimento_msg_com_ev_erro
Quais falhas que ocorrem durante o recebimento da mensagem	Cada falha_recebimento é observada por cada rotulo_fluxo_sequencia que sai do desvio_exclusivo do subprocesso cujo rótulo é “Realizar_Tratamento_de_Exceção_de_Recebimento”
Como resolver as falhas durante o recebimento da mensagem	Cada solução_falha_recebimento é observada por ev_interm_temporal, ev_interm_receb_msg e por cada task após cada falha_recebimento
Ação	Durante o recebimento de mensagem do constituinte_origem para o constituinte_destino, quando momento_falha_recebimento *[ao ocorrer falha_recebimento, então solução_falha_recebimento]
Rastreabilidade	Participant constituinte_origem para Participant constituinte_destino na ReceiveTask tarefa_recebimento_msg_com.ev_erro



**Figura 6. Subprocesso de tratamento de exceções no envio associado à tarefa “Realizar autenticação no Portal Graduação” [Pereira 2024]**

## 7.2. Prova de Conceito

As diretrizes de extração estabelecidas também foram avaliadas por meio de uma prova de conceito. Para isso, foi desenvolvido um módulo chamado PoP-ARE [Costa et al. 2024b]

**Tabela 4. Comparação da especificação dos requisitos de tolerância a falha extraído manualmente e automaticamente da tarefa de envio “Realizar autenticação no Portal Graduação” adaptado de [Pereira 2024]**

Conteúdo dos requisitos extraídos		
Campo	Extração Manual	Extração Automática
ID	01	Activity_0t3g3fe
Classe	Tolerância a falhas	Tolerância a falhas
Sujeito	SoS Educacional-UFMS	SoS Educacional-UFMS
Constituinte de origem	Solicitação de matrícula (parcialmente automático)	Solicitação de matrícula (parcialmente automático)
Constituinte de destino	Sistema Acadêmico de Graduação (SISCAD) (automático)	Sistema Acadêmico de Graduação (SISCAD) (automático)
Momento para ocorrência da falha durante o envio da mensagem	Ao “Realizar autenticação no Portal Graduação”	Ao “Realizar autenticação no Portal Graduação”
Quais falhas que ocorrem durante o envio da mensagem	“Falha ao realizar autenticação” e “Portal indisponível”	“Falha ao realizar autenticação” e “Portal indisponível”
Como resolver as falhas durante o envio da mensagem	Para “Falha ao realizar autenticação”, então “Aguardar intervalo de tempo” e “Realizar autenticação novamente”. Para “Portal indisponível”, então “Acionar AGETIC para verificação”, “Aguardar intervalo de tempo” e “Receber retorno da AGETIC”	Para “Falha ao realizar autenticação”, então “Aguardar intervalo de tempo” e “Realizar autenticação novamente”. Para “Portal indisponível”, então “Acionar AGETIC para verificação”, “Aguardar intervalo de tempo” e “Receber retorno da AGETIC”
Ação	Durante o envio de mensagem do “Solicitação de matrícula (parcialmente automático)” para o “Sistema Acadêmico de Graduação (SISCAD) (automático)”, quando “Realizar autenticação no Portal Graduação” ao ocorrer “Falha ao realizar autenticação”, então “Aguardar intervalo de tempo” e “Realizar autenticação novamente”; ao ocorrer “Portal indisponível”, então “Acionar AGETIC para verificação”, “Aguardar intervalo de tempo” e “Receber retorno da AGETIC”	Durante o envio de mensagem do “Solicitação de matrícula (parcialmente automático)” para o “Sistema Acadêmico de Graduação (SISCAD) (automático)”, quando “Realizar autenticação no Portal Graduação” ao ocorrer “Falha ao realizar autenticação”, então “Aguardar intervalo de tempo” e “Realizar autenticação novamente”; ao ocorrer “Portal indisponível”, então “Acionar AGETIC para verificação”, “Aguardar intervalo de tempo” e “Receber retorno da AGETIC”
Rastreabilidade	Piscina “Solicitação de matrícula (parcialmente automático)” para Piscina “Sistema Acadêmico de Graduação (SISCAD) (automático)” na Tarefa de envio “Realizar autenticação no Portal Graduação”	Participant “Solicitação de matrícula (parcialmente automático)” para Participant “Sistema Acadêmico de Graduação (SISCAD) (automático)” na SendTask “Realizar autenticação no Portal Graduação”

na ferramenta PoP *Modeler* [Mattia et al. 2024]<sup>5</sup>, que é uma ferramenta que apoia a modelagem de PoP em BPMN. O módulo PoP-ARE extrai e especifica automaticamente requisitos de interoperabilidade e de tolerância a falhas durante a interoperabilidade de SoS, no formato CSV, em consonância com a [ISO/IEC 29148 2018] e com rastreabilidade ao nível de negócio. O módulo desenvolvido utiliza as mesmas tecnologias de backend (Laravel Lumen e PostgreSQL) e frontend (React e Material IO) da PoP *Modeler*.

Atualmente, as funcionalidades disponíveis na PoP-ARE são: **(i)** extrair os requisitos de interoperabilidade (foco do trabalho de [Costa 2024]) e de tolerância a falhas durante a interoperabilidade de SoS a partir de cada *Modelo Detalhado de Missão de um PoP* em BPMN, elaborado na ferramenta PoP *Modeler* ou por meio de outra ferramenta que gere arquivo com a extensão “.bpmn”; **(ii)** gerar a especificação textual dos requisitos de tolerância a falhas durante a interoperabilidade (ou seja, no envio e no recebimento de retorno de mensagem) no **formato detalhado** (contendo *id*, *classe*, *sujeito*, *constituinte de origem*, *constituinte de destino*, *momento para ocorrência da falha*, *quais falhas ocorram, como contornas tais falhas, ação e rastreabilidade*) e no **formato com-**

<sup>5</sup>Disponível em: <https://popmodeler.ledes.net>

**pacto** (considerando somente os campos *id*, *classe*, *sujeito*, *ação e rastreabilidade*); **(iii)** permitir a seleção de idiomas português ou inglês para o formato da especificação dos requisitos extraídos; e **(iv)** permitir o download das especificações dos requisitos em arquivo no formato CSV.

Para observar a facilidade de uso e a utilidade do PoP-ARE, foi conduzida uma avaliação<sup>6</sup> com quatro engenheiros de SoS de dois SoS distintos, com base no Modelo de Aceitação de Tecnologia (TAM) [Davis 1989]. Os resultados indicaram que todos os participantes concordaram com a facilidade de uso da ferramenta e 75% declararam sua utilidade. Além disso, eles pretendem usar o PoP-ARE na evolução do SoS, uma vez que, por meio do *Modelo Detalhado de Missão* utilizado, é possível observar a ausência de requisitos que não foram especificados anteriormente [Costa et al. 2024b].

Devido ao espaço limitado, na Tabela 4 (colunas 1 e 3) é apresentado um dos requisitos extraídos e especificados de maneira automática pelo módulo PoP-ARE, tendo como fonte de informação a mesma tarefa de envio do estudo de caso (Seção 7.1). O campo “ID” foi preenchido automaticamente com o próprio ID da tarefa que apresentou a falha, o campo “Classe” está relacionado ao tipo do requisito (ou seja, tolerância a falhas) e o campo “Sujeito” foi preenchido automaticamente com o nome do PoP, que é correspondente ao nome do SoS que o automatiza. Logo, é possível observar que a ferramenta extrai e especifica os requisitos conforme definido pelas diretrizes de extração (Seção 7). Mais informações sobre como realizar a extração automática dos requisitos de tolerância a falhas de SoS são obtidas em [Pereira 2024]<sup>7</sup>.

## 8. Conclusão

Este trabalho colaborou para o avanço da Engenharia de Requisitos de SoS e, consequentemente de SoIS, alinhada ao negócio, especialmente nas etapas de extração e especificação de requisitos com enfoque em um tipo de requisito de confiabilidade, ou seja, tolerância a falhas durante a interoperabilidade. Para viabilizar o alinhamento dos requisitos de tolerância a falhas de SoS aos processos de negócio de alianças de organizações (ou seja, PoP), os requisitos são extraídos de maneira sistemática e automática a partir de informação valiosa sobre o tratamento de exceções durante a interoperabilidade de PoP representada em modelos de processos de negócio. Com isso, o trabalho contribuiu para o estado da arte e respondeu à questão de pesquisa (Seção 3), uma vez que possibilita que esses requisitos não funcionais sejam extraídos em fases iniciais do ciclo de vida do SoS e estejam alinhados às falhas que podem ocorrer durante a interoperabilidade em nível de negócio, colaborando para manter a estabilidade do SoS que automatiza os processos de negócio das alianças de organizações e para alcançar os seus objetivos estratégicos de negócio. Como publicações decorrentes do trabalho, foram realizadas uma publicação técnica<sup>8</sup> e seis publicações científicas, sendo três delas como primeiro autor [Molina et al. 2022, Molina et al. 2023, Molina and Andrade 2024] e as outras três como coautor [Mattia et al. 2024, Costa et al. 2024a, Costa et al. 2024b].

<sup>6</sup>Disponível em: <https://link.ufms.br/wEM5X>

<sup>7</sup>Video de demonstração da PoP-ARE: <https://doi.org/10.6084/m9.figshare.25913857.v1>

<sup>8</sup>Registro de software BR512024000971-0

## Agradecimentos

Este trabalho contou com o apoio da Universidade Federal de Mato Grosso do Sul (UFMS) e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por meio do Código de Financiamento 001.

## Referências

- Andrews, Z., Fitzgerald, J., Payne, R., and Romanovsky, A. (2013). Fault modelling for Systems of Systems. In *11th International Symposium on Autonomous Decentralized Systems (ISADS)*.
- Batista, T. (2013). Challenges for SoS architecture description. In *1st International Workshop on Software Engineering for Systems-of-Systems (SESoS)*.
- Cagnin, M. I. and Nakagawa, E. Y. (2021). Towards dynamic Processes-of-Business Processes: a new understanding. *Business Process Management Journal*.
- Cagnin, M. I. and Nakagawa, E. Y. (2022). M-PoP: Leveraging the Systematic Modeling of Processes-of-Business Processes. *Business Process Management Journal*.
- Cagnin, M. I. and Nakagawa, E. Y. (2024). Processes-of-Business Processes: a novel information source of Systems-of-Systems requirements. *Requirements Engineering. (Under review)*.
- Cook, J. L. (2008). Multi-state reliability requirements for complex systems. In *2008 Annual Reliability and Maintainability Symposium*.
- Costa, M., Molina, S., Paiva, D., and Cagnin, M. (2024a). Cenários abstratos de interoperabilidade para modelagem de Processos-de-Processos de Negócio. In *VI Workshop de Modelagem e Simulação de Sistemas intensivos em Software*.
- Costa, M., Molina, S., Paiva, D., and Cagnin, M. (2024b). PoP-ARE: A tool for extracting Systems-of-Systems non-functional requirements from Processes-of-Business Processes. In *XXXVIII Simpósio Brasileiro de Engenharia de Software (SBES)*.
- Costa, M. G. N. d. (2024). Abordagem baseada em cenários para extrair requisitos de interoperabilidade de Sistemas-de-Sistemas a partir de Processos-de-Processos de Negócio. Master's thesis, Facom, UFMS. <https://repositorio.ufms.br/handle/123456789/8783>.
- Damm, W. and Vincentelli, A. S. (2015). A conceptual model of system of systems. In *2nd International Workshop on the Swarm at the Edge of the Cloud*.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340.
- Dersin, P. (2014). Systems of Systems. IEEE-Reliability Society. Technical Committee on “Systems of Systems”, <https://rs.ieee.org/technical-activities/technical-committees/systems-of-systems.html> (Access in 28/02/2024).
- Despotou, G. and Kelly, T. (2009). A deviation based systems of systems safety view for modelling architectural frameworks. *IET Journals*.

- El Hachem, J., Pang, Z. Y., Chiprianov, V., Babar, A., and Aniorte, P. (2016). Model driven software security architecture of Systems-of-Systems. In *23rd Asia-Pacific Software Engineering Conference (APSEC)*.
- Ferreira, F. H., Nakagawa, E. Y., and dos Santos, R. P. (2021). Reliability in software-intensive systems: Challenges, solutions, and future perspectives. In *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*.
- Garro, A. and Tundis, A. (2014). On the reliability analysis of systems and SoS: The RAMSAS method and related extensions. *IEEE Systems Journal*.
- Hyun, S., Song, J., Shin, S., Baek, Y.-M., and Bae, D.-H. (2020). Pattern-based analysis of interaction failures in Systems-of-Systems: a case study on platooning. In *27th Asia-Pacific Software Engineering Conference (APSEC)*.
- Imamura, M., Ferreira, F. H., Fernandes, J. C., and Santos, R. (2021). System-of-systems reliability: An exploratory study in a brazilian public organization. In *XVII Brazilian Symposium on Information Systems*, pages 1–8.
- Ingram, C., Andrews, Z., Payne, R., and Plat, N. (2014). SysML fault modelling in a traffic management System of Systems. In *9th International Conference on System of Systems Engineering (SOSE)*.
- ISO/IEC 29148 (2018). ISO/IEC/IEEE international standard - systems and software engineering – life cycle processes – requirements engineering.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report Technical Report EBSE 2007-001, Keele University and Durham University.
- Laliberte, C. D., Giachetti, R. E., and Kolsch, M. (2022). Evaluation of natural language processing for requirements traceability. In *17th Annual System of Systems Engineering Conference (SOSE)*.
- Lewis, G. A., Morris, E., Place, P., Simanta, S., and Smith, D. B. (2009). Requirements engineering for Systems of Systems. In *3rd Annual IEEE Systems Conference*.
- Maier, M. (1998). Architecting Principles for Systems-of-Systems. *Systems Engineering*.
- Mattia, W., Franco, I., Costa, M., Molina, S., Paiva, D., and Cagnin, M. (2024). PoP Modeler: Uma ferramenta para facilitar a modelagem e compreensão de Processos-de-Processos de Negócio. In *VI Workshop de Modelagem e Simulação de Sistemas intensivos em Software*.
- Mohammadi, N. G. and Heisel, M. (2016). Patterns for identification of trust concerns and specification of trustworthiness requirements. In *21st European Conference on Pattern Languages of Programs*.
- Molina, S. and Andrade, A. (2024). Modelagem de falhas e exceções em processos de negócios em BPMN: Um mapeamento sistemático da literatura. In *VI Workshop de Modelagem e Simulação de Sistemas intensivos em Software*.
- Molina, S., Paiva, D., and Cagnin, M. I. (2022). Tratamento de requisitos de confiabilidade de Sistemas-de-Sistemas: Um mapeamento sistemático da literatura. In *XXV Congresso Ibero-Americano em Engenharia de Software*.

- Molina, S. d. A., Costa, M. G. N., Nazário, A. G., Paiva, D. M. B., and Cagnin, M. I. (2023). Cenários abstratos de tratamento de exceções na interoperabilidade de Processos-de-Processos de Negócios. In *V Workshop de Modelagem e Simulação de Sistemas intensivos em Software*.
- Ncube, C. and Lim, S. L. (2018). On systems of systems engineering: A requirements engineering perspective and research agenda. In *26th International Requirements Engineering Conference (RE)*.
- Neto, V. G., Oquendo, F., and Nakagawa, E. Y. (2016). Systems-of-Systems: challenges for information systems research in the next 10 years. In *I Grand Research Challenges in Information Systems in Brazil, Special Committee on Information Systems (CE-SI)*. Brazilian Computer Society, pages 1–3.
- Nogueira, F. A. and De Oliveira, H. C. (2017). Application of heuristics in business process models to support software requirements specification. In *19th International Conference on Enterprise Information Systems*.
- Object Management Group (2013). Business Process Model and Notation (BPMN). <http://www.omg.org/spec/BPMN/2.0.2/>. Access in: 28/02/2024.
- Pereira, S. d. A. M. (2024). Abordagem baseada em cenários para extrair requisitos de tolerância a falhas de Sistemas-de-Sistemas a partir de Processos-de-Processos de Negócio. Master's thesis, Facom, UFMS. <https://repositorio.ufms.br/handle/123456789/8784>.
- Prasad, G. and Milks, W. A. (2008). Autonomous verification architectures for complex systems. In *International Multi-Conference on Engineering and Technological Innovation (IMETI)*.
- Rambikur, A., Giammarco, K., and O'Halloran, B. (2017). Systems architecture in failure analysis (applications of architecture modeling to system failure analysis). In *12th System of Systems Engineering Conference (SoSE)*.
- Sanduka, I. and Obermaisser, R. (2015). Model-based development of systems-of-systems with reliability requirements. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*.
- Svenson, P., Olsson, T., and Axelsson, J. (2022). Constituent systems quality requirements engineering in co-operative Systems of Systems. In *17th Annual System of Systems Engineering Conference (SOSE)*.
- Tianjian, W. and Xin, P. (2014). Research on the task process reliability modeling of SoS. In *26th Chinese Control and Decision Conference (CCDC)*.
- Van Bossuyt, D. L., O'Halloran, B. M., and Arlitt, R. M. (2018). Irrational system behavior in a System of Systems. In *13th Annual Conference on System of Systems Engineering (SoSE)*.
- Wholin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). *Experimentation in Software Engineering: an introduction*. Massachusetts: Kluwer Academic Publishers.
- Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer.