

Tratamento de Requisitos de Confiabilidade de Sistemas-de-Sistemas: Um Mapeamento Sistemático da Literatura

Sidny de Almeida Molina, Débora Maria Barroso Paiva, Maria Istela Cagnin

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
CEP 79070-900 – Campo Grande – MS – Brasil

{sidny.molina,deborapaiva,istelamachado}@ufms.br

Abstract. *Systems-of-Systems (SoS) is a set of software systems from distinct organizations that form alliances of organizations. These systems work together to reach broader business goals, which cannot be achieved by a system alone. Due to SoS characteristics, their requirements change over time, bringing challenges for Requirements Engineering. This paper aims to discover how reliability requirements are addressed during the SoS life cycle. As the main results, there is evidence that this requirement type is being commonly handled at a phase later than the requirement phase, and there is a lack of techniques to address them appropriately.*

Resumo. *Sistemas-de-Sistemas (SoS) é um agrupamento de sistemas de software de diversas organizações que formam alianças de organizações. Esses sistemas trabalham em conjunto para alcançar objetivos de negócio mais complexos, não obtidos por um único sistema de software. Devido às características do SoS, seus requisitos mudam ao longo do tempo, trazendo desafios para a Engenharia de Requisitos. O objetivo deste artigo é descobrir como os requisitos de confiabilidade são tratados durante o ciclo de vida do SoS. Evidenciou-se que esse tipo de requisito está sendo geralmente considerado em fase posterior à fase de requisitos e que há carência de técnicas para tratá-los de maneira apropriada.*

1. Introdução

Sistemas-de-Sistemas (SoS) resultam da necessidade de interoperabilidade entre sistemas de software individuais (constituintes) para oferecerem recursos e funcionalidades mais amplas (ou seja, missões) que não poderiam ser entregues pelos sistemas constituintes individualmente [Dersin 2014]. SoS possuem algumas características peculiares que os diferenciam de outros sistemas complexos, como independência operacional e gerencial dos seus constituintes, comportamento emergente e desenvolvimento evolucionário [Maier 1998]. Os sistemas constituintes de um SoS podem pertencer a diferentes organizações que têm seus processos de negócios específicos. Tais processos, quando reunidos para alcançar objetivos estratégicos de alianças de organizações (tais como parcerias, fusões ou aquisições de empresas), fazem surgir processos de negócios grandes, complexos e dinâmicos que são automatizados por SoS e são denominados como Processos-de-Processos de Negócio (PoP - *Processes-of-Business Processes*) [Cagnin and Nakagawa 2021].

O SoS evolui continuamente para responder as mudanças das missões e, assim, obter comportamentos emergentes resultantes da comunicação entre os constituintes para alcançar os objetivos estratégicos de negócio das alianças de organizações. Por consequência, podem haver falhas nos próprios sistemas constituintes, na comunicação entre eles [Ferreira et al. 2021] (por exemplo, atrasos de comunicação, indisponibilidade do serviço e problemas de perda de pacotes de dados [Alexander et al. 2004]) ou durante a reorganização da arquitetura dinâmica do SoS quando os constituintes são adicionados, alterados ou removidos durante a sua execução [Batista 2013], afetando o funcionamento e a confiabilidade do SoS como um todo [Ferreira et al. 2021]. Diante desse cenário, é primordial manter a estabilidade do SoS mesmo quando ocorre algum tipo de falha ou comportamento inesperado. Apesar da importância dos requisitos de confiabilidade para a estabilidade do SoS, eles ainda não são muito explorados na literatura [Bianchi et al. 2015, Cadavid et al. 2020, Ferreira et al. 2021].

Em paralelo, devido às peculiaridades do SoS, diversos aspectos podem dificultar a Engenharia de Requisitos (que envolve a extração, análise, especificação, validação e gerenciamento das mudanças de requisitos [Dick et al. 2017]), por exemplo, múltiplas interações entre diversos sistemas constituintes que têm necessidades de uso diferentes com as suas particularidades; constituintes podem pertencer a diferentes domínios e conter conceitos homônimos com significados divergentes entre os domínios; variação da capacidade do SoS de acordo com situações operacionais específicas do negócio, mesmo dentro de um único domínio de negócio; fatores internos ou externos ao SoS que podem levar a demandas de mudanças rápidas nos constituintes e no próprio SoS [Lewis et al. 2009]. A partir desses desafios, é crucial utilizar técnicas e mecanismos apropriados para realizar o gerenciamento de requisitos durante todo o ciclo de vida do SoS [Ncube and Lim 2018].

Diante da importância dos requisitos de confiabilidade e dos desafios inerentes à Engenharia de Requisitos no contexto de SoS, o principal objetivo deste artigo é descobrir como esse tipo de requisito não funcional têm sido extraído, especificado e validado na literatura. Para isso, foi conduzido um mapeamento sistemático da literatura [Kitchenham and Charters 2007], em que foram selecionados 12 estudos primários. Os principais resultados obtidos indicam que os estudos dão maior enfoque na representação de requisitos de confiabilidade, principalmente de tolerância a falhas e de recuperabilidade. Esses requisitos são extraídos principalmente a partir de falhas que podem ocorrer durante a comunicação entre os sistemas constituintes e, em geral, são especificados por meio de modelos e diagramas em nível de projeto arquitetural.

A escrita deste artigo está organizada em mais quatro seções. A Seção 2 aborda sobre requisitos de confiabilidade no contexto de SoS. A Seção 3 detalha o processo do mapeamento sistemático conduzido no âmbito deste trabalho. A Seção 4 discute os resultados e, por último, a Seção 5 apresenta a conclusão.

2. Requisitos de Confiabilidade de SoS

De acordo com a norma ISO/IEC 25010 [2011], *confiabilidade* é uma característica de qualidade de software e se refere ao grau em que um sistema executa suas funções sob condições específicas por um determinado período de tempo. *Confiabilidade* é composta pelas seguintes subcaracterísticas [ISO/IEC 25010 2011]: (i) *tolerância a falhas*: grau

em que um sistema opera conforme planejado, apesar da presença de falhas de hardware ou software. Devido aos comportamentos emergentes de um SoS, resultantes das interações intrínsecas entre seus constituintes, existe a possibilidade de ocorrerem falhas ou erros [Hyun et al. 2020] que são considerados excepcionais em sistemas monolíticos tradicionais [Andrews et al. 2013]. De maneira complementar, dependendo de como os elementos do SoS colaboram, falhas podem ser propagadas e reveladas como outros tipos de falhas em diferentes elementos do sistema [Despotou and Kelly 2009]; (ii) *recuperabilidade*: grau em que, em caso de interrupção ou falha, um sistema pode se recuperar e retornar a um estado pretendido. No contexto de SoS, é a capacidade de recuperar dados do SoS diretamente afetados após uma falha (por exemplo, durante a interoperabilidade entre os constituintes) e restabelecer o SoS a um estado desejado. Porém, nem sempre é claro entender o processo da recuperação dos sistemas constituintes quando um SoS está sendo recuperado como um todo [Andrews et al. 2013]; (iii) *disponibilidade*: grau em que um sistema está operacional e acessível quando for utilizado. No contexto de SoS, como os constituintes podem ser inseridos, alterados ou removidos em tempo de execução, é primordial garantir que os constituintes essenciais para o funcionamento do SoS estejam disponíveis no momento adequado para o cumprimento de uma determinada missão; e (iv) *maturidade*: grau em que um sistema atende às necessidades de confiabilidade sob operação normal. Sob a perspectiva de SoS, é necessário garantir a sua estabilidade durante o seu uso pelos diferentes *stakeholders*. Para isso, é primordial assegurar a reconfiguração do SoS, devido à sua natureza dinâmica [Garro and Tundis 2014], em que os constituintes podem ser atualizados, fazer parte ou deixar de fazer parte de um SoS.

3. Mapeamento Sistemático

Para a execução do Mapeamento Sistemático da Literatura (MSL) foi adotado o processo proposto por Kitchenham and Charters [2007]. As etapas planejamento, condução e apresentação dos resultados estão descritas, respectivamente, nas Seções 3.1, 3.2 e 3.3. A ferramenta Parsifal¹ forneceu apoio para registrar principalmente as duas primeiras etapas. A ferramenta MS Excel também foi utilizada para tabular os dados dos estudos primários e confeccionar os gráficos.

3.1. Planejamento

O principal objetivo do MSL apresentado neste artigo é descobrir como os requisitos de confiabilidade de SoS têm sido extraídos, especificados e validados na literatura. Para isso, foi elaborada a seguinte questão de pesquisa principal (**QP**): *Como os requisitos de confiabilidade de SoS estão sendo elicitados, especificados e validados?*

Com o intuito de fornecer uma resposta mais detalhada para a QP, foram estabelecidas as seguintes subquestões de pesquisa:

- **Sub-QP1.1:** *Quais tipos de requisitos (subcaracterísticas) de confiabilidade, de acordo com a norma [ISO/IEC 25010 2011], estão sendo tratados pelos estudos primários no contexto de SoS?*
- **Sub-QP1.2:** *Em qual fase do ciclo de vida de SoS os requisitos de confiabilidade estão sendo extraídos e especificados?*

¹<https://parsif.al/>

- **Sub-QP1.3:** *Quais mecanismos (ou seja, método, técnica, abordagem, framework e etc) estão sendo utilizados para extrair requisitos de confiabilidade de SoS e quais fontes de informação estão sendo adotadas para isso?*
- **Sub-QP1.4:** *Quais técnicas estão sendo utilizadas para especificar ou representar os requisitos de confiabilidade de SoS?*
- **Sub-QP1.5:** *Quais critérios são utilizados para validar os requisitos de confiabilidade de SoS?*
- **Sub-QP1.6:** *Como os mecanismos de elicitação e técnicas de especificação de requisitos de confiabilidade têm sido avaliados?*

A estratégia de busca envolveu buscas eletrônicas em três bibliotecas digitais (*Scopus*², *ACM Digital Library*³, *IEEE Xplore*⁴) frequentemente adotadas em mapeamentos e revisões sistemáticas da literatura na área de Engenharia de Software. Como a *Scopus* indexa artigos de diferentes bibliotecas digitais, ela foi utilizada pelos autores para realizar a calibragem da string de busca padrão, tomando como base cinco artigos de controle destacados em negrito na Tabela 3. Esses artigos foram identificados pelos autores a partir de um levantamento realizado antes da execução do MSL.

Para a elaboração da string de busca padrão, foram selecionadas três palavras-chaves relacionadas aos tópicos de interesse do MSL (*Systems-of-Systems*, *Reliability* e *Discover*). Para obter uma string de busca mais abrangente, os autores identificaram sinônimos para cada palavra-chave. Esses sinônimos foram encontrados em artigos publicados em conferências e periódicos da área de pesquisa. Após vários refinamentos durante a calibragem da string de busca preliminar, com o apoio dos artigos de controle, obteve-se a string de busca padrão:

(“Systems of Systems” OR “SoS” OR “System of System” OR “Systems of System” OR “System of Systems” OR “Systems-of-Systems” OR “System-of-System” OR “Systems-of-System” OR “System-of-Systems” OR “Systems of Information Systems” OR “SoIS” OR “Systems of Information System” OR “Systems-of-Information Systems” OR “Systems-of-Information System”) AND (reliability OR fault OR failure OR recoverability) AND (discover OR discovery OR elicitation OR obtaining OR gaining OR acquisition OR capture OR capturing OR extracting OR specification OR representation OR modeling)

Como os motores das bibliotecas digitais selecionadas possuem sintaxe distintas, a string de busca padrão foi adaptada para cada uma delas. Adicionalmente, para apoiar a seleção dos estudos primários foram estabelecidos critérios de inclusão (CI) e de exclusão (CE) exibidos, respectivamente, nas Tabelas 1 e 2. Com base nesses critérios, estudos devem ser incluídos se atendem pelo menos um dos CI e excluídos se atendem ao menos um dos CE.

Com o intuito de facilitar a extração dos dados dos estudos primários para responder as questões de pesquisa, os autores definiram um formulário de extração contendo os seguintes campos: (1) identificador do artigo (id), (2) ano de publicação, (3) tipo de publicação, (4) tipo dos autores (academia, indústria ou ambos), (5) resumo, (6) pontos

²<https://www.scopus.com/>

³<https://www.acm.org/>

⁴<https://ieeexplore.ieee.org/>

Tabela 1. Critérios de inclusão

Código	Descrição
CI1	O estudo propõe uma solução de extração de requisitos de confiabilidade de Sistemas-de-Sistemas
CI2	O estudo avalia o mecanismo de extração de requisitos de confiabilidade de Sistemas-de-Sistemas
CI3	O estudo avalia requisitos de confiabilidade de Sistemas-de-Sistemas
CI4	O estudo considera fatores/elementos que influenciam a confiabilidade em Sistemas-de-Sistemas

Tabela 2. Critérios de exclusão

Código	Descrição
CE1	O estudo não atende a pelo menos um dos critérios de inclusão
CE2	O estudo não está disponível
CE3	O estudo está escrito em uma linguagem diferente do inglês
CE4	O estudo é similar a outro que reporta os mesmos resultados, no qual o mais recente é base para a análise
CE5	O estudo é um documento não avaliado por pares (por exemplo, prefácio, livro, editorial, resumo, pôster, painel, palestra, mesa redonda, oficina ou demonstração)

fortes, (7) pontos fracos, (8) domínio de aplicação, (9) subcaracterísticas de confiabilidade, (10) fase do ciclo de vida do SoS que ocorre a extração, (11) fase do ciclo de vida do SoS que ocorre a especificação, (12) mecanismos utilizados para a extração, (13) fontes de informação utilizadas para apoiar a extração, (14) técnicas utilizadas para a especificação, (15) elemento(s) da técnica utilizado(s), (16) linguagem/notação adotada pela técnica, (17) desafios/problemas de confiabilidade tratados, (18) solução, (19) técnica utilizada para validar os requisitos, (20) critérios utilizados para validar os requisitos e (21) método de avaliação do estudo.

As principais ameaças à validade deste trabalho estão relacionadas a: (i) *possível perda de estudos relevantes*: para contornar essa ameaça, conduzimos as buscas em bibliotecas digitais variadas e efetuamos diversos refinamentos na string de busca utilizando termos encontrados nos artigos de controle. O termo “requisito” não foi incluído na string para não correr o risco de limitar o escopo da busca. (ii) *parcialidade na seleção dos estudos*: para mitigar essa ameaça nós seguimos rigorosamente o processo de MSL utilizado [Kitchenham and Charters 2007] e todos os autores participaram das duas etapas de seleção; todas as divergências foram discutidas e resolvidas por todos os autores. (iii) *falta de padronização na extração dos dados e de confiabilidade dos resultados*: para tratar essa ameaça um formulário de extração foi criado na ferramenta Parsifal e validado por todos autores. Os dados dos estudos primários foram extraídos pelo primeiro autor e validados e refinados pelos demais. Todo resultado obtido foi discutido entre os autores para melhor síntese das respostas das questões de pesquisa.

3.2. Condução

A etapa de condução do MSL foi sistematizada em cinco passos. No **passo 1**, foram realizadas buscas no período de 27/09/2021 a 17/11/2021 em cada biblioteca digital selecionada (IEEE Xplore, ACM e Scopus), considerando a string de busca correspondente. As

buscas retornaram um total de 603 estudos, que foram importados na ferramenta Parsifal. No **passo 2**, 107 estudos duplicados foram identificados e removidos com o apoio da Parsifal, obtendo-se 496 estudos para serem submetidos à primeira seleção (**passo 3**). Nesse passo, a seleção se deu por meio da leitura do título e resumo dos artigos, eliminando um total de 392 estudos de acordo com os critérios de inclusão (Tabela 1) e exclusão (Tabela 2), obtendo-se um total de 104 artigos. No **passo 4**, foi realizada leitura completa dos 104 artigos obtidos no passo anterior, aplicando os critérios mencionados previamente. Nesse penúltimo passo foram selecionados 12 estudos primários (Tabela 3⁵), cujos dados foram extraídos no **passo 5** com o apoio do formulário de extração, definido na etapa de planejamento.

Tabela 3. Estudos primários selecionados

ID	Ano	Tipo de Publ.	Título
E1	2015	Workshop	A Conceptual Model of System of Systems [Damm and Vincentelli 2015]
E2	2018	Conferência	Irrational System Behavior in a System of Systems [Van Bossuyt et al. 2018]
E3	2020	Conferência	Pattern-based Analysis of Interaction Failures in Systems-of-Systems: a Case Study on Platooning [Hyun et al. 2020]
E4	2008	Conferência	Autonomous verification architectures for complex systems [Prasad and Milks 2008]
E5	2014	Conferência	Research on the task process reliability modeling of SoS [Tianjian and Xin 2014]
E6	2017	Conferência	Systems architecture in failure analysis (Applications of architecture modeling to system failure analysis) [Rambikur et al. 2017]
E7	2016	Conferência	Model Driven Software Security Architecture of Systems-of-Systems [El Hachem et al. 2016]
E8	2014	Periódico	On the Reliability Analysis of Systems and SoS: The RAMSAS Method and Related Extensions [Garro and Tundis 2014]
E9	2009	Conferência	A deviation based systems of systems safety view for modelling architectural frameworks [Despotou and Kelly 2009]
E10	2021	Conferência	System-of-Systems Reliability: An Exploratory Study in a Brazilian Public Organization [Imamura et al. 2021]
E11	2013	Conferência	Fault modelling for systems of systems [Andrews et al. 2013]
E12	2014	Conferência	SysML fault modelling in a traffic management system of systems [Ingram et al. 2014]

Os resultados detalhados da primeira (passo 3) e da segunda seleção (passo 4) são apresentados, respectivamente, nas Tabelas 4 e 5.

3.3. Resultados

Os dados dos estudos primários, extraídos na etapa anterior (Seção 3.2), foram tabulados e analisados para obter uma visão geral dos estudos (Seção 3.3.1) e responder as questões de pesquisa deste MSL (Seções 3.3.2 a 3.3.6).

⁵Os estudos destacados em negrito se referem aos artigos de controle.

Tabela 4. Resultado da primeira seleção dos estudos

Bases	Total	Dupl.	CE1	CE2	CE3	CE4	CE5	CI1	CI2	CI3	CI4	1ª Sel.
Scopus	398	60	-278	0	-2	0	-4	+1	+11	+14	+28	54
IEEE	185	47	-96	0	0	0	-1	+4	+7	+10	+20	41
ACM	20	0	-10	0	0	0	-1	0	0	+2	+7	9
Total	603	107	-384	0	-2	0	-6	+5	+18	+26	+55	104

Tabela 5. Resultado da segunda seleção dos estudos

Bases	CE1	CE2	CE3	CE4	CE5	CI1	CI2	CI3	CI4	2ª Sel.
Scopus	-37	-9	-2	0	-1	+1	+1	+1	+2	5
IEEE	-33	-2	0	0	0	0	+1	+4	+1	6
ACM	-8	0	0	0	0	0	0	0	+1	1
Total	-78	-11	-2	0	-1	+1	+2	+5	+4	12

3.3.1. Visão geral dos estudos primários

A Figura 1 apresenta um panorama geral dos estudos primários. Observa-se que a publicação dos estudos está distribuída, em geral, de maneira equilibrada ao longo dos anos (exceto no período 2010 a 2012) e que a maioria dos autores é da academia. Apenas um estudo foi publicado em periódico da área (*IEEE Systems Journal*) e a maioria deles foi publicado em conferências relevantes, como *System of Systems Engineering (SoSE)*, *Asia-Pacific Software Engineering Conference (APSEC)*, *International Symposium on Autonomous Decentralized Systems (ISADS)* e *Brazilian Symposium on Information Systems (SBSI)*.

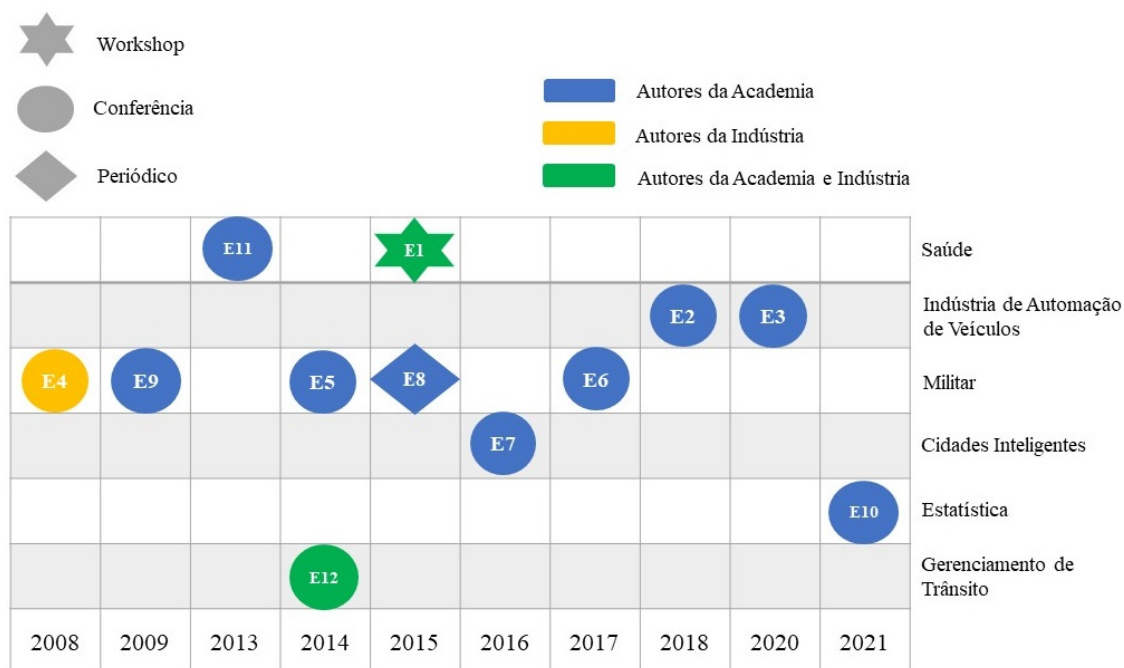


Figura 1. Visão geral dos estudos primários

Considerando o domínio em que os estudos foram aplicados, destaca-se o domínio

militar (5 de 12 estudos). Além desses, os domínios Saúde, Indústria de Automação de Veículos, Cidades Inteligentes, Estatística e Gerenciamento de Trânsito também estão presentes porém com menos recorrência.

Em relação ao país de origem dos autores dos estudos primários, se destacam os países da Europa e da América (em particular do Norte e do Sul), que totalizam 7 e 5 autores, respectivamente. O continente asiático possui 2 autores e a Oceania possui um. Assim, observa-se que o tópico de pesquisa deste MSL está sendo investigado em diversas partes do globo terrestre. Além disso, há colaboração de pesquisa entre autores de alguns países, como é o caso de autores da Alemanha e Estados Unidos (E1), França e Austrália (E7), Reino Unido e Holanda (E12), contribuindo para o fortalecimento dos grupos de pesquisa e para o amadurecimento do tópico sobre confiabilidade de SoS.

3.3.2. Requisitos de Confiabilidade Tratados (Sub-QP1.1)

Os estudos se preocupam mais especificamente com requisitos de tolerância a falhas (12 de 12 estudos) e de recuperabilidade (5 de 12), conforme ilustrado na Tabela 6. O primeiro tipo de requisito é bastante relevante no contexto de SoS pois podem ocorrer diversas falhas principalmente devido à sua arquitetura dinâmica, já que seus constituintes podem ser alterados, removidos ou inseridos, a qualquer momento, em tempo de execução. Além disso, podem ocorrer atrasos de comunicação (por exemplo, devido a dados que dependem de uma coleta manual e são enviados em lote para algum sistema, podendo levar algum tempo para serem incorporados ao banco de dados), indisponibilidade do serviço (por exemplo, em razão do excesso de chamadas de serviços) e também alteração em interfaces dos serviços sem notificar previamente os envolvidos [Imamura et al. 2021]. Assim, justifica-se o segundo tipo de requisito mais tratado pelos estudos pois, uma vez que as possíveis falhas são identificadas, é necessário definir procedimentos adequados para contornar os problemas levantados a fim de manter a estabilidade do SoS para cumprir as suas missões e, conseqüentemente, para atender os objetivos estratégicos de negócio das alianças de organizações. Devido à relação intrínseca entre requisitos de tolerância a falhas e de recuperabilidade, 5 de 12 estudos tratam ambos requisitos.

Tabela 6. Tipos de requisitos de confiabilidade cobertos pelos estudos primários

Tipo	ID
Tolerância a falha	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11 e E12
Recuperabilidade	E7, E8, E10, E11 e E12

3.3.3. Elicitação e especificação de requisitos de confiabilidade durante o ciclo de vida do SoS (Sub-QP1.2)

Poucos estudos tratam da descoberta de requisitos de confiabilidade (5 de 12) e em uma fase posterior à fase de requisitos, ou seja, na fase de projeto arquitetural (4 de 12), conforme indicado na Tabela 7. Apenas um estudo (E10) se preocupa em levantar e especificar esse tipo de requisito na fase de requisitos, porém não discute como isso poderia ser feito durante o desenvolvimento evolucionário do SoS. Esse resultado

evidencia que a área de Engenharia de Requisitos de SoS precisa ser mais explorada [Ncube and Lim 2018], visto que os esforços ainda estão bastante concentrados em nível arquitetural [Cadavid et al. 2020].

Tabela 7. Extração e especificação de requisitos de confiabilidade no ciclo de vida do SoS

Fase	ID (Extração)	ID (Especificação)
Requisito	E10	E10
Projeto Arquitetural	E2, E3, E5 e E9	E1, E2, E3, E4, E5, E7, E8, E11 e E12

Na fase de projeto arquitetural, as capacidades dos constituintes já foram identificadas em fases anteriores do ciclo de vida do SoS, utilizando técnicas apropriadas, como mKAOS [Silva et al. 2015]. Diante disso, o interesse maior dos estudos está em definir *como* as potenciais falhas, que podem surgir durante a interoperabilidade entre os sistemas constituintes ou em algum outro momento da execução do SoS, devem ser tratadas em nível arquitetural para garantir a estabilidade do SoS.

3.3.4. Mecanismos de extração de requisitos de confiabilidade de SoS (Sub-QP1.3)

Conforme já mencionado na Seção 3.3.3, os estudos E2, E3, E5, E9 e E10 tratam da extração de requisitos de confiabilidade de SoS voltados a tolerância a falhas. Para isso, esses estudos utilizam mecanismos distintos (Tabela 8).

Tabela 8. Mecanismos de extração de requisitos de confiabilidade de SoS

Mecanismo	Fonte de informação	ID
Abordagem para identificar vulnerabilidades do SoS	Interação entre constituintes representada em modelos funcionais	E2
Técnica para extrair sequências de interação impróprias entre constituintes	Dados de interação entre constituintes obtidos a partir de <i>logs</i> de execução do SoS	E3
Abordagem para identificar causas de falhas	Diagrama causal	E5
Abordagem DDA (<i>Dependability Deviation Analysis</i>)	Elementos do SoS	E9
Grupo focal ⁶	Equipe de engenharia do SoS	E10

Em mais detalhes, o estudo E2 identifica comportamentos impróprios a partir da análise dos fluxos que saem de cada constituinte do SoS e são eventos iniciais inesperados em outro constituinte. A análise é feita em modelos funcionais, que representam a comunicação entre os sistemas constituintes. Após a identificação dos comportamentos impróprios, os autores atribuem uma probabilidade de ocorrência a cada um deles. De acordo com os autores, a análise dos fluxos e a atribuição de probabilidade podem ser apoiados por técnicas existentes de análise de falhas de produtos (como FMEA (*Failure Modes and Effects Analysis*) e PRA (*Probabilistic Risk Assessment*)). No estudo E3, os autores propõem uma técnica de análise baseada em padrões para processar os dados de interação entre constituintes e extrair sequências de interação impróprias que levam a falhas e dificultam o cumprimento das missões do SoS. Os dados de interação, gerados por

simulador, são obtidos a partir de *logs* de execução de cenários do mundo real que falharam. Já o estudo E5 analisa cada possível falha que pode ocorrer durante o processo de tarefa do SoS, visto que esse processo é composto por outros processos e a falha de cada processo provavelmente afetará todo o processo do SoS. A análise é realizada em um diagrama causal⁷, que representa o processo de tarefa do SoS, e é baseada em uma técnica de análise de falhas de produtos (FMEA). No estudo E9, os autores utilizam a abordagem DDA (*Dependability Deviation Analysis*) para identificar as condições de falhas e efeitos atrelados a cada elemento do SoS. Cada elemento do SoS é investigado com o intuito de levantar desvios do comportamento pretendido (por exemplo, omissão de determinado serviço). Se os efeitos do desvio afetarem a operação do SoS, uma condição de falha é identificada. Em seguida, os elementos do SoS, as respectivas condições de falhas identificadas e os efeitos associados são representados em um modelo de falhas (F-Maps), a partir do qual são extraídos os requisitos de tolerância a falhas. Esses requisitos refletem o contexto de como as falhas afetarão toda a operação do SoS, e não o contexto local de uma única condição de falha. Por outro lado, no E10 os autores conduziram um grupo focal com a equipe de engenharia de SoS de uma organização pública na área de estatística (para apoio a tomada de decisões) para levantar situações que poderiam prejudicar a confiabilidade geral do SoS (por exemplo, chamadas excessivas para serviços de dados, atrasos de comunicação entre os sistemas constituintes e alteração nas interfaces dos serviços fornecidos pelos constituintes).

3.3.5. Técnicas para especificação de requisitos de confiabilidade de SoS (Sub-QP1.4)

Conforme apresentado na Tabela 9, os estudos utilizam diferentes técnicas para especificar os requisitos de confiabilidade, sendo que 8 de 12 estudos (E1, E3, E5, E7, E8, E10, E11 e E12) representam esses requisitos por meio de modelos ou diagramas. Desses estudos, quatro deles (E7, E8, E11 e E12) representam os requisitos de confiabilidade em diagramas distintos da notação SysML⁸, que é uma linguagem utilizada para documentar a arquitetura de sistemas complexos, como o SoS. Em particular, o estudo E7 estendeu o diagrama de blocos da SysML para criar dois outros modelos (de segurança e de objetivos) e o estudo E12 criou um perfil para a SysML, utilizando estereótipos específicos para representar mais apropriadamente as falhas de um SoS. Por outro lado, dois estudos (E8 e E11) utilizam na íntegra alguns diagramas (de requisitos, de definição de bloco, de bloco interno, de sequência e de atividades) da SysML para especificar os requisitos de confiabilidade do SoS com foco em tolerância a falhas.

Considerando os outros estudos que representam os requisitos de confiabilidade em diagramas ou modelos, o estudo E1 utiliza mapa conceitual para representar um panorama geral das falhas que podem ocorrer durante a execução de um SoS. Em outra perspectiva, o estudo E10 estendeu o modelo de missões da mKAOS [Silva et al. 2015], que é uma linguagem orientada a missões criada para representar informação associada a missões do SoS. O modelo estendido possui novos elementos inspirados em três elementos (evento *timer*, evento *cancel* e *gateway* exclusivo) da notação BPMN (*Business Process Model and Notation*). Já o estudo E3 utiliza um modelo de interação, com notação

⁷Esse diagrama mostra a hierarquia dos processos do SoS e suas respectivas tarefas [Tianjian and Xin 2014].

⁸<https://sysml.org/>

Tabela 9. Técnicas para especificação de requisitos de confiabilidade de SoS

Técnica	Notação	ID
Mapa conceitual	Mapa Conceitual	E1
Lista de caminhos de falhas	Textual	E2
Modelo de interação	Próprio autor	E3
Ontologia de falhas	Modelo de dados relacional	E4
Planilha	Textual	E5
Diagrama	IDEF3	E5
Modelo de segurança e modelo de objetivos	Extensão do diagrama de blocos da SysML	E7
Diagrama de requisitos	SysML	E8
Modelo de missões com confiabilidade representada	Extensão da mKAOS	E10
Diagramas de definição de bloco, de bloco interno, de sequência e de atividades	SysML	E11
Diagramas de atividades	SysML	E12

própria, para representar interações entre os sistemas constituintes com comportamento inesperado.

Três estudos (E2, E4 e E5) utilizam representação textual para especificar requisitos de confiabilidade de SoS. Em mais detalhes, E2 apresenta um listagem de cada caminho de execução do SoS que leva a falhas, enquanto que E5 utiliza uma planilha para documentar todas as falhas identificadas do SoS. Essa planilha contém, dentre outras informações, cada tarefa do processo com o respectivo modo, efeito e causa da falha, e também como mitigar a falha. A partir dessa planilha, são gerados diagramas na notação IDEF3 (*Integrated DEFINition for Process Description Capture Method*) para representar graficamente as falhas que podem ocorrer durante o fluxo de tarefas do processo. Por outro lado, E4 representa uma ontologia de falhas em um modelo de dados relacional semântico baseado em axiomas, que possibilita gerar ou atualizar um conjunto exaustivo de falhas de SoS.

3.3.6. Validação dos requisitos de confiabilidade do SoS ((Sub-QP1.5) e avaliação dos mecanismos de extração e das técnicas de especificação (Sub-QP1.6)

Ressalta-se que nenhum dos estudos primários, principalmente aqueles que propõem mecanismos para extrair requisitos de confiabilidade de SoS, explicitaram a forma que esses requisitos foram validados. Em relação à avaliação dos mecanismos de extração e das técnicas de especificação de requisitos de confiabilidade, todos os estudos primários correspondentes foram avaliados por meio de estudos de caso em domínios variados conforme mencionado na Seção 3.3.1. A maioria desses estudos de caso abordou desafios principalmente atrelados a tolerância a falhas (E2, E3, E5, E9 e E10) e recuperabilidade (E10).

4. Discussão e Perspectivas de Trabalhos Futuros

Os sistemas constituintes de um SoS comumente são controlados por diferentes organizações ou podem estar geograficamente distribuídos dentro de uma mesma organização [Maier 1998]. Nesse contexto, a Engenharia de Requisitos de SoS deve ser realizada por meio de colaboração de vários *stakeholders* durante o desenvolvimento evolucionário do SoS para que seu ciclo de vida seja perene [Andrews et al. 2013]. Essa forma de desenvolvimento é uma das características inerentes ao SoS em que os sistemas constituintes podem evoluir a qualquer momento, resultando na evolução do próprio SoS; por outro lado, SoS também evolui devido a mudanças em seu ambiente e, por isso, possui arquitetura dinâmica que causa um novo arranjo de seus constituintes [Maier 1998]. Apesar dessa preocupação, os estudos primários deste MSL dão maior enfoque na elicitação e especificação de requisitos de confiabilidade durante a fase de projeto arquitetural e não apresentam métodos e técnicas específicos para tratar esse tipo de requisito durante a fase de requisitos ao longo de todo o ciclo de desenvolvimento e evolução do SoS.

Outra característica peculiar é o dinamismo do SoS, visto que os sistemas constituintes podem, a qualquer momento, ser adicionados, alterados ou removidos durante a sua execução [Maier 1998], tornando vital garantir a confiabilidade do SoS [Batista 2013]. Compreender os comportamentos dos sistemas constituintes e as comunicações entre eles, bem como identificar anomalias e interrupções que impactem as operações durante a execução do SoS é primordial para estabelecer mecanismos que garantam a sua estabilidade [Ncube and Lim 2018]. Nessa direção, quase metade dos estudos primários que se preocupam com extração de requisitos de confiabilidade consideram como fonte de informação a interação entre os sistemas constituintes e, inclusive, sequências de interação que levam a falhas e prejudicam o alcance das missões do SoS. Entretanto, apesar da informação valiosa dos grandes, complexos e dinâmicos processos de negócios das alianças de organizações e dos diversos benefícios em extrair requisitos a partir de modelos de processos de negócio, dentre eles o alinhamento entre os níveis técnico e de negócio [Cruz et al. 2015], visto que confiabilidade deve ser considerada e gerenciada desde o nível de negócio, nenhum estudo primário utiliza esse artefato como fonte de informação. Os mecanismos de extração de requisitos de confiabilidade dos estudos primários também não se preocupam com possíveis falhas que podem ocorrer quando os constituintes evoluem, devido às características inerentes ao SoS (como independência operacional e gerencial, e desenvolvimento evolucionário), e podem deixar de oferecer as capacidades necessárias para o cumprimento das missões do SoS.

Diferentemente do pouco enfoque dado pelos estudos em extração de requisitos de confiabilidade de SoS, a especificação é abordada por quase todos devido à importância de registrar requisitos não funcionais (dentre eles, requisitos de confiabilidade) como parte da documentação do SoS, especialmente na fase de projeto arquitetural. A maioria dos estudos utiliza modelos ou diagramas para documentar os requisitos de confiabilidade, já que esse tipo de artefato é comumente adotado para representar arquitetura de SoS [Guessi et al. 2019]. Porém, nenhum estudo discutiu uma representação ou *template* padrão para especificar de maneira completa e adequada os requisitos de confiabilidade, considerando as particularidades do SoS, destacando-se a independência operacional e gerencial dos sistemas constituintes, bem como o comportamento emergente e o desenvolvimento evolucionário do SoS.

Com base nas respostas das questões de pesquisa (Seção 3.3) e nas discussões supramencionadas, a seguir são sugeridos alguns desdobramentos de pesquisas futuras para a área de Engenharia de Requisitos de Confiabilidade de SoS: (i) definir métodos e técnicas específicos para tratar requisitos de confiabilidade durante todo o desenvolvimento evolucionário do SoS, considerando inclusive outros tipos de requisitos, como maturidade e disponibilidade, e também o gerenciamento adequado das mudanças dos requisitos de confiabilidade dos sistemas constituintes que podem impactar no alcance da estabilidade do SoS como um todo; (ii) definir técnicas para extrair requisitos de confiabilidade a partir de modelos dos grandes, complexos e dinâmicos processos de negócios de alianças de organizações com o intuito de garantir o alinhamento entre os níveis técnico e de negócio a fim de alcançar de maneira mais efetiva as missões do SoS e, conseqüentemente, os objetivos estratégicos de negócio das alianças de organizações; (iii) estabelecer uma representação adequada, por exemplo, no formato de um *template*, para especificar de maneira completa requisitos de confiabilidade levando em consideração as diferentes características particulares do SoS; (iv) estabelecer técnicas apropriadas para validar os requisitos de confiabilidade, envolvendo os diferentes tipos de *stakeholders* do SoS e os diferentes fatores que afetam a falta de confiabilidade de SoS levantados por Ferreira et al. [2021]; e (v) conduzir estudos de casos e estudos empíricos para avaliar quantitativamente e qualitativamente os métodos e técnicas propostos.

5. Conclusão

Com os resultados obtidos no mapeamento sistemático conduzido no âmbito trabalho, espera-se contribuir para um melhor entendimento sobre como os requisitos de confiabilidade de SoS estão sendo tratados na literatura. Foi constatado que os requisitos de confiabilidade de SoS estão concentrados em tolerância a falhas e recuperabilidade e estão sendo identificados e especificados em fase posterior à fase de requisitos. Por essa razão, as especificações são baseadas principalmente em modelos e diagramas, dentre eles destacando-se os da SysML. A descoberta dos requisitos de confiabilidade é realizada principalmente com base em comportamentos inesperados que podem ocorrer durante a comunicação entre os constituintes. Ademais, este trabalho pode despertar o interesse de especialistas e pesquisadores da área sobre a importância de considerar os grandes, complexos e dinâmicos processos de negócios de alianças de organizações como uma fonte de informação valiosa para extrair os requisitos de confiabilidade do SoS, uma vez que é importante considerar desde o nível de processos de negócio as potenciais falhas (por exemplo, comportamentos não previstos ou interrupções inesperadas) que podem dificultar ou afetar a estabilidade e disponibilidade desses processos. Com essa fonte de informação, é possível manter o alinhamento entre os níveis técnico e de negócio, colaborando para que o SoS ajude as alianças a alcançarem seus objetivos estratégicos de negócio, tornando-as mais eficientes, competitivas e lucrativas. Além disso, é fundamental gerenciar os requisitos de confiabilidade de maneira apropriada durante todo o ciclo do SoS principalmente devido a sua característica evolutiva em que tanto os constituintes quanto o próprio SoS podem evoluir a qualquer momento, culminando em uma nova organização arquitetural do SoS.

Agradecimentos

O presente trabalho foi realizado com apoio da UFMS e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Código de financiamento 001.

Referências

- Alexander, R., Hall-May, M., and Kelly, T. (2004). Characterisation of Systems of Systems failures. In *22nd International System Safety Conference*, pages 499–508. IEEE.
- 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)*, pages 1–8. IEEE.
- Batista, T. (2013). Challenges for SoS architecture description. In *1st International Workshop on Software Engineering for Systems-of-Systems*, pages 35–37. ACM.
- Bianchi, T., Santos, D. S., and Felizardo, K. R. (2015). Quality attributes of Systems-of-Systems: A systematic literature review. In *3rd International Workshop on Software Engineering for Systems-of-Systems*, pages 23–30. IEEE.
- Cadavid, H., Andrikopoulos, V., and Avgeriou, P. (2020). Architecting Systems of Systems: A tertiary study. *Information and Software Technology*, 118:106202.
- Cagnin, M. I. and Nakagawa, E. Y. (2021). Towards dynamic Processes-of-Business Processes: A new understanding. *Business Process Management Journal*, 27(5):1545–1568.
- Cruz, E., Machado, R., and Santos, M. (2015). Bridging the gap between a set of interrelated business process models and software models. In *17th International Conference on Enterprise Information Systems (ICEIS)*, pages 338–345.
- 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*, pages 19–27. ACM.
- 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> (Acesso em 28/01/2022).
- Despotou, G. and Kelly, T. (2009). A deviation based systems of systems safety view for modelling architectural frameworks. *IET Journals*, 1:13.
- Dick, J., Hull, E., and Jackson, K. (2017). *Requirements engineering*. Springer, London.
- El Hachem, J., Pang, Z. Y., Chiprianov, V., Babar, A., and Aniorde, P. (2016). Model driven software security architecture of Systems-of-Systems. In *23rd Asia-Pacific Software Engineering Conference (APSEC)*, pages 89–96. IEEE.
- 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)*, pages 54–61.
- Garro, A. and Tundis, A. (2014). On the reliability analysis of systems and SoS: The RAMSAS method and related extensions. *IEEE Systems Journal*, 9(1):232–241.
- Guessi, M., Graciano-Neto, V. V., and Nakagawa, E. Y. (2019). Architectural description of Systems-of-Information Systems. *Minicourse of Brazilian Symposium on Information Systems*, 1:29–52.

- 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)*, pages 326–335. IEEE.
- 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 *17th Brazilian Symposium on Information Systems (SBSI)*, pages 1–8. ACM.
- 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)*, pages 124–129. IEEE.
- ISO/IEC 25010 (2011). ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
- 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, Keele, UK.
- Lewis, G. A., Morris, E., Place, P., Simanta, S., and Smith, D. B. (2009). Requirements engineering for systems of systems. In *3rd Annual IEEE International Systems Conference (SysCon)*, pages 247–252. IEEE.
- Maier, M. (1998). Architecting Principles for Systems-of-Systems. *Systems Engineering*, 1(4):267–284.
- Morgan, D. L. and Krueger, R. A. (1998). *The focus group guidebook*. Sage.
- 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)*, pages 112–123. IEEE.
- Prasad, G. and Milks, W. A. (2008). Autonomous verification architectures for complex systems. In *International Multi-Conference on Engineering and Technological Innovation (IMETI)*, volume 2, pages 159–163.
- 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)*, pages 1–6. IEEE.
- Silva, E., Batista, T., and Oquendo, F. (2015). A mission-oriented approach for designing System-of-Systems. In *10th System of Systems Engineering Conference (SoSE)*, pages 346–351. IEEE.
- Tianjian, W. and Xin, P. (2014). Research on the task process reliability modeling of SoS. In *26th Chinese Control and Decision Conference (CCDC)*, pages 2262–2266. IEEE.
- 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)*, pages 343–349. IEEE.