Submission date: 20/10/2023
Camera ready submission date: 06/02/2024

1st round notification: 08/01/2024
Available online: 21/03/2024
Publication date: 21/03/2024

Section: extended version
Extended versions of papers presented at the XIX Brazilian Symposium on Information Systems (SBSI 2023)

A Catalog of Heuristics to Support the Design of Systems-of-Systems

Marcio Imamura[1,2], Francisco Henrique Ferreira[1,3], Juliana Fernandes[4], Valdemar Vicente Graciano Neto[5], Rodrigo Pereira dos Santos[1]

[1]Federal University of the State of Rio de Janeiro (UNIRIO)
Rio de Janeiro, Rio de Janeiro – Brazil

[2]Brazilian Institute of Geography and Statistics (IBGE)
Rio de Janeiro, Rio de Janeiro – Brazil

[3]Federal University of Juiz de Fora (UFJF)
Juiz de Fora, Minas Gerais – Brazil

[4]Federal Institute of Piauí (IFPI)
Campo Maior, Piauí – Brazil

[5]Federal University of Goiás (UFG)
Goiânia, Goiás – Brazil

marcio.imamura@ibge.gov.br, francisco.henrique@ufjf.br, juliana.fernandes@ifpi.edu.br, valdemarneto@ufg.br, rps@uniriotec.br

Abstract. *As the presence of systems-of-systems (SoS) intensifies in daily life, professionals face challenges concerning the independence of constituent systems. Addressing these challenges in the design phase is imperative to ensure seamless integration and operational efficiency. In this context, we introduce a catalog of design concerns and practices, herein referred as "heuristics", which aims to offer direction for SoS design. To do so, we grounded in a three-step research method. Firstly, we conducted a systematic mapping study (SMS) to identify which heuristics have been applied to SoS design. Next, we discussed the SMS results in a focus group with professionals to analyze and refine the heuristics. Finally, we surveyed experts to evaluate which heuristics are appropriate to SoS design, resulting in a heuristics catalog. This catalog serves as a resource for professionals, allowing them to pinpoint crucial concerns in the design phase. Moreover, it offers a repository of knowledge, ensuring that professionals do not have to "reinvent the wheel" but can leverage tried-and-true practices to enhance their efficacy.*

Keywords. *Systems-of-Systems; SoS; heuristics; design*

1. Introduction

Large software-intensive systems are increasingly present in the modern world and are developed with various technologies such as Internet of Things (IoT), cloud comput-

Cite as: Imamura, M., Ferreira, F. H., Fernandes, J., Neto, V. V. G. & dos Santos, R. P. (2024). A Catalog of Heuristics to Support the Design of Systems-of-Systems (Um Catálogo de Heurísticas para Apoiar o Design de Sistemas-de-Sistemas). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 17(1), 1:1-1:26. DOI: 10.5753/isys.2024.3774

ing, blockchain, machine learning, and big data. This growth is justified, in part, by the need for fast responses and a reduction in delivery times. Recently, the COVID-19 pandemic has dramatically changed our lives and has made many activities technology-dependent. The call for social distance forced us to stay home for long periods and increased the use of solutions, such as webconferencing [Hacker et al. 2020] and e-learning [Dhawan 2020]. This scenario has also leveraged the adoption of emergent software-intensive solutions, such as smart appliances and virtual reality for health [Singh et al. 2020], sports [Westmattelmann et al. 2020], and entertainment[1], just to mention a few. This leveraged the integration of independent systems, forming arrangements that enable collaboration among different entities by increasing information flows among them. These arrangements of independent systems are referred to as Systems-of-Systems (SoS) [Dersin 2014, Maier 1998, Nielsen et al. 2015].

SoS are complex networks of constituent systems, each capable of operating independently, that contribute towards achieving, when interconnected, common goals, surpassing the capabilities of each single constituent system operating in isolation [Maier 1998]. The complexity of SoS arises from the diversity of the constituent systems, their ability to evolve independently and their interactions. Such interactions can lead to emergent behaviors at the SoS level, a phenomenon that cannot be predicted from individual behaviors of constituent systems [Raman and Murugesan 2022].

SoS are present in several domains, including critical ones such as military [Dietterle 2005], healthcare [Wickramasinghe et al. 2007], and Earth observation [Butterfield et al. 2008], to mention a few. Still, while SoS holds significant potential in addressing complex challenges of the contemporary world, the design of these systems also comes with inherent challenges. For example, there are concerns about how to design and maintain SoS in a more secure, reliable, and efficient operation, since the constituent systems are independent [Ferreira et al. 2021]. These include concerns with multiple decision-makers and operating environments, with constituent systems changing their behavior throughout their lifecycle [Oquendo 2016] and potentially affecting an SoS operation. Addressing these concerns at the design phase can mitigate potential impacts. However, professionals often struggle to address what to consider during the SoS design phase [Keating et al. 2003].

In our previous work [Imamura et al. 2023], we presented an initial catalog of heuristics that outline concerns and best practices to help practitioners during SoS design. To do so, we reviewed the literature to identify good practices and recommendations that should be applied to the design of SoS. In addition, we conducted a focus group with professionals to organize and refine the set of heuristics extracted from the literature review. Now, we consolidated this catalog by conducting a survey with experts to evaluate the heuristics produced after the focus group. This resulted in a catalog with fourteen heuristics divided into five categories. The following research question guided the conduction of our work: *"Which concerns should a SoS professional take into account during the design stage of an SoS?"*

[1] https://www.forbes.com/sites/lisakocay/2020/06/30/lost-horizon-virtual-reality-music-festival/?sh=48ba9ebf3900#442599103900

In summary, the main contributions of this study are as follows:

- A SMS regarding heuristics that can be applied in the design phase of SoS;
- A catalog of heuristics to support the design of SoS;
- An evaluation of the catalog through a survey and a focus group with professionals.

The rest of this article is organized as follows: Section 2 presents the background; Section 3 describes our research method; Section 4 presents the results; Section 5 presents a discussion of the heuristics catalog, presents some implications for practitioners and researchers, and reports the threats and limitations; and Section 7 concludes the article.

2. Background and Related Work

This section sets the foundation for understanding the key concepts relevant to our study: SoS and heuristics. Additionally, we discuss related work.

2.1. Systems-of-Systems

SoS has been extensively studied in the last years. This increasing interest has led several researchers to explore the properties of this class of systems. Mark Maier [Maier 1998] firstly described such properties and defined SoS as an assembly of constituent systems that individually can be considered systems and that have two main properties: the operational and managerial independence of the constituent systems. Operational independence means that an SoS is composed of constituent systems that are independent and useful in their context, i.e, they are capable of performing useful operations independently of the SoS. Managerial independence means that constituent systems are independently managed to accomplish specific purposes. Maier also states that SoS could exhibit geographic distribution, evolutionary development, and emergent behaviors. Over the years, several studies proposed other characterizations for SoS. In particular, Boardman and Sauser [Boardman and Sauser 2006] reviewed more than 40 studies and extended Maier's characterization to encompass five unique SoS properties, namely autonomy, belonging, connectivity, diversity, and emergence, as described in Table 1.

Table 1. Characteristics of SoS [Boardman and Sauser 2006].

Characteristic	Description
Autonomy	Each constituent system is free and independent. Autonomy is the capability of a constituent system to pursue its specific purpose.
Belonging	Constituent systems choose to belong to the SoS on a cost-benefit basis; also, to cause greater fulfillment of their own purposes.
Connectivity	Synergism is enabled by the highly dynamic distributed network.
Diversity	Constituent systems are heterogeneous, self-sufficient systems that are open for enhancement by evolution and adaptation.
Emergence	The cumulative actions and interactions among constituents of an SoS give rise to the behaviors that can be attributed to the SoS as a whole.

SoS can be categorized according to the levels of authority and coordination over the constituent systems into four types. The types directed, collaborative, and virtual were initially defined by Maier [Maier 1998], while the type acknowledged was later added by Dahmann *et al.* [Dahmann et al. 2008], as follows: (i) *directed*, which is centrally managed by an authority responsible for driving operations and ensuring the accomplishment of missions [Maier 1998]; (ii) *acknowledged*, which has recognized objectives, an acknowledged manager, and resources at the SoS level. However, the manager has no complete authority over constituent systems [Dahmann et al. 2008]; (iii) *collaborative*, in which there is no central authority and constituent system engineering teams work together more or less voluntarily to fulfill agreed objectives [Maier 1998]; and (iv) *virtual*, in which there is no managerial authority, common purposes, or established objectives, and constituent systems are often unaware of their participation in an SoS. A virtual SoS also relies upon relatively invisible mechanisms to maintain it [Maier 1998]. The level of authority affects the SoS architectures, as it determines how adaptive and cooperative each constituent system will be concerning its requirements, interfaces, data formats, and technologies.

The design of an SoS presumes that the collaboration among independent systems (and resulted emergent behavior) can help to solve a problem that could not be solved by a single system alone [Sage and Cuppan 2001]. However, ensuring SoS quality is particularly challenging since it is necessary to deal with the intrinsic characteristics of this type of system, which are often related to unpredictability and uncertainties. Dealing with these issues can be challenging for SoS engineers. For instance, is there any way to monitor or mitigate undesirable behaviors when a constituent system changes the level of its contribution or leaves the SoS? This situation can continually affect the achievement of SoS missions and it is necessary to create solutions to minimize uncertainties when designing SoS. In this context, heuristics can be valuable to help professionals identify and resolve design issues that may negatively impact the SoS operation.

2.2. Heuristics

A heuristic is a "strategy that ignores part of the information, intending to make decisions quickly, frugally" than more complex strategies [Gigerenzer and Gaissmaier 2011a]. Heuristic processing relies on judgmental rules (e.g., rules of thumb, industry conventions, and best practices). Decisions made with heuristics processing rely on information cues rather than more in-depth analyses. Unlike statistical methods, a heuristic is a strategy that ignores part of the information, with the goal of making decisions more quickly, not relying on optimization to find the best solution, but instead find a "good enough" satisfying answer [Gigerenzer and Gaissmaier 2011b]. For instance, calculating a function with the maximum precision is a form of optimization, while choosing the first option that exceeds an aspiration level is a form of satisfying.

Since the early twentieth century, heuristics have been used in a variety of fields, including psychology, where cognitive heuristics are studied in human decision making under uncertain conditions [Sherman and Corty 1984]. In political science, moral heuristics are observed to play a pervasive role in moral, political, and legal judgments [Sunstein 2005]. In usability, heuristics are used to find problems in the evaluation of user interface design [Nielsen 1994].

In software engineering, heuristics have been used in several contexts, including design patterns, describing group problems, and helping designers find and correct them. Furthermore, heuristics can allow the construction of a 'softer' model to obtain a more holistic and subjective view of quality [Churcher et al. 2007]. The heuristics proposed by Arthur Riel are an example that covers aspects of object-oriented software development, including classes and objects, multiple inheritances, and association relationship [Riel 1996]. The heuristics proposed by Nielsen [Nielsen 1994] facilitate the user interface design process, and those proposed by Riel [Riel 1996] help to avoid dangerous decisions in object-oriented software development, can be used as guidelines to assist software designers without providing perfect solutions. In this paper, we propose to use heuristics as a strategy to support professionals who identify critical issues during the SoS design phase.

2.3. Related Work

There is still a shortage of studies that specifically address SoS design heuristics. However, there are notable contributions that have advanced research in this area. For example, Beale *et al.* [Beale et al. 2022] introduces a set of heuristics for the design and development of complex systems, including SoS, based on a review of the literature and expert experience in the field. These heuristics are organized into five categories: system definition and requirements, system architecture, system development, system operation and maintenance, and system evaluation and improvement. The study concludes that these heuristics can be valuable for systems engineers working on complex systems, providing comprehensive guidance and contributing to success in the development of these systems. The heuristics were further revised in 2023 to address cultural aspects [Beale et al. 2023].

Axelsson [Axelsson 2022] presents a literature review on design patterns for SoS. The author defines a pattern as "*an abstract description of a solution to a common SoS design problem, which can be adapted to specific contexts*". The study presents a consolidated framework to describe individual patterns and suggests the dimensions along which a catalog of patterns can be organized.

Unlike these works, our research was carried out to find in the literature a set of specific heuristics for SoS design and discuss them in a focus group to verify the applicability and usefulness of the found heuristics, aiming to organize them. Furthermore, the organized set of heuristics was refined by conducting a survey with experts to produce the final heuristic catalog. This catalog aims to help practitioners and decision-makers identify issues to be addressed at SoS design time, optimizing problem-solving costs.

3. Research Method

The catalog of heuristics is grounded in the existing literature. We started the process by conducting an SMS to identify heuristics related to SoS design. We define heuristics as any design principles, rules, or patterns documented in the existing literature. Additionally, we explored the applicability of these heuristics to different types of SoS. Subsequently, we organized a focus group session to capture expert opinions and insights on the identified heuristics, allowing us to refine our results. Figure 1 provides an overview of the stages in our research process.

3.1. Systematic Mapping Study

We conducted the SMS with the aim of identifying and understanding the heuristics applicable in SoS design phase. Our investigation involved a detailed analysis of the current

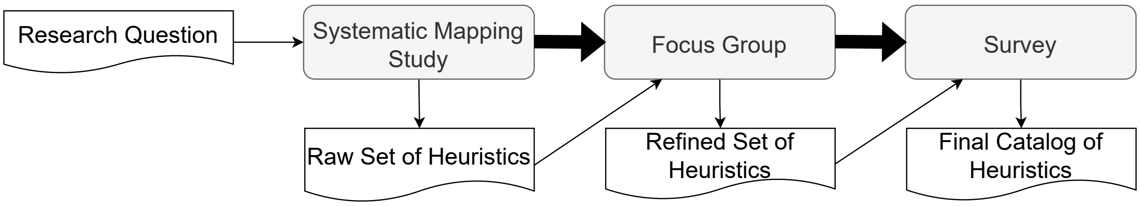

Figure 1. Overview of the research method.

state of the art, looking for patterns, best practices, and recommendations to inform and enhance our understanding of effective SoS design principles. To do so, we followed the guidelines presented by Kitchenham and Charters [Kitchenham et al. 2015] and adopted the process described in Figure 2 as our research method to carry out the SMS. The research question was defined as in the following:

RQ. Which heuristics have been applied to SoS design?

- By answering this research question, we can provide a set of principles, rules, and standards that have been adopted in the design of SoS.

A generic search string was defined from the keywords *"System-of-Systems"*, *"Heuristics"*, *"Design Principle"*, *"Pattern"*, and *"Rule"*. Keywords were connected using the AND logical operator, while variations and synonyms were connected using the OR operator. The terms of the search string were selected with the aim of a broader search, that is, extensive coverage of studies. We tested different configurations of the search string in Scopus[2], which is considered the largest scientific publication database that indexes the most relevant publication venues. After calibration, the final string was:

```
("system-of-systems" OR "SoS" OR "system of systems" OR
"systems of systems" OR "systems-of-systems") AND
("heuristics" OR "heuristic" OR "design principle" OR
"design principles" OR "pattern" OR "patterns" OR "rule" OR
"rules")
```

Besides Scopus, we also carried out searches in ACM Digital Library[3] and IEEExplore[4]. The searches were conducted in May 2020. In the end, 3,765 studies were recovered (1,759 from the ACM Digital Library; 450 from IEEExplore and 1,556 from Scopus). Then we eliminated duplicate studies, resulting in 3,645 studies. Next, we looked at the title, abstract, and keywords and applied the selection criteria described in Table 2 where IC1 is the inclusion criterion and EC1 to EC4 are the exclusion criteria.

After applying the selection criteria, we discarded 3,273 studies, and 372 remained for a detailed analysis. Following, we read the Introduction and Conclusion sections of the remaining 245 studies, and after applying the selection criteria, 14 studies were selected for data extraction. Table 3 lists the selected studies.

[2]http://www.scopus.com
[3]https://dl.acm.com/
[4]https://ieeexplore.ieee.org/

Table 2. Selection criteria.

ID	Criteria
IC1	The study addresses systems-of-systems design principles, patterns, or rules.
EC1	The study is not available for download openly or through institutional access and could not be retrieved from the author.
EC2	The study is not written in English.
EC3	The study is a book, tutorial, editorial, abstract, poster, panel, lecture, round table, workshop, demonstration, or preface.
EC4	The study is not a primary study.

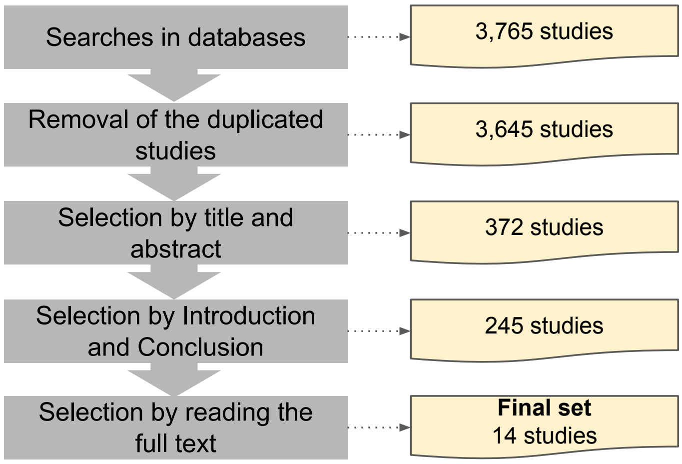

Figure 2. Process of selection of primary studies.

Table 3. Selected studies

ID	Study	Reference
S01	SoS-Centric Middleware Services for Interoperability in Smart Cities Systems, 2016	[Lopes et al. 2016]
S02	Developing systems thinking skills using healthcare as a case study, 2018	[McDermott 2018]
S03	Architecting principles for systems-of-systems, 1998	[Maier 1998]
S04	Challenges for SoS Architecture Description, 2013	[Batista 2013]
S05	Towards a Dynamic Infrastructure for Playing with Systems of Systems, 2014	[Schneider et al. 2014]
S06	When Ecosystems Collide: Making Systems of Systems Work, 2014	[da Silva Amorim et al. 2014]
S07	A cooperative SoS architecting approach based on adaptive multi-agent systems, 2018	[Bouziat et al. 2018]
S08	A generalized options-based approach to mitigate perturbations in a maritime security system-of-systems, 2013	[Ricci et al. 2013]
S09	A Meta-Process to Construct Software Architectures for System of Systems, 2015	[Gonçalves et al. 2015]
S10	Harnessing Emergence: The Control and Design of Emergent Behavior in System of Systems Engineering, 2015	[Mittal and Rainey 2015]
S11	On the Challenges of Self-Adaptation in Systems of Systems, 2013	[Weyns and Andersson 2013]
S12	Randomisation in designing software tests for systems of systems, 2012	[Liang and Rubin 2012]
S13	“Understanding Patterns for System of Systems Integration, 2013	[Kazman et al. 2013]
S14	A Process to Establish, Model and Validate Missions of Systems-of-Systems in Reference Architectures, 2017	[Garcés and Nakagawa 2017]

We adopted the process described in Figure 2 as our research method to carry out the SMS. At the end of the process, 14 studies were selected. Table 3 shows the studies with respective references.

3.2. Focus Group

After extracting the heuristics from the literature, we conducted a focus group with experts. A focus group is a meeting of a group of people to evaluate concepts and/or problems [Caplan 1990]. The focus group’s main objective is to identify participants’ feelings about a particular subject, product, or activity. Its specific objectives vary according to the research approach adopted. This step was necessary since the initial set of heuristics came from different authors. Thus, there are similar, complementary, and divergent views on the heuristics. Hence, additional work should be done to organize the initial set of heuristics. A focus group was selected as the appropriate instrument to capture experts' perceptions and refine the ideas and concepts from the heuristics.

The focus group was mediated by a moderator responsible for guiding discussions, promoting interaction and the engagement of the participants, and enabling the emergence of new ideas about heuristics for SoS design [Zaganelli et al. 2015]. The participants answered the questions and contributed to the discussions that emerged. They also suggested modifications and improvements in the set of heuristics and the organization of the catalog. The focus group meeting was further recorded an transcribed.

We followed the guidelines provided by Beck and Manuel [Beck and Manuel 2008] and Zaganelli *et al.* [Zaganelli et al. 2015]. Figure 3 describes the focus group process. In the planning phase, we invited professionals to participate in the focus group. We selected the professionals according to their familiarity with the SoS theme. Four professionals accepted the invitation. Two participants hold a Master's degree, and 2 were Ph.D. students. After the participants accepted the invitation, we sent material to them providing preliminary knowledge to support the focus group discussions.

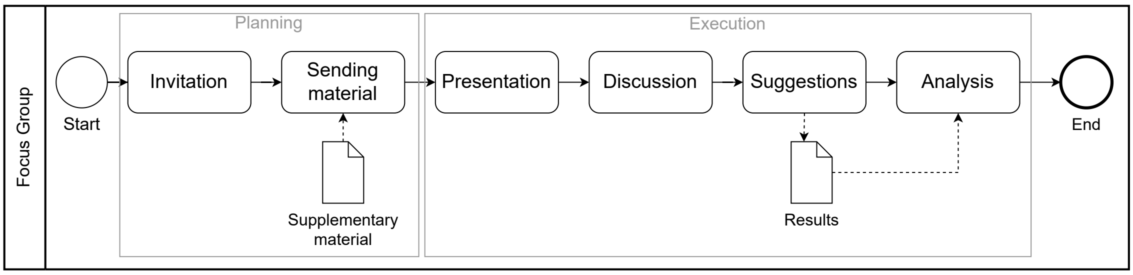

Figure 3. Process of execution of the focus group.

We started the conduction phase with a presentation in which we provided initial considerations and clarifications about our research. We also presented a document containing the initial set of heuristics. The objective of the discussion in the focus group was to clarify the following points:

1. In which situations should a heuristic be applied or not;
2. The appropriate use sequence of the heuristics.

To clarify these points, the mediator encouraged the discussion by asking the following questions:

- **Q1**. Do you understand the goal of the catalog of heuristics?
- **Q2**. Do you think that this catalog is useful for SoS design?
- **Q3**. Are the descriptions of the heuristics appropriate?
- **Q4**. Are there heuristics that can be combined with other heuristics?

Based on theoretical knowledge and practical experiences, participants contributed with suggestions regarding heuristics. Next, we organized the data obtained and conducted an analysis to clarify how the heuristics was perceived by the group [Zaganelli et al. 2015].

The focus group analysis aimed to:

- Verify the validity of the heuristics extracted from the studies selected in the SMS according to the research questions;
- Adjust the description and rationale of each heuristic by standardizing the used terms;
- Combine similar heuristics and exclude others not considered suitable for SoS design, reducing the number of heuristics in final catalog;
- Categorize and group the heuristics, suggesting a logical sequence to be applied in SoS design.

3.3. Survey

After the focus group, a survey was conducted to evaluate the refined set of heuristics. The survey was based on a catalog of heuristics produced conducting an SMS with further refinement in the focus group dynamics described in the previous section. The survey aimed to evaluate the applicability of the heuristics catalog regarding SoS design from the perspective of experts in the SoS context. We invited researchers that formed the scientific committee of SESoS/WDES 2021[5] as SoS experts to evaluate the heuristics catalog for this step. The researchers' emails were collected from personal websites or the websites of the departments of their respective research institutions.

We elaborated a questionnaire describing the objective of the study, how the set of heuristics was produced, and questions to evaluate each heuristic with a statement and example of use. A pilot was conducted with an professional aiming to make adjustments to the questionnaire before sending the invitation to participate in the survey.

The questionnaire was organized into three sections. The first contained the free and clear term to participate in the research, while the second section contained questions regarding the participant's profile. The third section had objective questions for each of the 15 heuristics to be evaluated with the agree, neutral, and disagree options [Chueri 2021] and also an open field so that the participant could provide additional information, such as comments and suggestions. At the end of the questionnaire, an open field was presented for general comments and suggestions. We invited 31 researchers, and 15 responded to our invitation. Out of the total, 14 participants held a PhD, and one had a master's degree. Five participants reported having "very good" knowledge of SoS, while another five reported having "good" knowledge, and the remaining five reported "average" knowledge.

For both the focus group and the survey data, we employed open coding procedures to support the analysis of participants' responses and opinions. This systematic approach allows us to analyze the responses and identify common themes and patterns.

[5]http://sesos-wdes-2021.icmc.usp.br/Committee.php

Open coding offers a structured way to understand the raw data, enabling us to categorize the responses based on their key elements, contributing to a proper data interpretation [Corbin and Strauss 2008].

4. Results

This section presents the results of the SMS, focus group, and survey.

4.1. SMS Results

We conducted our SMS with a specific focus on heuristics applicable to SoS design, as established in our research question. Hence, studies S12, S13, and S14, despite explicitly employing the term 'heuristics,' were excluded due to their lack of relevance to the SoS design phase.

Regarding the heuristics relevant to SoS design, we identified *H.01* from S01, in which the authors propose a middleware service tailored for SoS to streamline their management and execution in smart cities. From S02, we extracted heuristics *H.02* through *H.09*. In this study, the authors adopt a systems thinking approach within a sociotechnical context. The heuristics derived from this study are aimed at ensuring interoperability among constituent systems.

S03 introduces a basic set of architecting principles to assist in SoS design. From this study, we extracted the heuristics *H.10*, *H.11*, *H.12*, and *H.13*. S04 addresses challenges in SoS architecture, considering perspectives at both the SoS and constituent systems levels, and proposes the maintenance of interfaces at both levels. We extracted *H.14* as a heuristic from this study.

From S05, we extracted heuristics *H.15*, *H.16*, *H.17*, and *H.18*. In this study, the authors propose principles for a functional architecture simulator within the context of SoS engineering, aiming to assist in defining the SoS architecture. S06 adopts a sociotechnical ecosystem approach to enhance the sustainability of the SoS composition process. We gathered *H.19*, *H.20*, and *H.21* as heuristics from this study.

The heuristic *H.22* was extracted from S07, in which the authors proposed a new model called SApHESIA (SoS Architecting HEuriStIc based on Agents), focusing on the SoS environment and its dynamics. S08 presents an approach that allows for the identification of measures for mitigating perturbations in SoS. The heuristic *H.23* was extracted from this study.

From S09, we derived an extensive set of heuristics for SoS software architectures. This study presents a meta-process for SoS software architectures (SOAR) designed to aid in constructing SoS software architectures. The extracted heuristics include *H.24* through *H.39*.

Finally, the heuristic *H.40* was extracted from S10, in which the authors discuss the lack of computational and systems engineering approaches to prevent the engineering of emergent behaviors in SoS modeling and simulation. The authors proposed an approach to deal with this issue. Table 4 details the heuristics extracted from our investigation.

Table 4: Raw set of heuristics.

ID	Description	Rationale	Ref.
H.01	SoS coordination must ensure that each constituent systems can exchange and understand data and messages with other constituent systems.	Communication standards need to be preserved by the SoS coordination to ensure that each constituent systems can communicate with the others when necessary.	S1
H.02	The management of SoS must ensure that the policies for accessing and using the capabilities of constituent systems can be shared and understood by all constituent systems.	It is necessary to ensure that each constituent systems can participate in SoS respecting other constituent systems policies. The management and dissemination of these policies are necessary given the characteristics of independence of constituent systems and the SoS dynamic architecture.	S2
H.03	The design process must identify who benefits from SoS.	Every SoS is built and operates to accomplish a set of objectives. It is important to identify those who benefit from the activities and objectives of SoS.	S2
H.04	The design process must identify who pays for the development and operation of SoS.	It is necessary to inform the stakeholders how and by whom SoS construction and operation activities will be funded to deal with these issues correctly.	S2
H.05	The design process must identify who provides the necessary capabilities and resources for SoS operation.	An SoS depends on a synergy between systems that provide the capabilities and resources necessary for its operation.	S2
H.06	The design process must identify which stakeholders are disadvantaged when participating in an SoS.	It is not always possible to guarantee that all SoS participants can only benefit from being part of the system. Therefore, it is necessary to manage the disadvantages of participating in SoS to ensure CS' encouragement and/or replacement.	S2
H.07	The value of a product or service provided by an SoS for each participant must be determined.	The value-centered philosophy of the products or services provided by SoS should be used so that efforts are focused on the benefits brought by the results and not on the results themselves.	S2
H.08	The participants' perception of how an SoS is executed and what the results are must be identified.	The participants' perception of what occurs in an SoS and how it delivers the results is more important than the facts on the operation and the results themselves. Therefore, it is necessary to manage the information flows to understand and appreciate how an SoS is managed.	S2
H.09	Open architectures should be adopted for SoS development and evolution.	An SoS must evolve to adapt to its consumers' needs. Thus, open architectures facilitate the SoS evolution. On the other hand, closed architectures that require the use of proprietary technologies or some licensing can make the SoS evolution difficult or unpractical.	S2
H.10	Stable intermediate forms of architecture must be defined in the SoS evolution process.	For an SoS to dynamically evolve yet maintaining continuous operation, it is necessary to think of intermediate states to enable the feasible transition.	S3
H.11	It must be defined which constituent systems should be helped, which ones can recover on their own and which ones are not worth investing efforts to recover.	Only realistic and possibly practical maintenance strategies should be applied for the maintenance of constituent systems.	S3
H.12	Interfaces between constituent systems must be defined already at design time.	Interfaces between constituent systems are a crucial factor in the operation of an SoS as they enable collaboration.	S3
H.13	Incentive mechanisms for collaboration among constituent systems must be identified in the design phase.	Collaboration among independent systems may have no apparent use for participants.	S3
H.14	The components represented in the model at both SoS-level and CS-level must describe interfaces, properties, and constraints declared unambiguously.	It is required to define interfaces, properties, and constraints for the SoS and constituent systems levels as ways to address how individual properties impact SoS reliability.	S4
H.15	Interoperability must be assured rather than integration.	Unlike integration, interoperability works with loose coupling among systems, favoring the replacement of constituent systems and the evolution of SoS.	S5
H.16	Define which functions should be implemented and which are already available in constituent systems for developing an SoS.	The design process must anticipate whether there are functionalities available in the constituent systems to collaborate or if a functionality needs to be developed.	S5
H.17	Define the data that should be exchanged among constituent systems.	The specification of the data set exchanged among constituent systems must be described in detail to generate the necessary capabilities for an SoS to fulfill its mission.	S5

Continued on next page

Table 4 – *Continued from previous page*

ID	Description	Rationale	Ref.
H.18	It is necessary to justify the reason to constituent systems exchange data.	All constituent systems must be aware of why data is exchanged to ensure value delivery and avoid conflicts.	S5
H.19	Interface patterns resulting from the evolution must be identified.	The evolutionary process can generate the need to adopt new communication patterns or to update existing patterns. It is necessary to maintain the standards as constituent systems evolve.	S6
H.20	Interfaces must be treated in layers or as service buses.	Layered or service bus design allows replacing modules or studying alternatives for required capabilities without affecting SoS operation.	S6
H.21	Validation and verification activities must be applied to all phases of development.	As with traditional systems, rather than testing validation and verification at the end of the lifecycle, each development phase should be followed immediately with these activities that provide feedback so errors can be fixed as soon as possible.	S6
H.22	Criticality must be considered for SoS evolution.	Criticality takes into account the dynamics among the constituent systems and the environment. Each constituent system must autonomously choose which interaction produces less criticality.	S7
H.23	Perturbations must be identified to assess architectural alternatives.	It is necessary to understand which alternatives produce the least disturbance during the SoS operation. Alternatives must be evaluated with metrics that make it possible to evaluate the solution.	S8
H.24	The self-regulatory capabilities of each constituent system must be identified.	Each constituent system has the autonomy to regulate itself at a certain level. This autonomy must be represented in the SoS design.	S9
H.25	SoS architecture must allow feedback.	It is necessary to monitor SoS operation to detect problems during its operation. Monitoring can reveal the behavior of constituent systems and the dynamism of SoS.	S9
H.26	The integration of self-managed systems must be consistent with the processes and individual interests of constituent systems.	How SoS can provide feedback to those responsible for its operation must be foreseen from the design.	S9
H.27	Support for connectivity among geographically dispersed constituent systems must be provided.	It is necessary to ensure that autonomously managed systems consistently contribute to the processes and interests of other systems in an SoS.	S9
H.28	Support for connectivity among heterogeneous environments must be provided.	How connectivity among geographically distributed systems occurs must be identified both for the means of communication and the protocols used.	S9
H.29	Emergent behaviors must be represented in capability composition schemes.	It is desired that the emergent behaviors are known regarding the capabilities required by constituent systems and their relationships.	S9
H.30	Individual capabilities must be validated.	Each constituent system provides a capability to SoS that must be checked to see if it complies with SoS expectations.	S9
H.31	The design must anticipate desired emergent behaviors.	Emergent behaviors required to fulfill missions must be explicitly identified, allocated to the respective constituent system responsible for them, and evaluated frequently.	S9
H.32	Emergent behaviors must be assigned to constituent systems capabilities.	Emerging behaviors required to fulfill SoS missions must be explicitly identified, assigned to the respective constituent system responsible for them, and evaluated frequently.	S9
H.33	SoS capabilities must be constantly analyzed and evaluated.	The capabilities needed to fulfill the SoS missions should be constantly analyzed and monitored by those responsible.	S9
H.34	An incremental development and deployment strategy for the SoS should be adopted.	The design process must embrace steps that allow the incremental development of an SoS, promoting and facilitating the dynamism of its architecture.	S9
H.35	SoS functions must be revised as the system operates.	SoS dynamics and constituent systems independence can lead to problems in fulfilling the SoS mission over time.	S9
H.36	Design decisions must be explicit in the SoS architecture.	During the design, decisions are made to build the SoS architecture. It is necessary to document these decisions.	S9

Continued on next page

Table 4 – *Continued from previous page*

ID	Description	Rationale	Ref.
H.37	Design decisions must be developed and continually refined.	The independence of constituent systems in an SoS can lead to changes in its architecture. New architecture decisions must be re-evaluated and recorded for better SoS maintenance.	S9
H.38	SoS scenarios must be developed and continually refined.	Scenarios in which an SoS runs are important to verify its effectiveness in the context in which it operates.	S9
H.39	A strategy for dynamic integration must be provided.	Constituent systems may leave or join an SoS at any time. It is necessary to adopt mechanisms to minimize the impact of acquiring or losing capabilities.	S9
H.40	Weak emergent behaviors must be identified at design time.	Weak emergent behaviors are those that can be identified at design time. They are not accidental or produce any change in the behavior of constituent systems. A "strong" behavior cannot be predicted at the design phase. It can usually be observed during simulation or in operation.	S10

4.2. Focus Group Results

After reading and clarifying the raw set of heuristics, the focus group participants argued that they clearly understood the purpose of the heuristics catalog (Q1) and were unanimous in stating that it is valuable for SoS design (Q2).

Regarding Q3, the participants argued that improving the descriptions and explanations of the heuristics was needed. This would make it easier for the professional to understand and adopt the heuristics. In addition, they argued that it is essential to maintain uniformity to the terms treated in the heuristics catalog. Although the heuristics come from several studies, it is necessary to adopt terms that avoid confusing readers and use a standardized vocabulary.

Table 5. Heuristics categories.

Category	Description
Initiation (IN)	Concern activities that are applicable right at the beginning of the SoS design.
Constituent systems (CS)	Concern activities of the phase of selecting the necessary capabilities for the SoS and which systems can provide them.
Interoperability (IO)	Concern activities that deal with how constituent systems communicate and what messages and data will be exchanged among them to fulfill the SoS mission.
Emerging behaviors (EB)	Concern activities that describe emerging behaviors that can be identified already at design time, not requiring SoS simulation or execution to be observed.
Monitoring (MO)	Concern activities that should be planned to maintain the SoS operation.

The participants presented their opinions regarding the possibility of aggregating multiple heuristics (Q4). For example, defining interfaces at design time (H.12) is related to interoperability. This heuristic was embraced by others that mentioned communication standards, catalog of protocols used, datasets etc. Hence, it was combined with other heuristics that deal with those same issues. In addition, participants reported that it would be possible to aggregate heuristics related to interoperability, constituent systems capabilities, emergent behaviors, testing, and incentive mechanisms.

After the focus group dynamics and analysis, it was possible to adjust the set of heuristics, revising the statements and respective rationales, combining some heuristics that deal with similar purposes and removing from the set those heuristics that are not suitable for SoS design. The focus group discussions also made it possible to create a sequence in which the heuristics should be applied, grouping them into five categories, as described in Table 5: initiation heuristics (*IN*), constituent systems heuristics (*CS*), interoperability heuristics (*IO*), emerging behaviors heuristics (*EB*), and monitoring heuristics (*MO*). After the categorizations and the adjustments made through the focus group analysis, it was possible to build a refined version of the heuristics catalog for SoS design shown in Table 6.

4.3. Survey Results

For each heuristic, a question was proposed to know if the participant agreed, was neutral, or disagreed with the heuristic statement and if they had any comments about the statement formulation or suggestions to improve the heuristic statement. Figure 4 shows the respective percentages for each answer. The first bar chart represents the responses considering all 15 respondents (group 1), while the bar chart on the right represents only the responses of 10 researchers who declared having good or very good knowledge of SoS (group 2).

The responses and comments were reviewed for each of the heuristics. In general, the agreement regarding the heuristics was higher or equal to 50% in the two groups. We rewrote some heuristics based on the observations received. Also, according to the researchers, the SoS type of coordination was raised as an important factor when evaluating the applicability of each heuristic.

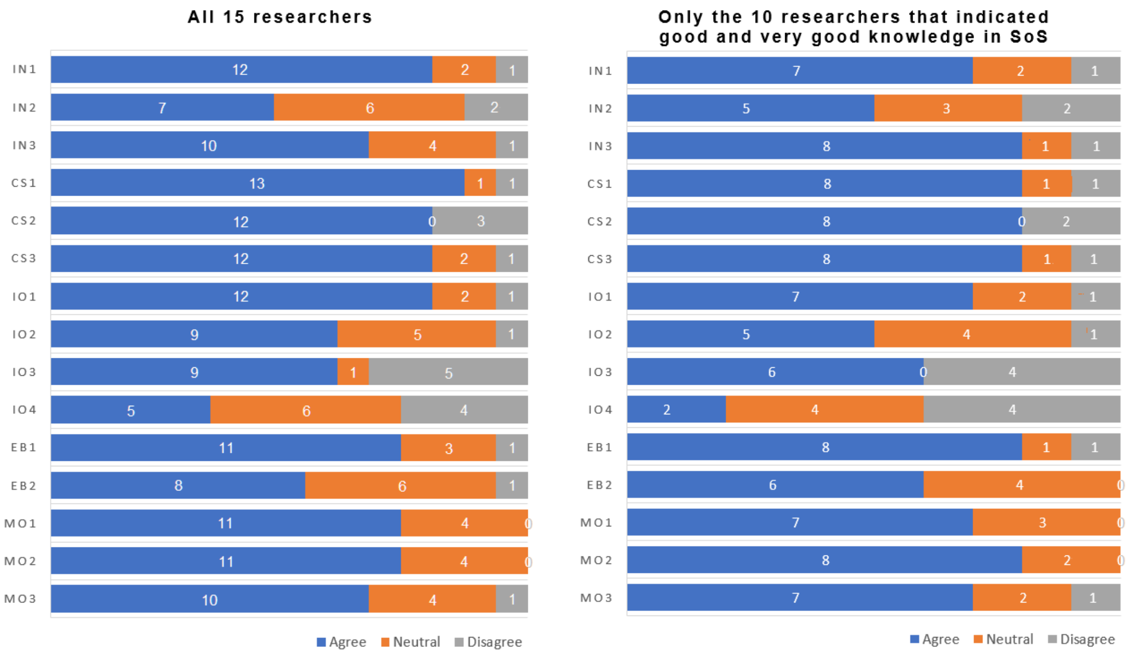

Figure 4. Responses to the survey.

Heuristic IN1 - The design should clearly identify who provides the necessary capabilities and resources for the operation of the SoS.

Table 6. Final catalog of heuristics.

ID	Description	Rationale
IN1	The design process should clearly identify who provides the necessary capabilities and resources for the operation of SoS.	An SoS depends on the synergy among systems that provide the necessary capabilities for operation, being necessary to guarantee adequate management for this supply.
IN2	The design should clearly consider who is responsible for the construction and operation of SoS.	When funding is required for the SoS operation, stakeholders should be informed how and by whom it will be done so that these issues are dealt with appropriately at the right time.
IN3	The design process should clearly identify who benefits from SoS.	Every SoS is built and operates for a purpose. It is important to identify who are the beneficiaries of the activities or operation of SoS.
CS1	The design process should clearly define which capabilities are already available and which should be implemented in constituent systems for the construction and operation of SoS.	The design process should foresee if there are enough functionalities available in constituent systems to collaborate with an SoS or if it is necessary to implement new functionalities.
CS2	The individual capabilities of each constituent system should be checked.	Each constituent system provide an SoS a capability that should be checked if it conforms to what is expected from it.
CS3	The design principles that generate the least possible disruption to SoS operation should be applied.	It is required to select which design alternative generate less disruption during an SoS operation, defining metrics that make it possible to evaluate the identified alternatives.
IO1	SoS coordination should ensure that each constituent system can exchange and understand data and messages exchanged among others.	It is necessary that the coordination of the SoS maintains the set of communication standards among constituent systems in order to ensure that each one can communicate with the others when necessary to achieve the purposes of SoS.
IO2	The interfaces of constituent systems should be defined at design time.	The interfaces of a constituent system are a crucial factor for the operation of SoS. They are points where the designer can influence.
IO3	SoS coordination should ensure that the policies for access and use of the capabilities of each constituent system can be exchanged and understood by other constituent systems.	It is necessary to ensure that each constituent system is able to participate in an SoS, respecting the access and use policies of other constituent systems. The management and dissemination of these policies are important due to independence characteristics of constituent systems and the dynamic SoS architecture.
IO4	All data sets to be exchanged among constituent systems should be defined.	The specification of all the data sets to be exchanged among constituent systems should be sufficient to use them to fulfill SoS missions.
EB1	Emergent behaviors should be allocated to constituent systems requirements.	The emergent behaviors necessary to fulfill SoS missions should be identified and explicitly associated with the respective constituent system or refinements responsible for them.
EB2	Weak emergent behaviors should be identified at design time.	Weak emergent behaviors are those that can already be identified in the design phase, not requiring SoS simulation or execution to be observed.
MO1	SoS missions should be periodically revised as the system evolves.	It is necessary to frequently monitor the fulfillment of the SoS missions since the dynamics of the SoS and the independence of constituent systems can bring problems to SoS over time.
MO2	The SoS design should include a feedback policy for the operation of an SoS.	It is necessary to monitor an SoS to detect problems during its operation and define the actions required to deal with them.
MO3	The interface patterns that emerged in the evolutionary process should be identified.	The evolutionary process may generate the need to use new communication standards or to update the existing standards. It is necessary to maintain the set of standards used as the evolution of constituent systems and SoS takes place, generating a roadmap for the process.

Most (80% in group 1 and 70% in group 2) agreed with the heuristic. From the answers and comments, it can be concluded that the heuristic was well mapped, although it is not regarded as valid for virtual SoS, where there is no type of coordination. Also, from the comments, it is noted that the understanding of what a heuristic is can vary depending on the researcher, and it is necessary to describe better the concept of heuristics used in this work.

Another interesting point brought up in the comments drew attention to the fact that this heuristic holds at the design phase, but when SoS begins operation, properties may appear that are not initially foreseen by designers. It may be difficult for this heuristic to remain valid over time while the SoS operates and evolve. Although it is important to know who provides the resources for the functioning of SoS, it is necessary to take into account that there will be unforeseen issues along its evolution cycle since SoS is never fully ready.

Heuristic IN2 - The design should clearly consider who is responsible for the construction and operation of the SoS.

Respondents agreed with this heuristic in the following proportion: 47% in group 1 and 50% in group 2. It can be noted from the comments that a highly evident characteristic is the unpredictability in the composition and evolution of the SoS. One of the comments reported the unpredictability of the applications developed and would be part of the Apple store, bringing a parallel with software ecosystems. The idea is that many events occur in a very automatic way in SoS.

A question about the type of SoS coordination was raised again, based on the fact that the different configurations that an SoS can assume can lead to behavior not foreseen by the designers. For example, in a convoy of autonomous vehicles, the decision of which vehicle goes ahead is made based on many factors like fuel consumption, routes, air resistance, and accident risks, resulting in unpredictable behaviors due to the complexity of all car systems having to work together.

Heuristic IN3 - The design should clearly identify who benefits from SoS.

A total of 67% of group 1 and 80% of group 2 agreed with heuristic IN3. The participants' observations show the dynamic scenario of the SoS, with the change in who benefits from it throughout its evolution, and the application of this heuristic is more appropriate to targeted SoS. The type of SoS coordination is treated again as mandatory for the application of the heuristic.

The comments showed that it might not be possible to identify who benefits from SoS. Furthermore, it is also necessary to understand if the benefit is at the business level, as in the case of a constituent system, or if it is a social benefit, as in the case of those who are not directly connected to the SoS.

Heuristic CS1 - Define which capabilities are already available and which should be implemented in the constituent systems for the construction and operation of SoS.

The major part of researchers agreed with this heuristic (87% in group 1 and 80% in group 2). Despite the high degree of agreement with this heuristic, comments were made about whether this statement is a heuristic or an activity for SoS design, reinforcing the need to define the adopted heuristic concepts better.

Another important question was about the choices of which capabilities exist, and therefore susceptible to reuse, and which will need to be developed. It noted that choosing to reuse available capabilities rather than constructing new ones should be evaluated by designers as these choices can lead to poor performance of the built SoS.

Heuristic CS2 - The individual capabilities of each constituent system should be checked.

For this heuristic, 80% of group 1, and 80% of group 2 agreed with the statement. Despite being well evaluated, this heuristic was criticized in the comments for not being an objective evaluation form, as informed in the definition of heuristic at the beginning of the questionnaire. In addition to the already mentioned need to better define what heuristic for this work is, it is also necessary to better define what is considered capability, despite the term being well known and applied in SoS.

Heuristic CS3 - Design principles that generate the least possible disruption to SoS operation should be applied.

For this heuristic, 80% agreed in group 1, and 80% agreed in group 2. Despite the high degree of agreement, it was suggested to mention which possible topologies an SoS can assume to decide on the most stable one. It was also mentioned that this heuristic is essential for critical systems, which may lead to further classification when considering the application of this heuristic. Once again, the importance of defining the type of coordination existing in the SoS was mentioned.

Heuristic IO1 - SoS coordination should ensure that each constituent system can exchange and understand data and messages exchanged among constituent systems.

The rate of agreement for this heuristic was 80% in group 1 and 70% in group 2. The SoS coordination issue was brought up again in the comments, and it is important to check how to address it in the heuristics catalog. It was suggested to split this heuristic into two parts: one to understand the data and the other to understand the messages, which should make the statement clearer.

Heuristic IO2 - The interfaces among constituent systems should be defined at design time.

The rate of agreement for this heuristic was 60% in group 1 and 50% in group 2. Although half of the participants in group 2 agree with this heuristic, this was one of the most explicit found in the literature. The characterization of the type of SoS coordination was again identified as essential for the application of this heuristic, needing to think about additional layers for connectivity in virtual SoS, not necessary for directed SoS. SoS coordination issues should be addressed in the review of the heuristics catalog.

Heuristic IO3 - SoS coordination should ensure that the policies for access and use of the capabilities of each constituent system can be exchanged and understood by other constituent systems.

For this heuristic, 60% agreed in both groups 1 and 2. Besides this, 40% of the group 2 participants disagree with this heuristic, which is a quite negative evaluation. Access policies were recognized as essential to help valuable data protection, but they were pointed out as unimportant for other classes of services like the internet, which the services work perfectly without knowing how each other works. Comments suggest that this heuristic is more adherent to directed SoS, not being possible to apply it to virtual SoS.

Heuristic IO4 - All data sets to be exchanged among constituent systems should be defined.

Only 33% of group 1 and 20% of group 2 agreed with this heuristic. According to the comments, applying this heuristic may restrict emergent properties in the SoS. Moreover, constituent systems in virtual SoS can completely ignore being part of the system, making it challenging to define their data sets. Another question was whether data sets or data formats should be defined.

IO4 was **not accepted** to be applied to any of the SoS types. For this reason, this heuristic was excluded from the final catalog.

Heuristic EB1 - Emerging behaviors should be allocated to constituent systems requirements.

Respectively, 73% and 80% agreed with this heuristic in group 1 and group 2. However, it was commented that the term ”emergent behavior” was not defined in the initial guidelines, and there was doubt whether the heuristic was saying that the behaviors would be observed or should be indicated in the constituent systems requirements.

Heuristic EB2 - Weak emerging behaviors should be identified at design time.

Respectively, 53% and 60% agreed with this heuristic in group 1 and group 2. A comment was made that this heuristic is more easily applied at design time than at the execution time, which, by the way, coincides with the proposal of this catalog. It was also indicated to apply this heuristic in a non-mandatory way.

Heuristic MO1 - SoS missions should be periodically revised as the system evolves.

For this heuristic, 73% and 70% agreed in group 1 and group 2. A comment shows an interpretation problem about whether the heuristic is an action that must take place to review the missions or whether it is a behavior to be observed as the SoS evolves. Other participants mentioned that this heuristic should be a quality criterion for SoS, and it is important to monitor it continuously. However, it was asked how to measure the mission accomplished, what could be used as evidence, and what is the period for these revisions.

Heuristic MO2 - The SoS design should include a feedback policy for the operation of the SoS.

Respectively, 73% and 80% agreed with this heuristic in group 1 and group 2. The only comment made was that it could be challenging to provide this type of feedback. This situation can lead to further detailing activities beyond the SoS design, such as actions required when problems are detected.

Heuristic MO3 - The interface patterns that emerged in the evolutionary process should be identified.

Group 1 agreed in 67%, and group 2 agreed in 70% with this heuristic. Again the SoS coordination is mentioned as a key factor to make it possible to apply the heuristic. Besides this, a metric was considered an important factor to make this heuristic applicable.

Table 7 summarizes the suitability of each heuristic by type of SoS coordination, taking into account the observations made by the survey participants.

Table 7. Suitability of heuristics for each type of SoS coordination.

ID	Heuristic	Directed	Acknowledged	Collaborative	Virtual
IN1	The design should clearly identify who provides the necessary capabilities and resources for the operation of the SoS.	✓	✓		
IN2	The design should clearly consider who is responsible for the construction and operation of the SoS.	✓	✓	✓	
IN3	The design should clearly identify who benefits from SoS.	✓	✓		
CS1	Define which capabilities are already available and which should be implemented in constituent systems for the construction and operation of SoS.	✓	✓	✓	
CS2	The individual capabilities of each constituent system should be checked.	✓	✓	✓	
CS3	Design principles that generate the least possible disruption to SoS operation should be applied.	✓			
IO1	SoS coordination should ensure that each constituent system can exchange and understand data and messages exchanged among constituent systems.	✓	✓		
IO2	The interfaces among constituent systems should be defined at design time.	✓			
IO3	SoS coordination should ensure that the policies for access and use of the capabilities of each constituent system can be exchanged and understood by the others.	✓	✓		
IO4	All data sets to be exchanged among constituent systems should be defined.				
EB1	Emerging behaviors should be allocated to constituent systems requirements.	✓			
EB2	Weak emerging behaviors should be identified at design time.	✓			
MO1	SoS missions should be periodically revised as the system evolves.	✓	✓		
MO2	The SoS design should include a feedback policy for the operation of the SoS.	✓	✓		
MO3	The interface patterns that emerged in the evolutionary process should be identified.	✓	✓	✓	

5. Discussion and Limitations

Information systems are increasingly present in the daily lives of people and companies, performing the most diverse functions such as financial services, health support, mobility, and even entertainment. These systems' capabilities make it increasingly unfeasible to build new systems from scratch, leading designers to think of solutions such as SoS that integrate the capabilities of several independent systems to deliver new functionality. However, the question is "*how to ensure a proper SoS operation?*".

Several techniques and tools are available for traditional systems to deal with all stages of their life cycle, covering design to eventual obsolescence. However, designing SoS can be challenging using techniques and tools intended for traditional systems because of the degree of uncertainty related to the independence of constituent systems. To deal with this challenge, we investigated heuristics that can be applied to SoS and produce a catalog that can be used at design time to support practitioners in dealing with issues that could negatively impact SoS. This catalog can help minimize problems and can support professionals in improving an SoS proper operation before the deployment.

Those heuristics were grouped into five categories. The Initiation category (*IN*) describes heuristics that can be used before the actual SoS design regarding issues such as who should provide financial and technical resources and who benefits from SoS operation. It is worth mentioning that it is not possible to state that the heuristics of the catalog apply to virtual SoS. Research on virtual SoS is still embryonic, and there is a lack of evidence of what can be applied to this type of SoS.

Regarding the constituent systems heuristics (*CS*), they are related to the concerns that the professionals should address to identify the required capabilities and how to acquire and maintain the constituent systems to provide the required capabilities. Aspects such as availability, reliability, price, data quality, and possible constituent systems replacement can be raised with support of this heuristics.

Interoperability heuristics (*IO*) deal with how constituent systems will communicate with the SoS. It helps to define the communication standards necessary to guarantee the collaboration of each constituent systems to an SoS, maintain a collection of constituent systems interfaces, and raise the policies involved in accessing and using data and services from constituent systems.

Emergent behaviors heuristics (*EB*) deals with identifying the emergent behaviors necessary for SoS operation early in the design phase and allocating them to the respective set of constituent systems, avoiding complicated simulation processes in SoS.

Monitoring heuristics (*MO*) helps SoS professionals to plan the resources that must be monitored. This is especially important due to the dynamics involved in SoS operation, which can be affected by a new configuration of a constituent system, for example. Moreover, monitoring is crucial in providing feedback to maintain the communication standards and policies necessary for SoS proper operation.

5.1. Implications

A catalog of heuristics has an important role for researchers. It helps expand the body of knowledge by identifying and organizing a comprehensive set of specific principles for SoS design. This provides a foundation for future studies and helps researchers better understand how to design SoS. Additionally, the catalog guides research by pointing out areas that require additional investigation. It can also be used to assess how effective SoS design is in different situations, making analyses and comparisons easier.

As practical implication, the catalog can help systems engineers working with SoS as it offers clear guidance for design, ensuring that important aspects are considered. Moreover, the heuristics help identify issues and challenges, allowing project teams to take proactive measures. This can lead to standardizing best practices in SoS design, making the process more consistent across different projects.

5.2. Threats and Limitations

In contrast to quantitative studies, qualitative research is typically more vulnerable to threats to credibility than to validity [Greiler et al. 2023, Ribeiro et al. 2022]. The assurance of validity and reliability in qualitative research depends on the scrupulousness, comprehensiveness, and integrity exhibited by researchers during both the data collection and analysis processes [Robson 2002]. Consequently, we delineate potential threats to both external and internal credibility below.

Internal credibility pertains to the credibility of interpretations and conclusions within the specified context or group [Onwuegbuzie and Leech 2007]. In this study, interpretive validity poses a potential threat to internal credibility, encapsulating the risk that researchers impose their interpretations rather than truly understanding the perspectives of participants. This threat was mitigated by posing clear questions to survey participants and encouraging deep reflection on opinions during the focus group, aiming for an authentic interpretation of gathered information. Additionally, the interviews were conducted in Portuguese, the native language of the participants.

In terms of analysis, we applied coding to interview transcriptions, a systematic method of interpreting and analyzing data to ensure consistent evaluation of responses, thus minimizing the risk of bias or interpretive errors.

Recognizing that the participant pool may not be sufficiently representative to generalize results, we addressed this by deliberately selecting professionals with diverse backgrounds. This deliberate choice contributed to a more extensive array of information, incorporating diverse perspectives.

Recognizing the potential influence of survey and focus group format on participant responses and its consequent impact on **reliability**, we took proactive measures. Thorough discussions among the authors were conducted to refine study protocols. During the focus group, the researcher attentively considered aspects such as voice intonation and body language, guiding the interview process with additional questions to ensure a comprehensive capture of participants' perceptions.

Regarding the limitations of our study, the completeness of the SMS may have been affected due to the lack of relevant primary studies. To mitigate this, we conducted searches in the most relevant publications in engineering and computer science databases. However, some studies may have been missed due to limitations of the search engines or these studies do not present the terms included in our search string in the title, abstract, and keywords. To assure the selection of those relevant primary studies, we defined a detailed protocol following well-established guidelines for SMS [Kitchenham and Charters 2007]. Moreover, the authors of this paper are researchers in SoS and are aware of no bias that may have been introduced during the conduction of this SMS. However, it might be possible that our background had influenced the analysis of studies. Moreover, not all primary studies detailed information to be extracted to answer our research question. Therefore, we had to infer certain pieces of information in our analysis. To minimize the inaccuracy of such inferences, we conducted several discussions in the focus group aiming at solving any disagreement and clarifying potential ambiguities. Furthermore, by considering only studies published in academic databases, it is not possible to generalize the results. We did not consider all possible solutions created by industry and possibly published in the grey literature (such as technical reports, white papers, and web blog postings). In addition, the small number of participants in the focus group may limit the generalization of the findings.

One of the key advantages of utilizing focus groups is that they provide participants with the opportunity to collaboratively explore ideas, resulting in the emergence of fresh insights on a specific subject. However, while a smaller number of participants can facilitate in-depth discussions on a particular topic, it would be beneficial to consider including more participants in a focus group to enrich the discussions regarding the proposed heuristics.

Finally, despite the survey is a useful tool for research. However, recruiting practitioners and researchers to respond to surveys was difficult due to the required background knowledge. In addition, there are other barriers, such as respondents considering invitations as unwanted email messages, such as an invasion of privacy or "junk mail", leading to a low response rate.

6. Conclusion

This article contributes SoS research with a catalog of heuristics for SoS design. To do so, we performed an SMS to identify the heuristics. The SMS findings were evaluated through a focus group that allowed us to organize an initial set of fifteen heuristics. This set of heuristics was evaluated through a survey with experts. After the analysis of the survey results, we reached a consolidated catalog with fourteen heuristics grouped into five categories,i.e. three for Initiation (IN), three for Constituent Systems (CS), three for Interoperability (IO), two for Emerging Behaviors (EB), and three for Monitoring (MO).

A catalog of heuristics can help professionals anticipate problems and concerns still at design time, making it possible to mitigate risks before the SoS operation. Resources and functionalities can be planned, and design decisions can be made, considering a more detailed scenario. As future work, we will develop a tool to assist professionals during the SoS design. From an SoS model, the tool can apply the heuristics catalog to verify interoperability, emergent behavior, and monitoring aspects.

Acknowledgements

This study was partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001, FAPERJ (Proc. 211.583/2019 & Proc. 210.688/2019) and UNIRIO (DPq/PPQ 2022 & 2023).

References

[Axelsson 2022] Axelsson, J. (2022). Systems-of-systems design patterns: A systematic literature review and synthesis. In *System of Systems Engineering Conference (SoSE)*, pages 171–176, Rochester, EUA.

[Batista 2013] Batista, T. (2013). Challenges for SoS architecture description. In *International Workshop on Software Engineering for Systems-of-Systems (SESoS)*, pages 35–37, Montpellier, France.

[Beale et al. 2022] Beale, D., McKinney, D., Oosthuizen, R., Smith, G., and Watson, M. D. (2022). System engineering heuristics for complex systems. In *INCOSE International Symposium*, volume 32, pages 293–312. Wiley Online Library.

[Beale et al. 2023] Beale, D., Oosthuizen, R., McKinney, D., Smith, G., and Watson, M. (2023). Systems engineering heuristics for complex systems revisited. In *System of Systems Engineering Conference (SoSE)*, pages 1–7, Lille, France.

[Beck and Manuel 2008] Beck, S. E. and Manuel, K. (2008). *Practical research methods for librarians and information professionals*. Neal-Schuman Publishers New York.

[Boardman and Sauser 2006] Boardman, J. and Sauser, B. J. (2006). System of systems - the meaning of of. In *International Conference on System of Systems Engineering (SoSE)*, pages 1–6, Los Angeles, USA.

[Bouziat et al. 2018] Bouziat, T., Camps, V., and Combettes, S. (2018). A cooperative sos architecting approach based on adaptive multi-agent systems. In *International Workshop on Software Engineering for Systems-of-Systems (SESoS)*, pages 8–16, Gothenburg, Sweden. IEEE.

[Butterfield et al. 2008] Butterfield, M., Pearlman, J., and Vickroy, S. (2008). A System-of-Systems Engineering GEOSS: Architectural Approach. *IEEE Systems Journal*, 2(3):321–332.

[Caplan 1990] Caplan, S. (1990). Using focus group methodology for ergonomic design. *Ergonomics*, 33(5):527–533.

[Chueri 2021] Chueri, L. (2021). *SIDE: A Framework for managing social innovation digital ecosystems*. PhD thesis, Universidade Federal do Estado do Rio de Janeiro.

[Churcher et al. 2007] Churcher, N., Frater, S., Huynh, C. P., and Irwin, W. (2007). Supporting OO design heuristics. In *Australian Software Engineering Conference (ASWEC)*, pages 101–110, Melbourne, Australia. IEEE.

[Corbin and Strauss 2008] Corbin, J. and Strauss, A. (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. SAGE, 3 edition.

[da Silva Amorim et al. 2014] da Silva Amorim, S., de Almeida, E. S., McGregor, J. D., and von Flach G. Chavez, C. (2014). When ecosystems collide: making systems of systems work. In *ECSA*, pages 1–4, Vienna, Austria.

[Dahmann et al. 2008] Dahmann, J., Rebovich, G., and Lane, J. (2008). Systems engineering for capabilities. *CrossTalk*, 21:7.

[Dersin 2014] 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.

[Dhawan 2020] Dhawan, S. (2020). Online learning: A panacea in the time of COVID-19 crisis. *Journal of Educational Technology Systems*, 49(1):5–22.

[Dietterle 2005] Dietterle, R. (2005). The future combat systems (FCS) overview. In *Military Communications Conference (MILCOM)*, pages 3269–3273 Vol. 5, Atlantic City, USA.

[Ferreira et al. 2021] Ferreira, F. H., Nakagawa, E. Y., and Santos, R. P. (2021). Reliability in software-intensive systems: Challenges, solutions, and future perspectives. In *Euromicro SEAA*, pages 54–61, Palermo, Italy.

[Garcés and Nakagawa 2017] Garcés, L. and Nakagawa, E. Y. (2017). A process to establish, model and validate missions of systems-of-systems in reference architectures. In *SAC*, pages 1765–1772, Marrakech, Morocco.

[Gigerenzer and Gaissmaier 2011a] Gigerenzer, G. and Gaissmaier, W. (2011a). Heuristic decision making. *Annual Review of Psychology*, 62(1):451–482.

[Gigerenzer and Gaissmaier 2011b] Gigerenzer, G. and Gaissmaier, W. (2011b). Heuristic decision making. *Annual review of psychology*, 62:451–482.

[Gonçalves et al. 2015] Gonçalves, M. B., Oquendo, F., and Nakagawa, E. Y. (2015). A meta-process to construct software architectures for system of systems. In *Symposium on Applied Computing (SAC)*, pages 1411–1416, Salamanca, Spain.

[Greiler et al. 2023] Greiler, M., Storey, M., and Noda, A. (2023). An actionable framework for understanding and improving developer experience. *IEEE Transactions on Software Engineering*, 49(04).

[Hacker et al. 2020] Hacker, J., Vom Brocke, J., Handali, J., Otto, M., and Schneider, J. (2020). Virtually in this together – how web-conferencing systems enabled a new virtual togetherness during the COVID-19 crisis. *European Journal of Information Systems*, 29(5):563–584.

[Imamura et al. 2023] Imamura, M., Ferreira, F. H., Fernandes, J., Neto, V. V. G., and dos Santos, R. P. (2023). Towards a catalog of heuristics for the design of systems-of-systems. In da Cunha, M. X. C., de Souza Júnior, M. F., Marques, J. C., de Classe, T. M., and Araújo, R. D., editors, *Proceedings of the XIX Brazilian Symposium on Information Systems, SBSI 2023, Maceió, Brazil, 29 May 2023- 1 June 2023*, pages 128–135. ACM.

[Kazman et al. 2013] Kazman, R., Schmid, K., Nielsen, C. B., and Klein, J. (2013). Understanding patterns for system of systems integration. In *International Conference on System of Systems Engineering (SoSE)*, pages 141–146, Hawaii, USA. IEEE.

[Keating et al. 2003] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., Peterson, W., and Rabadi, G. (2003). System of Systems Engineering. *Engineering Management Journal*, 15(3):36–45.

[Kitchenham et al. 2015] Kitchenham, B., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*, volume 4. CRC Press.

[Kitchenham and Charters 2007] Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

[Liang and Rubin 2012] Liang, Q. and Rubin, S. H. (2012). Randomisation in designing software tests for systems of systems. *International Journal of Information and Decision Sciences*, 4(2-3):108–129.

[Lopes et al. 2016] Lopes, F., Loss, S., Mendes, A., Batista, T., and Lea, R. (2016). Sos-centric middleware services for interoperability in smart cities systems. In *Middleware*, pages 1–6, Trento, Italy.

[Maier 1998] Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284.

[McDermott 2018] McDermott, T. (2018). Developing systems thinking skills using healthcare as a case study. In *International Conference on System of Systems Engineering (SoSE)*, pages 240–244, Paris, France. IEEE.

[Mittal and Rainey 2015] Mittal, S. and Rainey, L. (2015). Harnessing emergence: The control and design of emergent behavior in system of systems engineering. In *Proceedings of the Conference on Summer Computer Simulation (SummerSim)*, pages 1–10, San Diego, USA.

[Nielsen et al. 2015] Nielsen, C., Larsen, P., Fitzgerald, J., Woodcock, J., and Peleska, J. (2015). Systems of systems engineering: Basic concepts, model-based techniques, and research directions. *ACM Computing Surveys*, 48(2).

[Nielsen 1994] Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI)*, pages 152–158, Boston, USA.

[Onwuegbuzie and Leech 2007] Onwuegbuzie, A. J. and Leech, N. L. (2007). Validity and Qualitative Research: An Oxymoron? *Quality Quantity*, 41(2):233–249.

[Oquendo 2016] Oquendo, F. (2016). Formally describing the software architecture of systems-of-systems with sosadl. In *System-of-Systems Engineering Conference (SoSE)*, pages 1–6, Kongsberg, Norway.

[Raman and Murugesan 2022] Raman, R. and Murugesan, A. (2022). Framework for complex sos emergent behavior evolution using deep reinforcement learning. *INCOSE International Symposium*, 32(1):809–823.

[Ribeiro et al. 2022] Ribeiro, B. B., Costa, C., and Pereira dos Santos, R. (2022). Understanding and analyzing factors that affect merge conflicts from the perspective of software developers. *Journal of Software Engineering Research and Development*, 10:12:1 – 12:17.

[Ricci et al. 2013] Ricci, N., Ross, A. M., and Rhodes, D. H. (2013). A generalized options-based approach to mitigate perturbations in a maritime security system-of-systems. *Procedia Computer Science*, 16:718–727.

[Riel 1996] Riel, A. (1996). *Object-Oriented Design Heuristics*. Addison-Wesley.

[Robson 2002] Robson, C. (2002). *Real World Research - A Resource for Social Scientists and Practitioner-Researchers*. Oxford: Blackwell Publishing.

[Sage and Cuppan 2001] Sage, A. P. and Cuppan, C. D. (2001). On the systems engineering and management of systems of systems and federations of systems. *Information knowledge systems management*, 2(4):325–345.

[Schneider et al. 2014] Schneider, J.-P., Teodorov, C., Senn, E., and Champeau, J. (2014). Towards a dynamic infrastructure for playing with systems of systems. In *European Conference on Software Architecture (ECSA)*, pages 1–4, Vienna, Austria.

[Sherman and Corty 1984] Sherman, S. J. and Corty, E. (1984). Cognitive heuristics.

[Singh et al. 2020] Singh, R. P., Javaid, M., Kataria, R., Tyagi, M., Haleem, A., and Suman, R. (2020). Significant applications of virtual reality for COVID-19 pandemic. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 14(4):661–664.

[Sunstein 2005] Sunstein, C. R. (2005). Moral heuristics. *Behavioral and brain sciences*, 28(4):531–541.

[Westmattelmann et al. 2020] Westmattelmann, D., Grotenhermen, J., Sprenger, M., and Schewe, G. (2020). The show must go on - Virtualisation of sport events during the COVID-19 pandemic. *European Journal of Information Systems*, pages 1–18.

[Weyns and Andersson 2013] Weyns, D. and Andersson, J. (2013). On the challenges of self-adaptation in systems of systems. In *International Workshop on Software Engineering for Systems-of-Systems (SESoS)*, pages 47–51, Montpellier, France.

[Wickramasinghe et al. 2007] Wickramasinghe, N., Chalasani, S., Boppana, R., and Madni, A. (2007). Healthcare system of systems. In *International Conference on System of Systems Engineering (SoSE)*, pages 1–6, San Antonio, USA.

[Zaganelli et al. 2015] Zaganelli, B. M., Nisenbaum, M. A., Alves, K. d. S. G., Marques, S. B., and Olinto, G. (2015). O grupo focal na ciência da informação. *Informação & Sociedade*, 25(3).