

# System of Systems Requirements: A Systematic Literature Review using Snowballing

Renata Martinuzzi de Lima<sup>1</sup>, Daniel de Vargas<sup>1</sup>, Lisandra Manzoni Fontoura<sup>1</sup>

<sup>1</sup> Universidade Federal de Santa Maria (UFSM) - Santa Maria, RS, Brazil

{rlima, ddvargas, lisandra}@inf.ufsm.br

**Abstract** - System of Systems (SoS) is a subject of interest for many researchers from different research lines. System of Systems Engineering (SoSE) deals with many challenges mainly in the activities related to requirements. In order to explore studies related to Requirements Engineering (RE) for SoS, this Systematic Literature Review, using the Snowballing approach, aims to answer four research questions. These questions are related to the activities of translating capability objectives in SoS requirements, SoS requirements specification, SoS requirements management during the evolutionary process of a SoS, and as the main goal, understanding the research challenges of RE for SoS which still need to be explored.

**Keywords** - *System of Systems; Requirements Engineering; Capability Objectives; Research challenges.*

## I. INTRODUCTION

In different application domains, it is possible to find distinct systems working together to satisfy a specific goal. It represents what is named as “System of Systems” (SoS). System of Systems Engineering (SoSE) is a subject of increasing interest within the Systems Engineering (SE) community. However, structural differences, the diversity of stakeholders in a SoS development project leads to a complexities and challenges still unexplored in SE [7].

Requirements Engineering (RE) is one of the most critical and difficult tasks in the development of any kind of system. Therefore, in SoS context the complexity, problems and challenges increase. The traditional RE for individual system is still a big challenge for system engineers and managers, but it is already quite explored and there are many techniques proposed with success. However, in the context of SoS Engineering, these techniques from SE not always can be perfectly applied to SoSE. Hence, it is necessary studies to develop new methods, techniques or processes specifically directed to address demands on requirements for SoS [5].

To begin new studies in this topic, it is interesting an investigation about how the subject has been treated by researchers in the last years. Many reviews and surveys can be found in the literature of SoS field, however there is no evidence on Systematic Literature Reviews (SLR) or Systematic Mapping Reviews (SMR) specific on Requirements for SoS. Therefore, the main goal of this SLR is to explore studies related to RE for SoS, using the Snowballing approach, in order to find papers that answer four research questions.

This paper presents the results of a SLR in studies published from 2005 to 2016 and it was conducted following the Snowballing approach. The section II presents a background about Requirements Engineering for SoS; Section III has the details about the method used in the SLR; Section IV presents the results and Section V the conclusions.

## II. BACKGROUND

A classic definition, given by [6], says that System of Systems (SoS) is a collection of components that individually must be recognized as systems and may have two fundamental characteristics: i) Operational Independence of Components and ii) Managerial Independence of Components. In addition, it defines six main features of a SoS: Operational Independence, Managerial Independence, Geographic Distribution, Evolutionary Development and Emergent Behavior [6].

There are seven core elements of SoS Engineering (SoSE), defined by [3], which deal with the application of the SE processes in the SoS context. Three of these seven core elements are directly related to requirements: “Translating Capability Objectives”, “Developing and Evolving a SoS Architecture”, and “Addressing Requirements and Solution Options”. The first one describes one of the most important tasks of SoSE that aims to develop an initial understanding of the capability objectives, which may be stated in high level and based on desired operational tasks or missions.

Traditional Systems Engineering (SE) has its methods and processes well defined, thus the elicitation and definition of requirements becomes a trivial task. However, assuming that SoSE is not only a extension of SE, it is true to say that SoS requirements definition practices are also not a simple extension of traditional RE [5]. In this sense, RE for SoS involves the development of the tasks of “translating capability objectives into requirements”, as well as the use of methods, tools, and processes that support this task. Also, the management of requirements throughout the development and evolution of a SoS.

## III. METHODOLOGY

SLR is a specific research methodology that is developed with the purpose of gathering and evaluating studies in a particular topic of interest and using a specific approach [1]. The snowballing approach is a way to develop a SLR using the reference list and the citations of a specific paper in order

to identify additional studies about that subject, looking at where papers are actually referenced and cited [8].

The planning phase of a SLR includes the formulation of the research questions, the definition of the approach that will be used, and the definition of the inclusion and exclusion criteria [4]. In this SLR, the research questions are the following:

*(RQ1) Which papers are related to the translation of capability objectives in high-level SoS requirements?*

*(RQ2) Is there some way commonly used to specify SoS requirements?*

*(RQ3) Which papers are related to the management of the requirements changes in the evolutionary process of a SoS?*

*(RQ4) What are the research challenges found in Requirements Engineering for SoS?*

In addition, in the snowballing approach, it is necessary to define a tentative of Start Set, that is an initial set of papers which will be analyzed in the backward and forward snowballing. The first attempt of a start set has been defined based on studies that were used in a previously research.

After that, it has been defined the inclusion and exclusion criteria, in the context of the SLR. For inclusion: the paper should have been published between 2005 and 2016; the title, keywords, or abstract should make explicit that the paper is related to the research topic; and the paper should answer, in some way, at least one of the research questions. For exclusion: The publication should not be a tutorial, workshop, only an abstract or a technical report; the full text should be available; and redundant publications should be excluded.

The second phase begins with the start set. In the first interaction, the backward and forward snowballing analyze each paper. The papers included in the first interaction are analyzed in the second interaction and so on. Backward Snowballing analyzes the reference list to identify new papers to include in the SLR. First, the papers are excluded based on the exclusion criteria cited above, and then, based on title, abstract, and conclusion, candidates papers are reserved to be more carefully analyzed [8]. Forward Snowballing refers to identifying new papers to include by analyzing the list of papers citing the paper being examined. Using the Google Scholar search tool, the first screening is done based only on the information available on Google. If this information is not sufficient to decide whether the paper is a candidate or not, it is analyzed more carefully applying the exclusion and inclusion criteria [8].

The initial start set had 5 papers identified by [A1] to [A5] to begin the first interaction. In the first interaction, after backward and forward snowballing, 28 candidates were identified and 7 papers were included ([A6] to [A12]). In the second interaction, 7 candidates were examined and 2 papers were included ([A13] and [A14]). In the third interaction, from 4 candidates, 3 papers were included ([A15] to [A17]). In the fourth interaction, after examining 3 candidates, more 2 papers were included ([A18] and [A19]). Since after the fifth interaction no new paper has been identified, the loop is finished and the data extraction phase could be initiated. All the papers included are identified on Table II.

Data extraction was done through the Reading Sheets that gather information such as: title, authors, year, country and place of publication, the main research topic, objective, type of article, application context, as well as the relevant information that answers the research questions.

#### IV. RESULTS

At the end of the execution phase, 19 papers met the inclusion criteria and their data were extracted. Table II shows the details of all the papers included in the SLR with their IDs, titles, authors and year of publication. We also present the calculation of the efficiency and the citation matrix of the Snowballing approach, both suggested by [8], and then the results according to each Research Question.

##### A. Efficiency

An important efficiency measure for Systematic Literature Reviews is to analyze the number of papers included in relation to the total number of candidates examined. WOHLIN [8] emphasizes that efficiency is calculated based on the candidates. For example, if we remove those candidates from backward snowballing where the exclusion decision is taken by the year of publication or title in the reference list, efficiency increases. In forward snowballing, the analysis is done in the same way, based on the observation of the data found in Google Scholar, so efficiency can increase in general.

Therefore, the efficiency of this work (24%) can be considered high because the number of candidates is low due to the research topic is very limited. Thus, the analysis of the efficiency in the different interactions is shown in Table I.

TABLE I. EFFICIENCY

<i>Interaction</i>	<i>Candidates</i>	<i>Included</i>	<i>Efficiency</i>
Start Set	7	5	5/7 = 71%
First Interaction	23	7	7/23 = 30%
Second Interaction	6	2	2/6 = 33%
Third Interaction	4	3	3/4 = 75%
Fourth Interaction	3	2	2/3 = 66%
Fifth Interaction	0	0	0
<b>Total = 24% of efficiency</b>			

##### B. Citation Matrix

To understand the references and citations among the included papers, a citation matrix was created, based on [8] suggestion. Table III shows how the 19 papers reference to each other, denoted by an "x" (e.g. paper [A2] references to papers [A3] and [A4] and it is cited by [A11]). Table III is complemented with information on the possibility of citation, denoted by a "-". For example, paper [A4] does not mention any other of the included papers, since by checking the year of publication it can be seen that papers marked with "-" could not be referenced because they have not yet been published. However, cells left in blank refer to papers that were published prior to [A4] and could have referenced them.

##### C. RQ 1- Which papers are related to the translation of capability objectives in high-level SoS requirements?

[A3] defines three core elements of SoSE related to requirements: "Translating Capability Objectives," "Developing and Evolving an Architecture", and "Meeting Solution Requirements and Options." [A8] provides guidance for defining capability objectives, it defines SoS

Engineering methods, processes and tools tailored to support this activity. Finally, [A15] proposes a capability-based approach that mathematically explores the structural

semantics of representing user needs for formulating requirements in complex systems.

TABLE II. ALL THE PAPERS INCLUDED IN THE SLR

ID	Authors	Year	Title
[A1]	BOARDMAN, John; SAUSER, Brian	2006	System of Systems - the meaning of of.
[A2]	DAHMAN, J. S. et al	2010	Systems Engineering Artifacts for SoS.
[A3]	DoD-USA	2008	Systems Engineering Guide for Systems of Systems.
[A4]	DAHMAN, Judith S.; BALDWIN, Kristen J	2008	Understanding the current state of US defense systems of systems and their implications for systems engineering.
[A5]	NIELSEN, Claus Ballegaard et al	2015	SoSE: basic concepts, model-based techniques, and research directions.
[A6]	LOCK, Russell	2012	Developing a methodology to support the evolution of SoS using risk analysis.
[A7]	CECCARELLI, Andrea et al	2015	Introducing Meta-Requirements for Describing System of Systems.
[A8]	LANE, Jo Ann	2014	System of systems capability to requirements engineering.
[A9]	HOLT, Jon et al	2015	A model-based approach for requirements engineering for systems of systems.
[A10]	VIERHAUSER, Michael et al	2015	A requirements monitoring model for systems of systems.
[A11]	HALLERSTEDTE, Stefan et al	2012	Technical challenges of SoS requirements engineering.
[A12]	HOLT, Jon et al	2012	Model-based requirements engineering for system of systems.
[A13]	LEWIS, Grace A. et al	2009	Requirements engineering for systems of systems.
[A14]	KEATING, Charles B.; PADILLA, J.; ADAMS, K.	2008	System of systems engineering requirements: challenges and guidelines.
[A15]	RAVICHANDAR, R.; ARTHUR, J. D.; BOHNER, S.	2007	Capabilities engineering: Constructing change-tolerant systems.
[A16]	MACDIARMID, Alisdair; LINDSAY, Peter	2010	Can system of systems be given self-x requirement engineering capabilities?
[A17]	WALKER, Randy G.; KEATING, Charles B	2012	Defining SoS requirements: an early glimpse at a methodology.
[A18]	KATINA, Polinpapilinho F.; KEATING, Charles B.; RA'ED, M. Jaradat	2014	System requirements engineering in complex situations.
[A19]	WALKER, Randy G. et al	2014	A method to define SoS requirements.

TABLE III. CITATION MATRIX

Ref.	Citation																			
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	
A1																				
A2			X	X																
A3				X																
A4																				
A5	X		X	X							X	X								
A6	X																			
A7	X			X																
A8			X																	
A9			X	X								X	X							
A10				X										X						
A11		X											X							
A12			X										X							
A13			X																	
A14																				
A15																				
A16													X	X						
A17			X											X					X	
A18			X											X						
A19			X											X					X	

D. RQ 2 - Is there some way commonly used to specify SoS requirements?

[A2] identifies 14 ESoS artifacts, including requirements-related artifacts: "Capability Objectives", "CONOPS", "Information about Systems That Impact SoS Capability Objectives" and "Requirements Space". Meanwhile, [A7] proposes a model of meta-requirements to describe points of view of a SoS and relate them. This model was developed to be used in the derivation of requirements of any SoS.

[A9] is an evolution of the research presented in [A12] and both of them present a model-based approach to RE for SoS which represents the best practices is SoS in terms of proven standards and research derived from a model-based SE approach to requirements engineering. Another RE method for SoS is presented by [A13], and this brings the following top-

down and bottom-up activities: identify SoS context, identify SoS and individual system goals, understand SoS interactions, identify system capabilities and constraints and analyze the gap.

[A14] provides a guideline within SoSE efforts exploring the nature of requirements from a SoSE perspective and establishing a foundation for differences between the SE and SoSE problem domains. Moreover, [A16] explores a vision of how the key artifacts of requirements engineering need to evolve with supporting tools and processes in order to support the development of a SoS, it also lists the four types of SoS and illustrates their relationship among RE and each of them.

[A18] presents an alternative perspective for systems thinking-based and an approach for requirements elicitation in complex situations, the case of a SoS, exploring some broad challenges associated with the requirements engineering elicitation.

The same authors published [A17] and [A19]. While [A17] brings a methodology proposal, still in development, that combines top-down and bottom-up approaches to the derivation of SoS requirements, [A19] provides an update to the [A17], now renaming the "methodology" by "method".

*E. RQ 3 - Which papers are related to the management of the requirements changes in the evolutionary process of a SoS?*

[A1] points out important characteristics that must be considered on the requirements management in the evolutionary process. While [A3] introduces the core elements: "Developing and Evolving a SoS Architecture" and "Requirements management" which are related to the changing management in the evolutionary process of a SoS.

[A6] proposes a methodology to support the identification, organization and discussion of needed information to manage the evolution of a SoS in terms of requirements. Meanwhile, [A10] describes a three-dimensional requirements monitoring model for SoS, an essential task after updating certain components or constituent systems in the evolutionary process.

Then, [A15] proposes an alternative approach to developing complex and emerging systems that need to be change tolerant, considering that they have long development cycles. Also, [A16] suggests that managing the requirements evolution must be an autonomous process.

*F. RQ 4 - What are the research challenges found in Requirements Engineering for SoS?*

Paper [A3] concludes that some SoSE issues, particularly in the military field, still need to be addressed including options for managing SoS that would facilitate SoSE and ensure more predictable progress, effective ways of achieving SoS evolution, strategies to effectively integrate constituent systems into a viable, evolving, and in some cases ad-hoc SoS.

[A4] suggests some areas for further investigation into SoS management, also in the military field, especially on the existing need to clarify the management relationships between SoS and constituent systems. Moreover, in the technical context, the authors cite, as an important role of SoSE, the need to create mechanisms that can anticipate the changes in the requirements and evaluate the implications of these changes with the constituent systems managers and engineers.

Paper [A5] identifies that the need for interoperability in a SoS adds some important requirements in terms of analysis and modeling methods, which increases a need for techniques that support a verification of requirements that serve the constituent systems. Also, [A5] states that many system-level tests fail because the emerging properties were insufficiently captured during the requirements development phase.

[A7] identifies some challenges from the perspective of SoSE requirements analysis. It highlights the following assumptions: i) "The requirements of time, reliability, and security can become more complex when the focus is on SoS as a whole". ii) "As a SoS is an evolutionary system, adaptability and flexibility should be considered in the short and long term, in which case specific SoSE approaches should be reviewed". iii) "A set of SoS requirements requires an understanding of a potentially large number of scenarios, where

boundaries can be difficult to define, to be able to build a SoS, engineers must understand where to set boundaries and how large and detailed their view of SoS must be".

The paper [A11] lists the main technical activities of requirements, validation, tracing and verification processes with the main characteristics of a SoS (independence, distribution, emergency and evolution) which allows to identify several technical challenges in RE for SoS.

[A13] discusses some RE for SoS challenges in the following contexts: scale, multiple domains, varied operational context, decentralized control, fast evolving environments, continuous and disconnected execution of the different phases of the life cycle and the needs for collaboration and integration.

[A14] indicates key implications for the development and use of requirements in SoSE in practice. (e.g. the nature of the SoSE domain problem suggests that requirements are simultaneously "loose and tight"; the resolution of requirements should increase with further understanding of the SoS domain problem and emerging conditions; SoS requirements and constituent system requirements belong to different classes; finally, that a balance must be reached for SoS requirements).

## V. CONCLUSION

This SLR pursued to understand how Requirements Engineering is planned and executed within the context of System of Systems. More specifically, this paper aimed to search for research challenges that still need to be addressed in the topic. Nineteen studies were included at the end of the SLR and analyzed through reading sheets.

This work is an interesting source for researchers who want to comprehend the challenges or gaps on the field of SoS Requirements and their implications for SoS Engineering. As a future work, we intend to explore one or more of the research challenges encountered in order to contribute to the field of research in Requirements Engineering for System of Systems.

## ACKNOWLEDGMENT

CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*) supported this work.

## REFERENCES

- [1] BIOLCHINI, Jorge et al. Systematic review in software engineering. System Engineering and Computer Science Department COPPE/UF RJ, Technical Report ES, v. 679, n. 05, p. 45, 2005.
- [2] DAHMANN, J. S. and Baldwin, K. J. (2008). Understanding the current state of us defensesystems of systems and the implications for systems engineering. In Systems Conference, 2008 2nd Annual IEEE, pages 1–7.
- [3] DoD-USA (2008). Systems Engineering Guide for Systems of Systems. Washington, DC.
- [4] KITCHENHAM, Barbara. Procedures for performing systematic reviews. Keele, UK, Keele University, v. 33, n. 2004, p. 1-26, 2004.
- [5] LEWIS, Grace A. et al. Requirements engineering for systems of systems. In: Systems Conference, 2009 3rd Annual IEEE. IEEE, 2009. p. 247-252.
- [6] MAIER, Mark W. Architecting principles for systems-of-systems. In: INCOSE International Symposium. 1996. p. 565-573.
- [7] BKCASE Editorial Board. 2016. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.7. R.D. Adcock (EIC). Hoboken, NJ.
- [8] WOHLIN, Claes. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. ACM, 2014. p. 38.