



Characterizing Architectural Evaluations and Identifying Quality Attributes addressed in Systems-of-Systems: A Systematic Mapping Study

Diego Gatica, Francisco Ponce, René Noël and Hernán Astudillo

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

November 15, 2018

Characterizing Architectural Evaluations and Quality Attributes addressed in Systems-of-Systems: A Systematic Mapping Study

Diego Gatica

Toeska research team

Universidad Técnica

Federico Santa María

Valparaíso, Chile

diego.gatica@sansano.usm.cl

Francisco Ponce

Toeska research team

Universidad Técnica

Federico Santa María

Valparaíso, Chile

francisco.ponceme@sansano.usm.cl

René Noël

Escuela de Ingeniería

Civil Informática

Universidad de Valparaíso

Valparaíso, Chile

rene.noel@uv.cl

Hernán Astudillo

Toeska research team

Universidad Técnica

Federico Santa María

Valparaíso, Chile

hernan@inf.utfsm.cl

Abstract—System-of-Systems (SoS) are composed by heterogeneous constituent systems that are autonomous, independently managed and operated, and geographically distributed. Architectural evaluation is a key activity of all systems design, construction and operation; unfortunately, the interrelationships among SoS constituent systems and quality attributes (QA) make hard to evaluate SoS. Several evaluation techniques have been proposed, but this corpus of knowledge has not yet been organized for easy access. This article describes the design, execution and results of a Systematic Literature Mapping (SLM) of architectural evaluation proposals, focusing on the QA's they address and the techniques they use. The SLM yield 1675 articles, of which 22 were selected for detailed analysis; most of them take a quantitative perspective (82%) and the rest are hybrid quantitative/qualitative (18%); most proposals offer no empirical validation at all (64%); and the most addressed QA's are performance and robustness.

Index Terms—System-of-System, SoS, Systematic Literature Mapping, Quality Attributes, Architecture Evaluation

I. INTRODUCTION

In the current era of information technologies (IT), systems are rarely built to operate alone. Examples of this type system can be found in specialized technological domains, like Smart Cities, Internet of Things, or Intelligent Transport Systems [1]. These systems are built to operate through the interaction of constituent systems to meet complex stakeholder needs, leading to the well-named System-of-systems (SoS). SoS are systems formed by heterogeneous constituent systems, autonomous, independently managed and operated, and geographically distributed [2]. Much of the success of SoS involves addressing a set of extra-functional properties (quality attributes, QA's) through an appropriate architectural design which can be justified by a reasoned architectural evaluation. The main goal of this activity is to identify whether the design decisions taken totally at any stage of system development address correctly the specified QA's.

Architectural evaluations in SoS context is a complex proposition [3], since architecture design in those big systems is not only influenced by QA's, but also by interaction with legacy systems, embedded systems, sensors, critical systems, and other characteristics inherent to SoS. Several techniques to evaluate SoS architecture have been proposed in the scientific literature; most of them differ both in their goals and used methods, which calls for a rigorous overview of the area in order to organize the growing corpus of knowledge. The goal of this article is to identify and characterize extant proposals for architectural evaluation of SoS according to their goals, QA's, and methods used. SoS Previous work to address this goal focuses on in making a holistic description of the SoS areas (specially [4] [5]) or are reviews related not conducted as rigorous systematic reviews [3].

In order to achieve our goal, we conducted a Systematic Literature Mapping (SLM) following the guidelines proposed by Petersen et al. [6]. Results show that most proposals focus on quantitative evaluations, offer no empirical validation, and are agnostic about the inherent SoS characteristic that can affect systems quality. Also, most proposals focus on a few QAs, like performance and robustness, which win by far to other QA's like evolution, interoperability, and evolvability.

The remainder of the article is structured as follows: Section II defines SoS and architectural evaluations; Section III describes related work; Section IV presents the article filtering criteria; Section V shows the SLM results and answers the research questions; Section VI synthesizes results; Section VII addresses threats to validity; and finally Section VIII summarizes and concludes.

II. BACKGROUND

The two key concepts for this article are systems-of-systems and architectural evaluation.

A. Systems-of-Systems

A System-of-Systems (SoS) is a system understood as a set of constituent systems that interact with each other to fulfill a common goal that can not be carried out by any of them independently [2]. Since this initial definition could also include a CPU with an LCD monitor, Nielsen et al. [7] reviewed several definitions in the literature and converged in a few characteristics:

- **Independence:** constituent systems can operate when detached from the rest of the SoS.
- **Autonomy:** a constituent system's behavior is governed by its own rules rather than by others external to the constituent.
- **Geographical distribution:** constituent systems are dispersed geographically.
- **Evolutionary development:** a SoS is in constant change, and functions may be constantly added or removed.
- **Emergent behavior:** the SoS behavior arises as a result of the synergistic collaboration of its constituents.
- **Interdependency:** mutual dependency that arises from constituent systems having to rely on each other to fulfill their common goal.
- **Evolution:** Many SoS are long-lasting and subject to change, in functionality delivered or its quality.
- **Emergence of behavior:** an SoS can to undertake structural changes without planned intervention.

B. Architectural evaluations

The architecture of a software system is the result of early design decisions, which are materialized through software elements and relationships that expose externally visible properties. The process of designing an architecture allows reasoning about fulfillment of QA's (also known as *ilities* or non-functional requirements), which are fundamental for the success of a software project [8].

Architectural evaluation is the activity concerned with evaluating the quality of a software system by determining whether its design addresses the specified QA's and sub-attributes [9]. Architecture evaluations can occur at an early or late stage: early evaluations require that some decisions has already been taken but there is no built system yet (e.g. to assess ongoing project progress), and late evaluations require that the system be already implemented (e.g. to make changes).

Architectural evaluations can be grouped into three categories [10] [11]: questioning, measuring, and hybrid. Questioning techniques (like SAAM) evaluate a design architecture in any phase of the process, by generating questions about how it addresses some QA's using in scenarios or checklist. Measuring techniques (like ArchOptions) provide answers to questions that an evaluation team has about particular qualities of the architecture, and require that architectural design artifacts be finished to apply simulations. Hybrid techniques (like ATAM) mix both approaches.

III. RELATED WORK

Some previous work has already addressed aspects related to the goal of this SLM.

Klein et al. [4] performed a systematic literature review on SoS architecture research. The authors conclude that many of the proposals are domain-oriented, and without broad adoption, leading to a low maturation of area. They did not focus on architectural evaluations.

Santos et al. [3] conducted a review of architectural evaluations in SoS. The study yield 16 relevant articles categorized into three evaluation type: mathematical-modeling (6 of 16), simulation-based (4 of 16), and scenario-based (6 of 16) [12]. The QA's most addressed were found to be performance, reliability, operability, complexity, and flexibility. Unfortunately, this study didn't follow a rigorous SLM approach.

Bianchi et al. [5] performed a SLR with to determine the QA's most important in the SoS, and analyzed the coverage of the ISO/IEC 25010 quality model. The study yield 56 QA's, among which the most important were security, interoperability, performance, reliability, and safety. Unfortunately, this study did not concern itself with evaluation approaches used by the techniques.

IV. METHOD DESIGN AND EXECUTION

To perform this SLM, we followed the guidelines proposed by Petersen et al. [13]. The following subsections describe in detail the study design and its execution.

A. Research Questions

The main goal of this study is *summarize and characterize the existing proposals in the academic literature of architectural evaluations of SoS, according to the types of evaluations and QA's mostly addressed by them*. To achieve this goal, we defined the following research questions:

- **MRQ:** Which techniques have been proposed to evaluate System-of-Systems architectures?
- **RQ1:** Which kinds of architecture evaluation techniques (questioning, quantitative, and/or hybrid) are used for SoS?
- **RQ2:** Which quality attributes are addressed by the techniques?
- **RQ3:** Which kinds of validation are used by the studies reporting techniques?

B. Search

Following the P.I.C.O. structure (**P**opulation, **I**ntervention, **C**omparison, and **O**utcome) [14], We articulated the search string with keywords derived from the research questions. However, since the study is a systematic mapping, we followed Petersen's suggestion [13] [6] and used only Population and Intervention.

Once we got our initial set of keywords, we performed searches for primary studies using digital search engines recommended for software engineering research [15] [16]. To validate the search string, we intersected the initial set of papers (recovered with the query applied to the databases)

TABLE I
SEARCH STRING

Item	Keywords
Population: Software architecture in System of System	"system of system" OR "system-of-system" OR "SoS" OR "architecture system of system" OR "architecture system-of-system"
Intervention: Method to evaluate quality aspects in System-of-System architecture	"evaluation" or "evaluating" or "simulation" or "simulating" or "assurance" or "quality"
Serch string	P + I

with the final set of papers reported by Santos et al. [3], considering that both survey have similar goals. Since our search string recovered 11 of the 16 articles identified by Santos et al., we can say that our search string has sufficient recall and representativity to recover relevant studies related to architectural evaluations of SoS.

Since each database engine had different limits on the length and syntax of the search string, We followed the strategy reported by Chet et al. [17], and customized the search string for each database to avoid losing search sensitivity while maintaining similarity among them. The final keyword set for each database can be perused at <https://doi.org/10.5281/zenodo.1336282>.

C. Inclusion/Exclusion criteria

To perform the LSM, the researchers applied the inclusion and exclusion criteria defined in the Table II in three phases.

In the first phase, the set of articles was divided into two equal parts and each of them was assigned to a researcher, who read the titles, keywords and abstract, and removed duplicate articles.

In the second phase, the resulting set was divided into three equal parts that were randomly assigned among the three designated researchers, who again read the title, keywords, and abstract, but now also read the introduction and conclusion.

Finally, in the third phase, the resulting set was divided into three equal parts that were randomly assigned among the three designated researchers, who read the full article.

If an article did not pass at least one criterion (in any of the phases), it was rejected. If a researcher had doubts about whether an article passed a criterion or not, the decision was taken in a face-to-face discussion. The number of articles at each stage is shown in Figure 2.

D. Quality Assessment

Regarding the the last Inclusion criterion (see Table II), we focused on articles with validations. However, there were three articles in particular [18]–[20] that after a face-to-face discussion, the researchers decided to leave in the set because their proposals were particularly interesting from SoS perspective. The articles distribution can be seen in Figure 1.

TABLE II
INCLUSION/EXCLUSION CRITERIA

Inclusion criteria	Exclusion criteria
Only articles in English	Articles that are in other language
Articles that refer to SoS	Proposal that refer to other type systems (e.g., Monolithic Systems, services-based systems)
Only primary studies	Secondary and Tertiary studies not are considered
Articles that present a proposal or initiative for evaluate SoS architecture	Proposals that focus on other activities (e.g., testing, design, etc)
Evaluation proposals that focus on QAs	Proposal that focus on other aspects (e.g, failure cascading, risk, etc)
Only articles that validate the proposal through a case study, experiment, or illustration.	Proposal without validation.

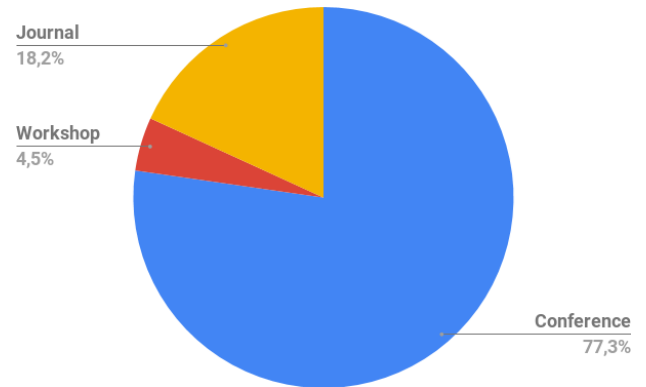


Fig. 1. Publication type

E. Data Extraction

Data extraction was done by three researchers, which used a worksheet to tabulate and organize data. The data values extracted from the papers and their relation with the research questions can be perused in <https://doi.org/10.5281/zenodo.1336282>.

V. RESULTS OF THE MAPPING

We run the SLM from 30-aug-2017 until 20-nov-2017, executing the phases mentioned in the subsection IV-C. Table III summarizes the result obtained applying the query in each database.

Figure 2 summarizes the results of each phase. The queries in all databases yield a total of 1675 articles, which were divided into 2 equitable subsets and reviewed independently by two researcher, who applied the first filter and obtained a total of 166 articles.

In the second phase, the set of 166 articles was divided into 3 equitable subsets and assigned to the three designated researchers, who read the title, abstract, introduction and conclusion, obtaining a total of 54 articles.

TABLE III
ARTICLES BY DATABASES ENGINE

Database	Number articles
ACM	364
IEEE	337
Scopus	794
Web of Science	96
Science Direct	84
Total	1675

In the third phase, the set of 54 articles was again divided into 3 equal subsets and assigned to the three designated investigators, who read the full article and reapplied the criteria, obtaining 22 relevant articles¹ distributed between 2005 and 2017. Finally, two researchers extracted the necessary data to answer the research questions and stored them in the worksheet in <https://doi.org/10.5281/zenodo.1336828>.

1) *RQ1: Which kinds of architecture evaluation techniques are used for SoS?:* Most proposals are quantitative (18/22 \approx 82%); 4/22 (\approx 18%) techniques are hybrid (see Table IV); and no purely qualitative techniques were found. The same table shows the techniques that the proposals use to perform the evaluations; 18/22 were based on quantitative techniques, mainly using a mixture of fuzzy logic with meta-heuristics (7/18) [21] [24] [22] [29] [35] [39] [23], and simulations models (6/18) [22] [20] [27] [28] [30] [31].

Generally method that use meta-heuristics with fuzzy logic focused on identifying a set of constituent systems that meet the desired quality levels, whilst model-driven evaluations (specially simulations) show how architecture design behaves under certain conditions.

Finally, most quantitative techniques focus on QA's, like Performance, that are obviously easy to measure by purely quantitative metrics [40] (see Figure 4). Hybrid techniques focus on well-known and accepted methods: ATAM & Mission Treat workshops [18], QFD matrix generation [26], and DoDAF [38].

2) *RQ2: Which quality attributes are addressed by the techniques?:* We considered the QA's used for proposals validation, as well as those that could be addressed outside the validation but were mentioned by the authors. Consequently the mostly addressed QA's are: Performance (11/22 \approx 50%), flexibility (6/22 \approx 27%), robustness (6/22 \approx 27%), and reliability (4/22 \approx 18%) (see Figure 4). Notice that operational QA's predominate over other types related with post-deployment, like maintainability, evolvability, sustainability and others. This is really interesting because in the SoS context the constituent systems must be reconfigured to collaborate and address a common goal; however, QA's related to those properties are addressed to a lesser extent. An example of systems (that can be considered as constituent systems) that claim for these QA's are legacy and SCADA systems, which

¹ [18]–[39]

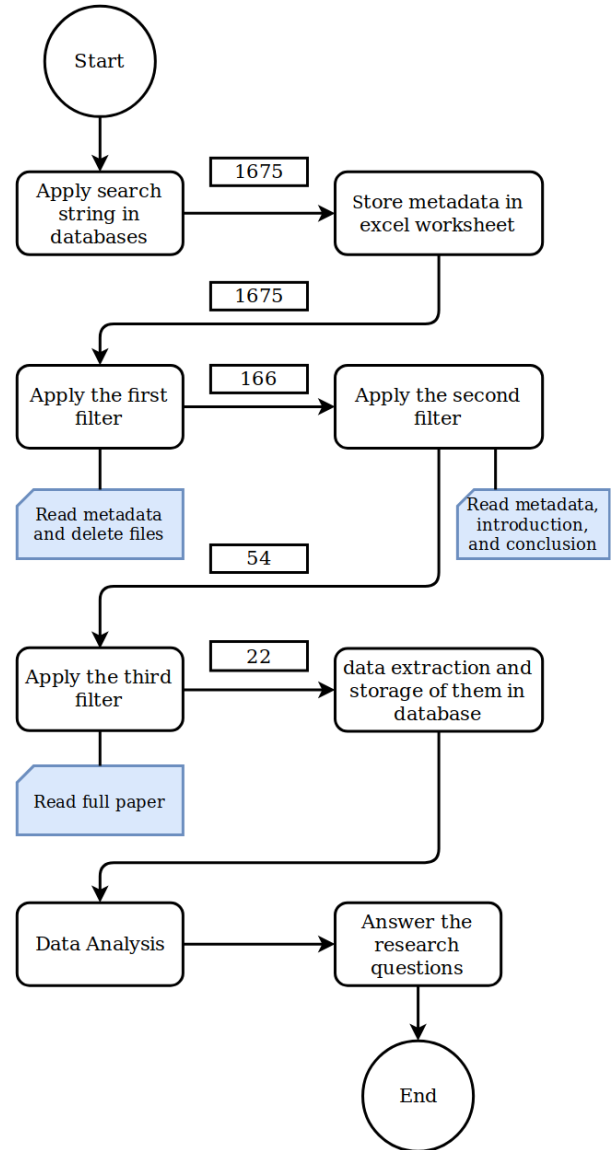


Fig. 2. Process for Systematic Mapping Study

within the current context of SoS are considered interesting [41] [42].

3) *RQ3: Which kinds of validation are used by the studies reporting techniques?:* Most proposals offer no empirical validation: 10/22 (\approx 45%) were illustrated with an example, whilst just 8/22 (\approx 36%) were validated with a case study (see Figure 3). It is curious to note that quantitative proposals preferred validation via illustrations, although the proposal [23] validated through experiment was also in this category (see Figure 4). Finally, is interesting remark that the proposal that used an experiment as a validation method focused only on one QA.

4) *MRQ: Which techniques have been proposed to evaluate SoS architectures?:* Most architectural evaluation proposals come from the systems engineering community² (21/22);

²See <https://doi.org/10.5281/zenodo.1336828>

TABLE IV
ARCHITECTURAL EVALUATION TYPES

Approaches	Meta-heuristic/ fuzzy logic	Model	FDA	Bayesian Net- works/event tree	Color nets	petri	ATAM/MTW	QFD matrix	DoDAF	None
Quantitative	7	6	2	2	1	0	0	0	0	0
Questioning	0	0	0	0	0	0	0	0	0	0
Hybrid	0	0	0	0	0	1	1	1	1	1

within them, 18/22 are quantitative, mostly using techniques related to meta-heuristics and fuzzy logic. These types of techniques have mainly been developed to evaluate the interaction of sets of systems according to the needs and expectations of stakeholders. In the same line, but to a lesser degree, were techniques that use formal models to corroborate the behavior of the design of the architecture. On the other hand, four hybrid techniques were found, of which only one comes from the software engineering community: a mixture of two well-known techniques, ATAM and MTW, but for which no validation was found. Respect to QA's, most approaches are operational, like performance, reliability, etc. Flexibility, despite being a non-operational attribute, was highly considered according to the evolutionary nature of SoS; however, proposals that address it from a qualitative perspective were not found.

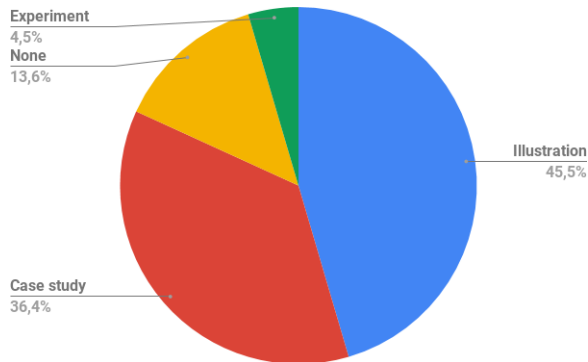


Fig. 3. Validation Types. *None* means that it has not illustrated the proposal

VI. SYNTHESIS AND DISCUSSION

We aimed to identify existing proposals to evaluate architectures and to determine which QA's were most addressed. To this end, we designed and executed an SLM which yield articles since 2005 to 2017, with an increasing number of publications per year (see Figure 5). However, considering the related works mentioned in Section III, we consider that there is a gap between the most QA's addressed by the proposals found in this SLM, and the key QA's identified recently.

Bianchi et al. [5] determined that the QA's most addressed in several SoS application domains (Military, IT systems, Smart Grid, Automotive, Transport) are Security, Performance, Interoperability, and Reliability. However, our SLM found that only Performance and Reliability are among the QA's most addressed by architectural evaluations (with 11 and 6 proposals, respectively).

Even more so, Nielsen et al. [7] performed a survey that synthesized the most important dimensions of SoS found in the literature, and then mapped each of them to specific studies (see Table 1 from Nielsen paper). Results showed that emergent Behavior (22), Evolution (21), Interoperability (20), and Distribution (19) were the characteristic most mentioned; however, our SLM found that both Evolution and Interoperability were the least addressed by evaluation proposals (1/22 each), while no proposal addressed the emergent behavior and geographic distribution. This glaring mismatch suggests opportunities to develop evaluation techniques that address these QA's in the SoS context.

The most used Software Engineering evaluation proposals use methods based in scenarios (such as ATAM) rather than formal methods (quantitative); thus, it is quite a contrast to notice that the preferred techniques to evaluate SoS are Genetic Algorithm, Fuzzy Logic and other similar approaches. This may be explained because SoS are usually associated to critical systems in fields like defense, health care and emergency, which have an interest in quantitative techniques. Specialized technological domains such as Smart Cities are of current interest and share similar characteristics with SoS, which opens up important opportunities to study techniques to identify trade-offs more precisely than with classic scenario-based methods, and to be able to articulate more quantitative assessments.

VII. THREATS TO VALIDITY

Throughout the execution of the protocol we identified and mitigated two validity threats.

- 1) **Selection bias:** We tried to mitigate the bias through the strict protocol defined in Section IV. In each phase (except the first), the articles were randomly redistributed among the researchers, so in the next stage each read articles selected by another one; if a reader was not sure whether an article passed all criteria, it was discussed by the entire research team applying a voting systems.
- 2) **Missing literature:** We executed the search string in the selected databases and counted the number of articles in the intersection between the initial set recovered by our query and the final set obtained by [3], which had a similar goal to this SLM; we found that their final set of articles is a subset of our initial set. Finally, since several database engines have different query syntax and rules, we personalized the initial set of keywords for each database, following Chen's example [17].

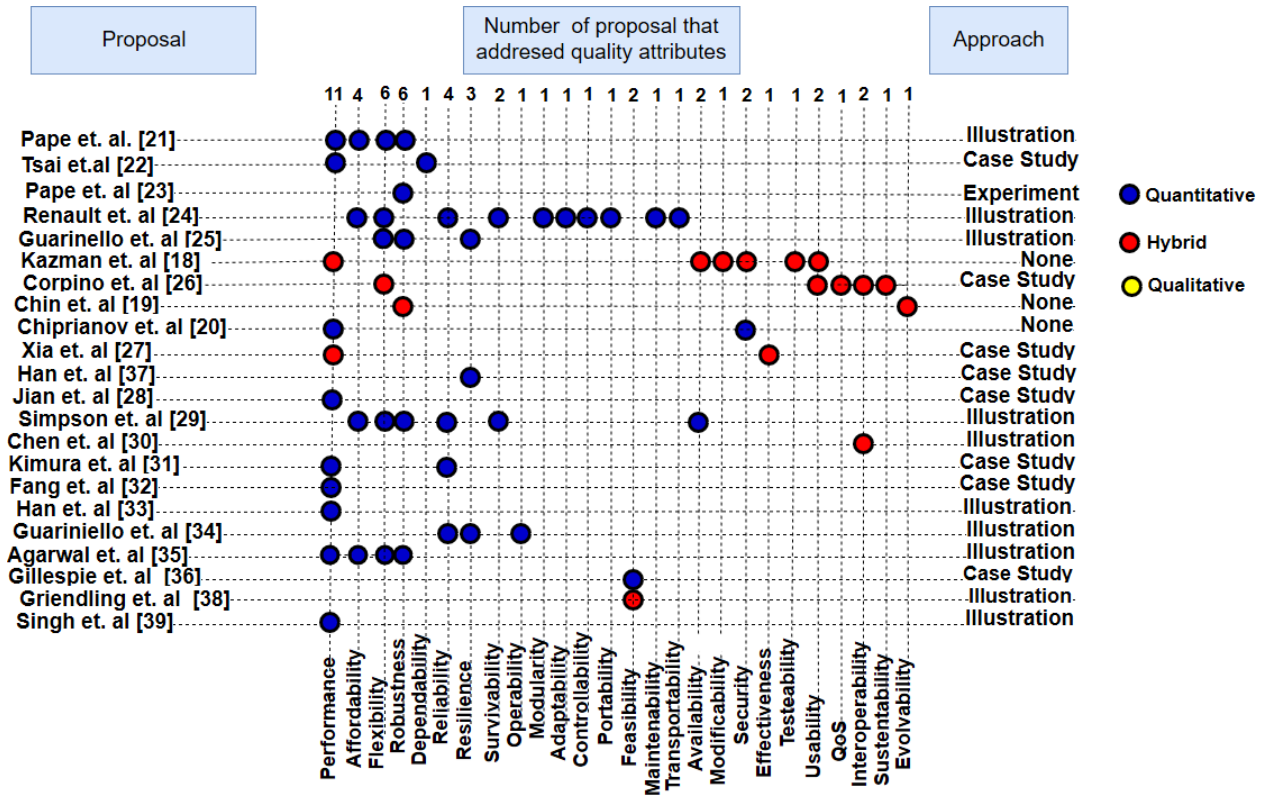


Fig. 4. Mapping between proposal and quality attributes addressed

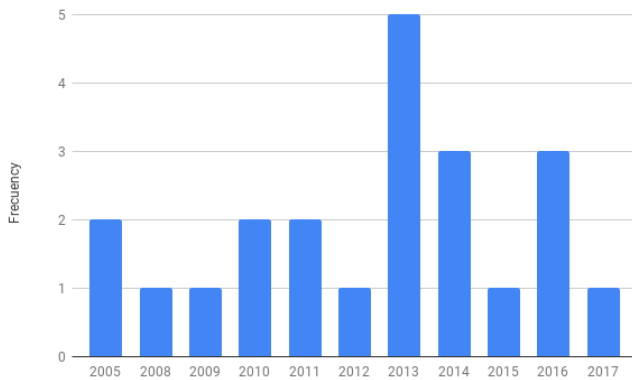


Fig. 5. Publications per year

VIII. CONCLUSIONS AND FUTURE WORK

SoS are complex systems with constituent systems which interact to achieve a common goal, characterized by independence, autonomy, distribution, evolutionary development, emergent behavior, etc. [7]. The concept has been used to describe systems within technological domains of interest to the software engineering community, such as Industry 4.0 and Smart Cities.

Much of the success of a SoS can be linked to the fulfillment of quality attributes or extra-functional properties, which can be evaluated by architectural evaluations. To understand

how evaluations are carried out, we conducted a Systematic Literature Mapping to identify and characterize proposals to evaluate System-of-Systems. The study yield 22 unique articles, of which 21 were from System Engineering community and just one from Software Engineering. Most proposals are quantitative (18/22) and focus on operational properties (e.g. performance, reliability, etc), and many of them use sophisticated techniques like fuzzy logic, meta-heuristics, and model simulation. Other properties like maintainability, evolution and sustainability were addressed to a lesser degree (one proposal each).

Considering the finding of Nielsen et al. [7] who attempt to find the most relevant characteristics in the context of SoS, there is a clear gap between QA's most addressed by the proposals found by this SLM and the properties/characteristics that these two articles have determined as relevant. Architectural evaluations that consider the impact of these characteristics in QA's were found in few proposal (just proposal that addressed interoperability, evolvability, sustainability, and others).

These results may help practitioners to identify available evaluation techniques for SoS, allow researchers to identify research opportunities, and help all readers to understand the current status of architectural evaluations of SoS.

ACKNOWLEDGMENTS

This work has been supported by UTFSM-DGIIP through Puente Grant and by CONICYT (Basal FB0821 CCTVal).

REFERENCES

- [1] E. Cavalcante, N. Cacho, F. Lopes, and T. Batista, "Challenges to the development of smart city systems: A system-of-systems view," in *Proceedings of the 31st Brazilian Symposium on Software Engineering*. ACM, 2017, pp. 244–249.
- [2] M. W. Maier, "Architecting principles for systems-of-systems," in *IN-COSE International Symposium*, vol. 6, no. 1. Wiley Online Library, 1996, pp. 565–573.
- [3] D. S. Santos, B. Oliveira, M. Guessi, F. Oquendo, M. Delamaro, and E. Y. Nakagawa, "Towards the evaluation of system-of-systems software architectures," in *VIII Workshop em Desenvolvimento Distribuido de Software, Ecossistemas de Software e Sistema de Sistemas (WDES)*, 2014, pp. 53–57.
- [4] J. Klein and H. Van Vliet, "A systematic review of system-of-systems architecture research," in *Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures*. ACM, 2013, pp. 13–22.
- [5] T. Bianchi, D. S. Santos, and K. R. Felizardo, "Quality attributes of systems-of-systems: A systematic literature review," in *Software Engineering for Systems-of-Systems (SESoS), 2015 IEEE/ACM 3rd International Workshop on*. IEEE, 2015, pp. 23–30.
- [6] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
- [7] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Pelseska, "Systems of systems engineering: basic concepts, model-based techniques, and research directions," *ACM Computing Surveys (CSUR)*, vol. 48, no. 2, p. 18, 2015.
- [8] P. Clements, R. Kazman, and M. Klein, *Evaluating software architectures*. Tsinghua University Press Beijing, 2003.
- [9] J. Knodel and M. Naab, *Pragmatic Evaluation of Software Architectures*. Springer, 2016.
- [10] L. Bass, P. Clements, R. Kazman, L. Northrop, and A. Zaremski, "Recommended best industrial practice for software architecture evaluation," 1997.
- [11] R. F. Barcelos and G. H. Travassos, "Evaluation approaches for software architectural documents: a systematic review," in *CIBSE*, 2006, pp. 433–446.
- [12] J. Bosch, *Design and use of software architectures: adopting and evolving a product-line approach*. Pearson Education, 2000.
- [13] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *EASE*, vol. 8, 2008, pp. 68–77.
- [14] S. C. B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn, 2007.
- [15] T. Dyba, T. Dingsoyr, and G. K. Hanssen, "Applying systematic reviews to diverse study types: An experience report," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. IEEE, 2007, pp. 225–234.
- [16] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
- [17] L. Chen and M. A. Babar, "A systematic review of evaluation of variability management approaches in software product lines," *Information and Software Technology*, vol. 53, no. 4, pp. 344–362, 2011.
- [18] R. Kazman, M. Gagliardi, and W. Wood, "Scaling up software architecture analysis," *Journal of Systems and Software*, vol. 85, no. 7, pp. 1511–1519, 2012.
- [19] K. S. Chin, P. E. Yau, S. K. Wah, and P. C. Kiang, "Framework for managing system-of-systemsilities," *DSTA HORIZONS*, vol. 14, 2013.
- [20] V. Chiprianov, K. Falkner, L. Gallon, and M. Munier, "Towards modelling and analysing non-functional properties of systems of systems," in *System of Systems Engineering (SOSE), 2014 9th International Conference on*. IEEE, 2014, pp. 289–294.
- [21] L. Pape, K. Giammarco, J. Colombi, C. Dagli, N. Kilicay-Ergin, and G. Rebovich, "A fuzzy evaluation method for system of systems meta-architectures," *Procedia Computer Science*, vol. 16, pp. 245–254, 2013.
- [22] W.-T. Tsai, C. Fan, Y. Chen, and R. Paul, "A service-oriented modeling and simulation framework for rapid development of distributed applications," *Simulation Modelling Practice and Theory*, vol. 14, no. 6, pp. 725–739, 2006.
- [23] L. Pape and C. Dagli, "Assessing robustness in systems of systems meta-architectures," *Procedia Computer Science*, vol. 20, pp. 262–269, 2013.
- [24] A. Renault and C. Dagli, "Genetic algorithm optimization of sos meta-architecture attributes for fuzzy rule based assessments," *Procedia Computer Science*, vol. 95, pp. 95–102, 2016.
- [25] C. Guariniello and D. DeLaurentis, "Integrated analysis of functional and developmental interdependencies to quantify and trade-offilities for system-of-systems design, architecture, and evolution," *Procedia Computer Science*, vol. 28, pp. 728–735, 2014.
- [26] S. Corpino and F. Nichele, "Anilities-driven methodology for the analysis of gaps of stakeholders needs in space systems conceptual design," *IEEE Systems Journal*, 2016.
- [27] X. Xia, J. Wu, C. Liu, and L. Xu, "A model-driven approach for evaluating system of systems," in *Engineering of Complex Computer Systems (ICECCS), 2013 18th International Conference on*. IEEE, 2013, pp. 56–64.
- [28] X. Jian, G. Bing-feng, Z. Xiao-ke, Y. Ke-wei, and C. Ying-wu, "Evaluation method of system-of-systems architecture using knowledge-based executable model," in *Management Science and Engineering (ICMSE), 2010 International Conference on*. IEEE, 2010, pp. 141–147.
- [29] J. J. Simpson and C. H. Dagli, "System of systems architecture generation and evaluation using evolutionary algorithms," in *Systems Conference, 2008 2nd Annual IEEE*. IEEE, 2008, pp. 1–6.
- [30] P. Chen, "Architecture-based interoperability evaluation in evolutions of networked enterprises," in *International Conference on Business Process Management*. Springer, 2005, pp. 293–304.
- [31] D. Kimura, T. Osaki, K. Yanoo, S. Izukura, H. Sakaki, and A. Kobayashi, "Evaluation of it systems considering characteristics as system of systems," in *System of Systems Engineering (SoSE), 2011 6th International Conference on*. IEEE, 2011, pp. 43–48.
- [32] Z. Fang, D. DeLaurentis, and N. Davendralingam, "An approach to facilitate decision making on architecture evolution strategies," *Procedia computer science*, vol. 16, pp. 275–282, 2013.
- [33] S. Y. Han and D. DeLaurentis, "Development interdependency modeling for system-of-systems (sos) using bayesian networks: Sos management strategy planning," *Procedia Computer Science*, vol. 16, pp. 698–707, 2013.
- [34] C. Guariniello and D. DeLaurentis, "Dependency analysis of system-of-systems operational and development networks," *Procedia Computer Science*, vol. 16, pp. 265–274, 2013.
- [35] S. Agarwal, L. E. Pape, C. H. Dagli, N. K. Ergin, D. Enke, A. Gosavi, R. Qin, D. Konur, R. Wang, and R. D. Gottapu, "Flexible and intelligent learning architectures for sos (fila-sos): Architectural evolution in systems-of-systems," *Procedia Computer Science*, vol. 44, pp. 76–85, 2015.
- [36] S. E. Gillespie, R. E. Giachetti, A. Hernandez, P. T. Beery, and E. P. Paulo, "System of systems architecture feasibility analysis to support tradespace exploration," in *System of Systems Engineering Conference (SoSE), 2017 12th*. IEEE, 2017, pp. 1–6.
- [37] S. Y. Han, K. Marais, and D. DeLaurentis, "Evaluating system of systems resilience using interdependency analysis," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1251–1256.
- [38] K. Griendling and D. Mavris, "An architecture-based approach to identifying system-of-systems alternatives," in *System of Systems Engineering (SoSE), 2010 5th International Conference on*. IEEE, 2010, pp. 1–6.
- [39] A. Singh and C. H. Dagli, "Multi-objective stochastic heuristic methodology for tradespace exploration of a network centric system of systems," in *Systems Conference, 2009 3rd Annual IEEE*. IEEE, 2009, pp. 218–223.
- [40] P. Bengtsson and J. Bosch, "Architecture level prediction of software maintenance," in *Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on*. IEEE, 1999, pp. 139–147.
- [41] M. A. Ramos, P. C. Masiero, R. T. Braga, and R. A. Penteado, "Reengineering legacy systems towards system of systems development," in *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on*. IEEE, 2012, pp. 624–630.
- [42] S. Karnouskos and A. W. Colombo, "Architecting the next generation of service-based scada/dcs system of systems," in *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2011, pp. 359–364.