UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

AN INVESTIGATION ON QUALITY ATTRIBUTES OF SYSTEMS-OF-SYSTEMS

DANIEL SOARES SANTOS
THIAGO BIANCHI
KATIA ROMERO FELIZARDO
ELISA YUMI NAKAGAWA

Nº 404

RELATÓRIOS TÉCNICOS

São Carlos – SP
Fev./2015

An Investigation on quality attributes of Systems-of-Systems

Daniel Soares Santos
Thiago Bianchi
Katia Romero Felizardo
Elisa Yumi Nakagawa

Abstract

System-of-Systems (SoS) have been considered the new trend of software systems for several reasons, such as the advancement of computer technology and the increasing complexity and size of the systems. The concept of SoS has the central idea of a set of operationally independent software systems that collaborate together to compose a larger and more complex system. For SoS, several quality attributes are critical due to its characteristics and domain constraints. However, the existent quality models were not conceived to address SoS. Therefore, the identification of quality models that can address the quality attributes for SoS needs to be investigated. So, the purpose of this work is to identify the most common quality attributes in SoS context and analyze their coverage by the well established ISO/IEC 25010 quality model. As a result, we have identified that the current quality models cannot adequately address the SoS quality attributes due its dynamic nature. Moreover, a first analysis about the ISO/IEC 25010 coverage shows that 48% of the quality attributes commonly considered in SoS are not addressed.

Contents

Abstract **i**

1 Introduction **1**

2 Background **3**
2.1 Systematic Literature Review 3
2.2 System-of-Systems 4
2.3 Quality Model 5
2.3.1 ISO/IEC 25010 6

3 Systematic Literature Review Application **8**
3.1 Phase 1 - Planning 8
3.2 Phase 2 - Conduction 11
3.3 Phase 3 - Analysis 17
3.4 Threats to Validity 20
3.5 Discussion 22

4 Conclusion and Future Work **26**

References **33**

List of Figures

2.1 ISO/IEC 25010 – Quality in Use Model. 6
2.2 ISO/IEC 25010 – Product Quality Model. 7

3.1 Studies included per year. 17
3.2 Authorship relationship between the studies. 18
3.3 Maturity of the included studies. 18
3.4 Quality Attributes identified in the included studies. 19
3.5 Application Domains identified on the included studies. 19
3.6 Quality attributes identified on the Information Technology domain. 20
3.7 Quality attributes identified on the Military domain. 20
3.8 Quality attributes identified on the Smart Grid domain. 21

List of Tables

3.1 Selected publication databases 9
3.2 Research sources and primary studies selected and included. 12
3.3 Included primary studies 12
3.4 ISO/IEC 25010 quality attributes coverage 23

Chapter 1

Introduction

Software-intensive systems have become increasingly large and complex and also are becoming important and even essential to the whole society. These systems are sometimes the outcome from the interoperability of constituent systems that work together in order to provide more complex missions that could not be completed by any of these systems separately (Maier, 1998; Nakagawa et al., 2013). This class of software systems has been referred as System-of-Systems (SoS) and can be identified in different domains, in particular, cyber-physical systems, such as medical systems, airport systems, robotic and automotive (Lane, 2013; Nakagawa et al., 2013).

For SoS, besides interoperability, several other quality attributes are critical (e.g., performance, reliability, and security) and, therefore, should be carefully considered to satisfy the mission (Ackermann et al., 2009), which defines a functional goal assigned to the SoS (Grondin et al., 2012). However, to achieve these goals is difficult, since the constituent systems are developed and maintained by different organizations (Gagliardi et al., 2009; Santos et al., 2014). In addition, the development of an SoS often happens in an unknown environment and surrounded by uncertainties. Thus, these systems must be prepared to support a dynamic, evolving development environment in which it is possible to incorporate new features and even reorganize the existent ones at run-time (Boardman and Sauser, 2006).

On the other hand, a well-accepted way to support quality control is to adopt software quality models that have been over the last three decades an area widely researched in

Sofware Engeneering (Wagner, 2013). A quality model intends to make the software quality better understandable and manageable. In this context, the ISO/IEC 25010 quality model has become the standard for evaluating the quality of modern computer-intensive systems and it is being also used in several context, such as to measure the quality of U.S. Departament of Defense (DoD) systems (Azizian et al., 2011). The ISO/IEC 25010 is based on the fact that the software product quality can be specified and evaluated using a hierarchical structure of quality attributes (ISO/IEC, 2010).

In this scenario, the quality attributes must be properly addressed by the quality models considering a variety of domains, since applications of SoS are migrating from traditional military domains to other domains such as smart home, integrated health systems and crisis management systems. However, there is not a clearly decomposition criteria that determines how the complex concept "quality" should be handled for SoS what leads to redundancies and overlaps considering the existent quality models (Wagner, 2013). In summary, the quality management of an SoS is a challenge that has not yet been overcome by classical software engineering (DoD, 2008).

The main contribution of this study is to present a panorama about the current state-of-the-art on quality attributes in the SoS context, considering different application domains. In addition, we analyzed the coverage of a well-established quality model, namely the ISO/IEC 25010, on the SoS quality attributes identified in this study. To build such panorama, we performed a Systematic Literature Review (SLR) (Kitchenham and Charters, 2007), an approach for finding and summarizing available evidences on a particular research topic.

The remainder of this technical report is organized as follows. In Chapter 2, a background about SLR, SoS, and quality models is showed. In Chapter 3 the conducted SLR and its results are detailed. In Chapter 3.5, discussion of the results, lessons learned, and limitations of this research are presented. Finally, conclusions are described in Chapter 4.

Chapter 2

Background

The term System-of-Systems (SoS) has been widely used in diverse domains such as, biology, ecology, and more recently, in software domain. In general, SoS can be considered as a set or arrangement of independent, useful systems integrated into larger systems that deliver unique capabilities (Department of Defense, 2008). In this context, quality models are being used with increasing frequency in order to identify relevant quality characteristics that can be further used to better establish requirements from the SoS application domain.

The next sections presents some important concepts related to Systematic Literature Review, SoS, and Quality Model areas in order to promote a better understanding about the topics covered in this technical report.

2.1 Systematic Literature Review

It is noticed that when a research area is mature, there is almost always an increase in the number of reports and results made available. During the study of a new knowledge area, researchers usually conduct a bibliographical review (almost always an informal review) to identify publications related to a specific subject. However, this kind of review do not use a systematic approach and do not offer any support to avoid bias during the selection of the publications that will be analyzed. Thus, it is important to have mechanisms to summarize and provide an overview about an area or topic of interest. In this context, SLR technique has investigated and proposed (Kitchenham and Charters, 2007). An individual evidence (for instance, a case study or an experimental study divulged in a publication/paper) that contributes to a SLR is called primary study. SLR aims at providing an overview of a research area to assess the quantity, quality, and type of primary studies existing on a topic of interest. Thus, the conduction of a SLR seems to be quite interesting in the context of this work.

2.2 System-of-Systems

The first definition of SoS appeared in studies published in the early 90s in domains such as, military, information technology and air traffic control. Although the term meet featured today, has no universally recognized definition (Luzeaux and Ruault, 2010). However, there is a convergence to some of the features identified by Maier (1998). These features have been widely shared among the international community and allow to understand the difference between SoS and complex monolithic systems: (i) the constituent systems are operationally independent and have their own goals, even when disconnected from the SoS; (ii) the managerial independence means that each constituent system is developed and maintained by different organizations, with their own stakeholders, development teams, processes, and resources; (iii) the SoS development is evolutionary and adaptive, where structures, functions, and purposes are added, removed, and modified according to the emerging needs of the system; (iv) the emergent behavior means that properties, functionality, and features of the SoS can not be identified or provided by any constituent system working separately; and (v) geographical distribution, where the constituent systems may be located in different places depending on the employed technologies and the communication means.

SoS started to gain their popularity mainly on military systems as a strategy to reach goals or deliver unique capabilities wherein a collaborative work of already existing and complex systems is needed (Department of Defense, 2010; Maier, 1998). Currently, considering the required interconnection among software systems, as well as the growing presence of software in systems engineering, SoS has also become focus of interest in Software Engineering (Boehm and Lane, 2006). Considering this, it is possible to affirm that evolution of computational systems points out that more and more software systems could be characterized as SoS. As previously mentioned, there is no universally accepted definition for SoS, and their characterization often depends on the viewpoint and its application. However, a practical definition is that a System-of-Systems is a “supersystem” composed by complex systems and operationally independent systems to achieve higher goal (Jamshidi, 2008). These characteristics can be observed in various application domains.

Nowadays, SoS is present in various application domains, such as, office (Carbon et al., 2008), telemedicine (Petcu and Petrescu, 2010), and e-business (Zhu et al., 2008). It is also important to mention that SoS has been also applied in critical domains (Aoyama and Tanabe, 2011; Farcas et al., 2010). For instance, Farcas et al. (2010) presents a complex software-intensive SoS for automotive and avionics. Aoyama and Tanabe (2011) also presents an SoS for automotive embedded systems and reports that it is a new perspective to a more effective architecture than conventional functional component-and-connector model.

It is observed that SoS can provide a suitable solution for a diversity of application domains where software-intensive systems are necessary and, as a consequence, SoS could certainly contribute to several sector of the society (Brøndum and Zhu, 2010). Therefore, attention to the development of SoS must be intensified, including new approaches to develop and maintain them.

2.3 Quality Model

The objective of a quality model is to describe, assess and/or predict quality, usually through hierarchical decomposition of the general product quality into sub-characteristics to make them better understandable and manageable (Wagner, 2013). The quality models can be used by developers, acquirers, quality assurance, control staff and independent evaluators, particularly those responsible for specifying and evaluating software product quality (ISO/IEC, 2010). In summary, the use of the quality models can benefit several software development activities such as (ISO/IEC, 2010):

- identifying software and system requirements;
- validating the comprehensiveness of a requirements definition;
- identifying software and system design objectives;
- identifying software and system testing objectives;
- identifying quality control criteria as part of quality assurance;
- identifying acceptance criteria for a software product and/or software-intensive computer system; and
- establishing measures of quality characteristics in support of these activities.

Published in the 70s, Boehm et al. (1978) and McCall et al. (1977) were the first quality models for software. The two models are similar and use a hierarchical decomposition of the concept quality into quality factors such as maintainability or reliability

(Wagner, 2013). The McCall's model was developed by the US Air-force Electronic System Decision (ESD), the Rome Air Development Center (RADC), and General Electric (Ravichandran and Rothenberger, 2003). The major contribution of the McCall's model is the relationship between the quality characteristics and metrics despite not consider directly the functionalities of software products. Boehm (Boehm et al., 1978) added new quality factors to McCall's model (AL-Badareen et al., 2011) but the author no had presented any suggestion to improve the measuring of quality. In 1987, Robert Grady and Hewlett-Packard Co. proposed the FURPS model (Grady and Caswell, 1987) that decomposes the software quality into functionality, usability, reliability, performance and supportability. However, this model had considered only the user's quality aspects and not took into account quality attributes such as portability and maintainability (AL-Badareen et al., 2011). Thereafter, several variations of these models have appeared over time in order to mitigate the shortcoming of the previous quality models. This motivated the world-wide standardization and consensus of the software quality model (AL-Badareen et al., 2011). In this context, ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) developed the ISO 9126 - quality model which is part of the ISO 9000 standard, which is the most important standard for quality assurance. In 2011, the ISO 9126 was technically revised and replaced by ISO/IEC 25010 - System and Software Quality Models, described in details as following.

2.3.1 ISO/IEC 25010

In this model, the software quality attributes were classified in a hierarchical structure of characteristics and sub characteristics divided in two quality models, quality in use model and product quality model, each one considering different quality aspects.

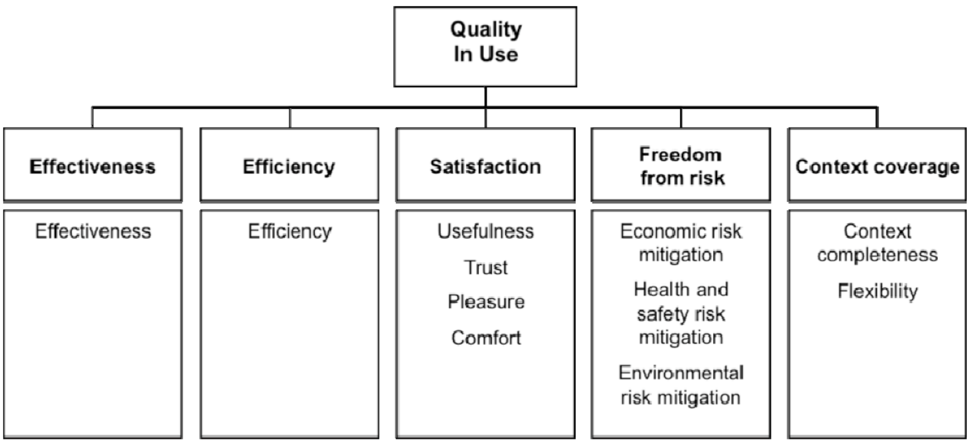

Figure 2.1: ISO/IEC 25010 – Quality in Use Model.

The quality in use model defines five characteristics related to outcomes of interaction with a system: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage as can be seen in Figure 2.1. The quality in use of a system characterizes the impact that the system or software product has on stakeholders. It can be determined by the quality of the software, hardware and operating environment, and the characteristics of the users, tasks and social environment (ISO/IEC, 2010).

The product quality model categorizes system/software/product quality properties into eight characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. Each characteristic is composed of a set of related sub-characteristics that are relevant and can be applied in both software and systems products as shown in Figure 2.2 (ISO/IEC, 2010).

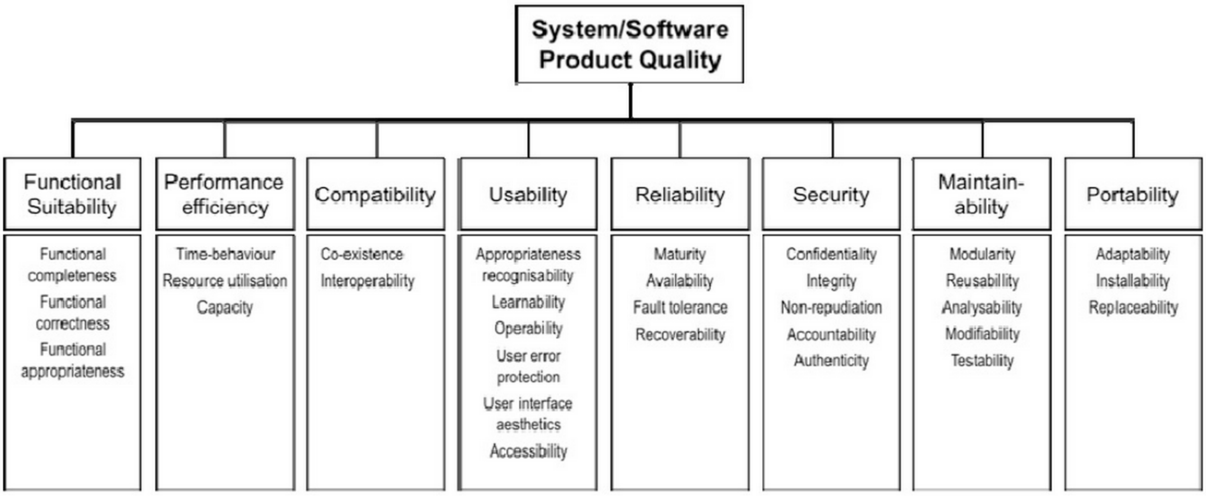

Figure 2.2: ISO/IEC 25010 – Product Quality Model.

Chapter 3

Systematic Literature Review Application

In order to conduct our SLR, we followed the process proposed by (Kitchenham and Charters, 2007). In short, the SLR process is composed by three main phases, planning, conduction and reporting. During the planning phase, the research objectives and the SLR protocol are defined. The protocol constitutes a pre-determined plan that describes research questions and how the SLR will be conducted. During the second phase, primary studies are identified, selected, and evaluated according to the inclusion and exclusion criteria previously established. For each selected study, data are extracted and synthesized. Finally, during the last phase, a final report is organized and presented. The next sections present these three phases in details.

3.1 Phase 1 - Planning

The objective of our SLR is to understand and summarize evidences on quality attributes for SoS considering its software domains. Therefore, the following research questions (RQ) were established: (i) **RQ1**: *Which are the most common quality attributes for SoS?*; (ii) **RQ2**: *Which are the most common application domains considered for SoS?*; and **RQ3**: *Which are the QA established for each SoS domains?*

In order to establish the search strategy, which aims finding possibly all primary studies to answer our RQs, there were defined the following five elements:

1. **Sources selection criteria**: Three types of sources were considered in our SLR: (a) publication databases: Aiming at establishing which publication database would be used to find the primary studies. It was adopted the following criteria (Dieste et al., 2009): *content update* (publications are regularly updated); *availability* (full text of the papers are available); *quality of results* (accuracy of the results obtained by the search); and *versatility export* (a mechanism to export the results of the search is available); (b) related work: it was also considered those studies cited as related works in the relevant primary studies found in the publication databases; and (c) specialist: it was also considered studies suggested by specialists of the areas of system of systems and software quality. Although the indication of studies by specialists can be considered as bias, it was adopted this source aiming to not lose any important evidence.

2. **Sources list:** The publication databases selected for this SLR are shown in Table 3.1. According to Dyba et al. (2005), these databases are efficient to conduct SLRs in the context of Software Engineering. Furthermore, *Scopus* has been added, since it is considered the largest database of scientific studies and its citations (Kitchenham and Charters, 2007)

Table 3.1: Selected publication databases

Source	Location
ACM Digital Library	`http://dl.acm.org/`
IEEE Xplore	`http://ieeexplore.ieee.org/Xplore/home.jsp`
Scopus	`http://www.scopus.com/home.url`
Web of Science	`http://apps.isiknowledge.com/`

3. **Studies language:** Only primary studies written in English were considered in our SLR. English was adopted because most of research in Computer Science have been reported in this language;

4. **Keywords:** The main keywords were "*System of Systems*" and "*Quality Attributes*", with the following related terms: (a) **System of Systems**: system-of-systems, systems of systems, and systems-of-systems; and (b) **Quality Attributes**: quality attribute, quality attributes, non functional requirement, non-functional requirement, nonfunctional requirement, non functional requirements, non-functional requirements, nonfunctional requirements, quality requirement, quality requirements, quality characteristic, quality characteristics, quality criteria, quality criterion OR non

functional property, non-functional property, nonfunctional property, non functional properties, non-functional properties, nonfunctional properties, non functional characteristic, non-functional characteristic, nonfunctional characteristic, non functional characteristics, non-functional characteristics, nonfunctional characteristics, quality models, and quality model.

5. **Search string:** Finally, the keywords were combined with the boolean operators OR and AND to link the main terms and their related synonyms. Thus, the final search string used was:

("system of systems" OR "system-of-systems" OR "systems of systems" OR "systems-of-systems") AND ("quality attribute" OR "quality attributes" OR "non functional requirement" OR "non-functional requirement" OR "nonfunctional requirement" OR "non functional requirements" OR "non-functional requirements" OR "nonfunctional requirements" OR "quality requirement" OR "quality requirements" OR "quality characteristic" OR "quality characteristics" OR "quality criteria" OR "quality criterion" OR "non functional property" OR "non-functional property" OR "nonfunctional property" OR "non functional properties" OR "non-functional properties" OR "nonfunctional properties" OR "non functional characteristic" OR "non-functional characteristic" OR "nonfunctional characteristic" OR "non functional characteristics" OR "non-functional characteristics" OR "nonfunctional characteristics" OR "quality models" OR "quality model")

Another important activity of the SLR planning is to define the Inclusion Criteria (IC) and Exclusion Criteria (EC). These criteria make possible to include primary studies that are relevant to answer the research questions and exclude studies that do not answer them. Thus, the inclusion criteria were:

1. **IC1**: The study addresses one quality attribute for SoS;

2. **IC2**: The study addresses more than one QA for SoS;

3. **IC3**: The study addresses QA for SoS in one application domain; and

4. **IC4**: The study addresses QA for SoS in more than one application domain.

The exclusion criteria were:

1. **EC1**: The study does not address SoS;

2. **EC2**: The study does not propose QA for SoS;

3. **EC3**: The study is an editorial, keynote, opinion, tutorial, poster or panel;
4. **EC4**: The study is a previous version of a more complete study about the same research;
5. **EC5**: The paper language is different from English;
6. **EC6**: The paper is duplicated; and
7. **EC6**: The full paper is not available.

In our SLR, the selection and evaluation of primary studies was performed in three steps:

1. **First selection:** Initially, the search string was customized and applied to each publication databases, previously listed in Table 3.1. During the first step, the title, abstract, and keywords of all primary studies available were considered. As a result, a set of primary studies possibly related to the research topic was obtained. Based on this set, the title and the abstract of each primary study were read and the inclusion and exclusion criteria were applied. Studies were selected as relevant or not. The introduction and the conclusion sections of each primary study were also considered when necessary. Finally, if a study was selected during the first step, it was read in full;
2. **Second selection:** During the second step, each primary study selected was read in full and analyzed again according to inclusion and exclusion criteria. In addition, the related work cited by these studies were evaluated intending to cover the whole research area. This additional selection might be a great source of evidence, since an included study often presents related works in the same research area. If the decision about the inclusion or exclusion of a study was not clear, the study was analyzed by two reviewers. When a disagreement occurred, discussions were conducted; and
3. **Data Extraction and Synthesis Method:** In order to extract data, it was planned to build data extraction tables related to each research question. These tables were used to synthesize results to facilitate the drawing of conclusions. To summarize and describe the set of data, statistical synthesis method and meta-analysis was applied.

3.2 Phase 2 - Conduction

Our SLR was conducted by three people, a software engineering researcher, a senior software developer/architect from the industry, and a SLR specialist. The SLR was started in October/2014 and it was finalized in December/2014. In this section, the details of the procedures to select the studies are described.

The primary studies were firstly identified on the selected databases, according to the SLR planning previously established. As a result, 116 studies were identified. Next, the primary studies were selected by reading of title and abstract and by the application of the inclusion and exclusion criteria. The Table 3.2 summarizes the number of studies obtained on each database, the number of included studies, and the precision rate[1]. ACM, IEEE Xplore, Scopus, and ISI Web of Science showed precision rates of 0.173, 0.181, 0.073, and 0.875, respectively. It is important to highlight that mainly Scopus stores studies from other databases, such as IEEE Xplore and ACM. After that, repeated studies were also excluded and a total of 50 studies were selected. Then, the inclusion and exclusion criteria were again applied after full reading of the selected studies. Finally, 40 studies were selected as the most relevant to this SLR. At the end, 12 studies were included by the reviewers that were not returned originally by the selected databases due to another SLR performed by the authors of this work (Santos et al., 2014) accounting a total of 52 studies included.

Table 3.2: Research sources and primary studies selected and included.

Source	Selected	Included	Precision
ACM Digital Library	5	5	1
IEEE Xplore	14	9	0.643
Scopus	89	42	0.472
Web of Science	8	7	0.875

Table 3.3 shows the 52 primary studies included. This table also presents the citation for each study and the inclusion criteria (IC) used to include it. Column "Document Type" indicates if the primary study was published as a Journal Article (JA) or as a Conference Paper (CP).

Table 3.3: Included primary studies

#	**Citation**	**IC**	**Type**
S1	Ackermann, C., Lindvall, M. and Cleaveland, R. Towards Behavioral Reflexion Models.(Ackermann et al., 2009)	IC2,IC4	CP

continued on next page ...

[1]Ratio between the total of included studies of a database and the total of primary studies obtained by this database

#	Citation	IC	Type
S2	Alghamdi, A.; Hussain, T. and Khan, G. Enhancing C4I security using threat modeling.(Alghamdi et al., 2010)	IC2,IC3	CP
S3	Allen, M. From substandard to successful software. (Allen, 2009)	IC2,IC3	JA
S4	Aoyama, M. and Tanabe, H. A Design Methodology for Real-Time Distributed Software Architecture Based on the Behavioral Properties and Its Application to Advanced Automotive Software. (Aoyama and Tanabe, 2011)	IC2,IC3	CP
S5	Azizian, N., Mazzuchi, T., Sarkani, S. and Rico, D. A framework for evaluating technology readiness, system quality, and program performance of U.S. DoD acquisitions. (Azizian et al., 2011)	IC2,IC3	JA
S6	Babar, M.A.a and Chauhan, M.A.b. A tale of migration to cloud computing for sharing experiences and observations. (Babar and Chauhan, 2011)	IC2,IC3	CP
S7	Balci, O. and Arthur, J. D. and Ormsby, W. F. Achieving reusability and composability with a simulation conceptual model. (Balci et al., 2011)	IC2,IC4	JA
S8	Batista, T. Challenges for sos architecture description. (Batista, 2013)	IC2,IC4	CP
S9	Belloir, Nicolas and Chiprianov, Vanea and Ahmad, Manzoor and Munier, Manuel and Gallon, Laurent and Bruel, Jean-Michel. Using Relax Operators into an MDE Security Requirement Elicitation Process for Systems of Systems. (Belloir et al., 2007)	IC2,IC4	CP
S10	Calinescu, R.a and Kikuchi, S.b and Johnson, K.a. Compositional reverification of probabilistic safety properties for large-scale complex IT systems. (Calinescu et al., 2012)	IC2,IC3	JA
S11	Hong-Mei Chen and Kazman, R. Architecting ultra-large-scale green information systems. (Chen and Kazman, 2012)	IC1,IC3	CP
S12	Chiprianov, Vanea and Falkner, Katrina and Gallon, Laurent and Munier, Manuel. Towards modelling and analysing non-functional properties of systems of systems. (Chiprianov et al., 2014)	IC2,IC4	CP
S13	Colin, C.V. and Duncan, J.R. and Lu, L. and Zongyang, L. and David, E.W. and Jie, X. A scenario-based architecture evaluation framework for network enabled capability. (Colin et al., 2009)	IC2,IC3	CP

continued on next page ...

#	**Citation**	**IC**	**Type**
S14	Eklund, U.a c and Bosch, J.b. Architecture for embedded open software ecosystems. (Eklund and Bosch, 2014)	IC2,IC3	JA
S15	Fang, Z., DeLaurentis, D. and Davendralingam, N. An Approach to Facilitate Decision Making on Architecture Evolution Strategies. (Fang et al., 2013)	IC2,IC3	JA
S16	Fischer, D., Sarkarati, M., Spada, M., Michelbach, T., Urban, W. and Tueffers, C. An application security framework for SOA-based mission data systems. (Fischer et al., 2011)	IC1,IC3	CP
S17	Fuchs, A. and Rieke, R. Identification of security requirements in systems of systems by functional security analysis. (Fuchs and Rieke, 2010)	IC2,IC3	JA
S18	Gagliardi, M., Wood, W., Klein, J. and Morley, J. A uniform approach for system of systems architecture evaluation. (Gagliardi et al., 2009)	IC2,IC3	JA
S19	Garro, A. and Tundis, A. On the Reliability Analysis of Systems and SoS: The RAMSAS Method and Related Extensions. (Garro and Tundis, 2014)	IC1,IC4	JA
S20	Ge, B., Hipel, K. W., Yang, K. and Chen, Y. A data-centric capability-focused approach for system-of-systems architecture modeling and analysis. (Ge et al., 2013)	IC2,IC3	JA
S21	Guariniello, C. and DeLaurentis, D. Communications, Information, and Cyber Security in Systems-of-Systems: Assessing the Impact of Attacks through Interdependency Analysis. (Guariniello and DeLaurentis, 2014a)	IC2,IC3	JA
S22	Guariniello, C. and DeLaurentis, D. Integrated Analysis of Functional and Developmental Interdependencies to Quantify and Trade-off Ilities for System-of-Systems Design, Architecture, and Evolution. (Guariniello and DeLaurentis, 2014b)	IC2,IC4	CP/JA
S23	Guo, F. and Wang, M. Quantitative measurement of interoperability by using Petri net. (Guo and Wang, 2012)	IC1,IC3	JA
S24	Iacobucci, J. and Mavris, D. A method for the generation and evaluation of architecture alternatives on the cloud. (Iacobucci and Mavris, 2011)	IC1,IC3	CP

continued on next page ...

#	**Citation**	**IC**	**Type**
S25	Joyce, H., Hin, O., Seng, Y. and Chan, S. A systems assurance perspective towards generic systems engineering. (Joyce et al., 2011)	IC1,IC3	CP
S26	Kazman, R. b., Gagliardi, M. and Wood, W. Scaling up software architecture analysis. (Kazman et al., 2012)	IC2,IC3	JA
S27	Kimura, D., Osaki, T., Yanoo, K., Izukura, S., Sakaki, H. and Kobayashi, A. Evaluation of IT systems considering characteristics as system of systems. (Kimura et al., 2011)	IC2,IC3	CP
S28	Leuchter, S., Reinert, F. and MÃ¼ller, W. Assessment of the integration capability of system architectures from a complex and distributed software systems perspective. (Leuchter et al., 2014)	IC2,IC3	CP
S29	Lin, Q., Cai, Z., Wang, Y., Yang, J. and Chen, L. Adaptive flight control design for quadrotor UAV based on dynamic inversion and neural networks. (Lin et al., 2013)	IC2,IC3	CP
S30	Madni, A. and Sievers, M. System of systems integration: Key considerations and challenges. (Madni and Sievers, 2014)	IC1,IC3	JA
S31	Michael, J., Riehle, R. and Shing, M.T. The verification and validation of software architecture for systems of systems. (Michael et al., 2009)	IC2,IC4	CP
S32	Nair, S., De La Vara, J., Sabetzadeh, M. and Briand, L. An extended systematic literature review on provision of evidence for safety certification. (Nair et al., 2014)	IC1,IC3	JA
S33	Oliveira, M. and Pereira, J. Extensible Virtual Environment Systems Using System of Systems Engineering Approach. (Oliveira and Pereira, 2007)	IC2,IC3	CP
S34	Osmundson, J. and Langford, G. Connections in system of systems. (Osmundson and Langford, 2012)	IC1,IC4	CP
S35	Ramaswamy, A. b., Monsuez, B. and Tapus, A. Formal models for cognitive systems. (Ramaswamy et al., 2013)	IC2,IC3	CP
S36	Rezaei, R., Chiew, T. and Lee, S. An interoperability model for ultra large scale systems. (Rezaei et al., 2014)	IC1,IC4	JA
S37	Schneider, D., Becker, M. and Trapp, M. Approaching runtime trust assurance in open adaptive systems. (Schneider et al., 2011)	IC2,IC3	CP
S38	Schneider, D. and Trapp, M. A safety engineering framework for open adaptive systems. (Schneider and Trapp, 2011)	IC1,IC3	CP

continued on next page ...

#	**Citation**	**IC**	**Type**
S39	Schneider, D. and Trapp, M. Runtime safety models in open systems of systems. (Schneider and Trapp, 2009)	IC2,IC3	CP
S40	Schugerl, P., Rilling, J., Witte, R. and Charland, P. A quality perspective of software evolvability using semantic analysis. (Schugerl et al., 2009)	IC2,IC4	CP
S41	Shukla, M. and Asundi, J. Considering emergency and disaster management systems from a software architecture perspective. (Shukla and Asundi, 2012)	IC2,IC3	JA
S42	Singh, A. and Dagli, C. H. Multi-objective stochastic heuristic methodology for tradespace exploration of a network centric system of systems. (Singh and Dagli, 2009)	IC2,IC3	CP
S43	Stratton, W., Sibol, D., Lindvall, M., Ackermann, C. and Godfrey, S. Developing an approach for analyzing and verifying system communication. (Stratton et al., 2009)	IC2,IC3	CP
S44	Tsadimas, A., Kapos, G.-D., Dalakas, V., Nikolaidou, M. and Anagnostopoulos, D. Integrating simulation capabilities into SysML for enterprise information system design. (Tsadimas et al., 2014)	IC1,IC3	CP
S45	Tucker, A. and Dagli, C. Design of experiments as a means of lean value delivery to the flight test enterprise. (Tucker and Dagli, 2009)	IC2,IC3	JA
S46	Urwin, E., Venters, C., Russell, D., Liu, L., Luo, Z., Webster, D., Henshaw, M. and Xu, J. Scenario-based design and evaluation for capability. (Urwin et al., 2010)	IC2,IC3	CP
S47	Wada, H., Suzuki, J. and Oba, K. A model-driven development framework for non-functional aspects in service oriented architecture. (Wada et al., 2008)	IC2,IC4	JA
S48	Walraven, S., Lagaisse, B., Truyen, E. and Joosen, W. Dynamic Composition of Cross-organizational Features in Distributed Software Systems. (Walraven et al., 2010)	IC2,IC4	CP
S49	Xia, X., Wu, J., Liu, C. and Xu, L. A Model-Driven Approach for Evaluating System of Systems. (Xia et al., 2013)	IC2,IC3	CP
S50	Zafar, N., Arnautovic, E., Diabat, A. and Svetinovic, D. System Security Requirements Analysis: A Smart Grid Case Study. (Zafar et al., 2014)	IC1,IC3	JA

continued on next page ...

#	Citation	IC	Type
S51	Zhu, L., Staples, M. and Jeffery, R. Scaling up software architecture evaluation processes. (Zhu et al., 2008)	IC2,IC4	JA
S52	Zuccato, A. b., Daniels, N., Jampathom, C. and Nilson, M. Report: Modular safeguards to create holistic security requirement specifications for system of systems. (Zuccato et al., 2010)	IC1,IC4	JA

3.3 Phase 3 - Analysis

In this section the analytical results of our SLR is presented. The results were based on the data extraction performed in the previous phase.

First of all, our results indicated that: (i) research on quality attributes for SoS is in increasing during the last years (see Figure 3.1); (ii) the topic quality attributes for SoS has has been investigated by small groups (see Figure 3.2). Analysing the maturity of the research, the included studies indicated that the research is overall on a early stage since most studies were classified as "case study" and "without validation" (see Figure 3.3).

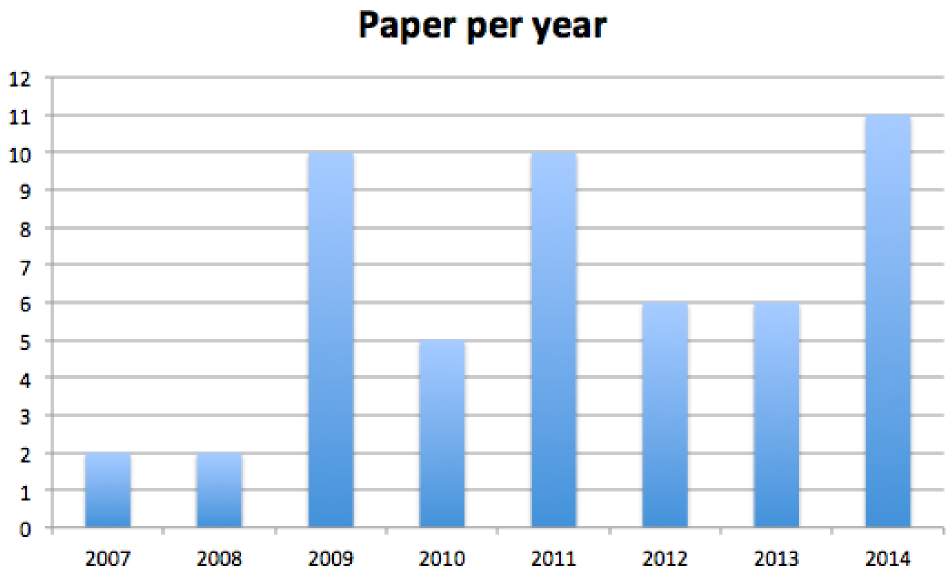

Figure 3.1: Studies included per year.

Considering the RQ1, our results pointed that the five most relevant quality on SoS research are: security, interoperability, performance, relialibity, and safety. The complete ranked list of quality attributes is shown in Figure 3.4.

Considering the RQ2, the military domain is the target of most studies. This is understandable since SoS started to gain their popularity in military domain.

However, new application scenarios of SoS have been considered as it is presented in Figure 3.5 that highlights the domains "IT Systems", "Smart Grids" and "Automotive".

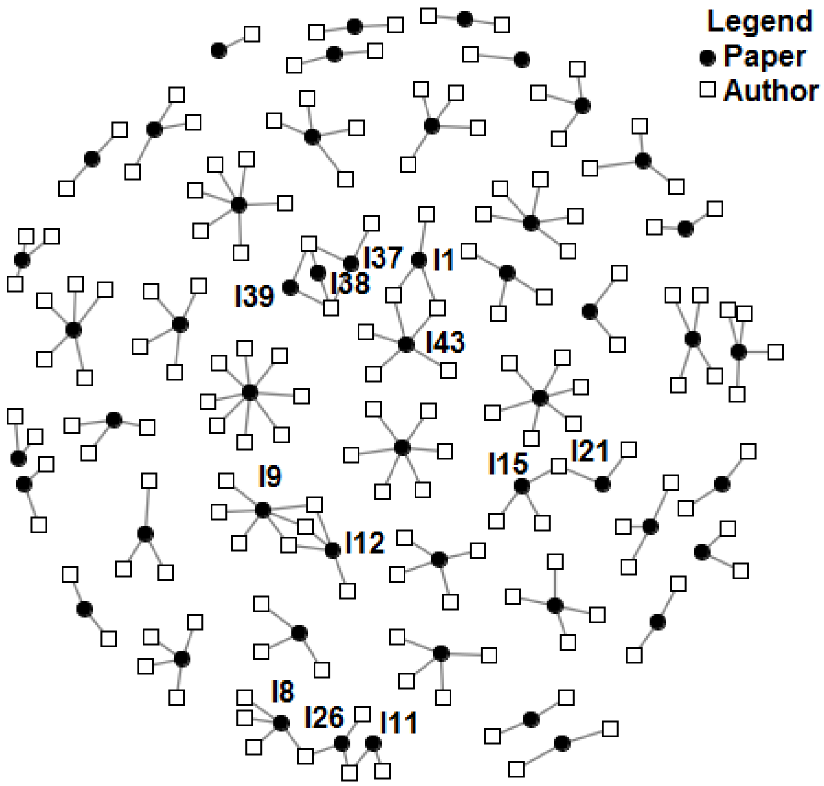

Figure 3.2: Authorship relationship between the studies.

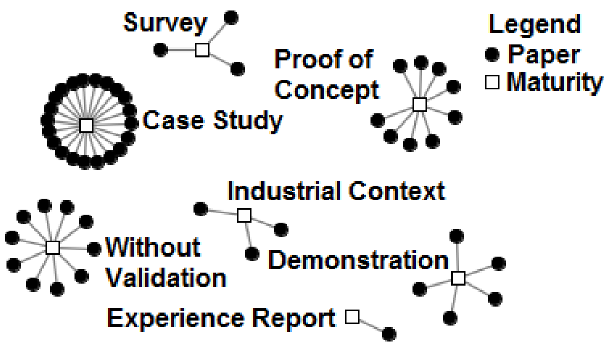

Figure 3.3: Maturity of the included studies.

On the IT Systems domain, studies S10, S27, S44 state that this systems have increasingly incorporated SoS characteristics due to their integration requirements with third party systems. More specifically, study S27 indicates that this is a tendency due to popularization of cloud computing on the industry. According to studies S42, S47 and S50, Smart Grids domain has been investigated in the context of energy management systems that have as main priority the integration with communication services in order to detect and address incidents before they compromise the power offering. Finally, the Automotive domain is addressed by studies S4, S14 and S17 which state that a modern automotive industry is typically driven by the integration of more than 50 embedded computers, also known as ECUs (Electronic Control Units). More specifically, study S17 investigated ve-

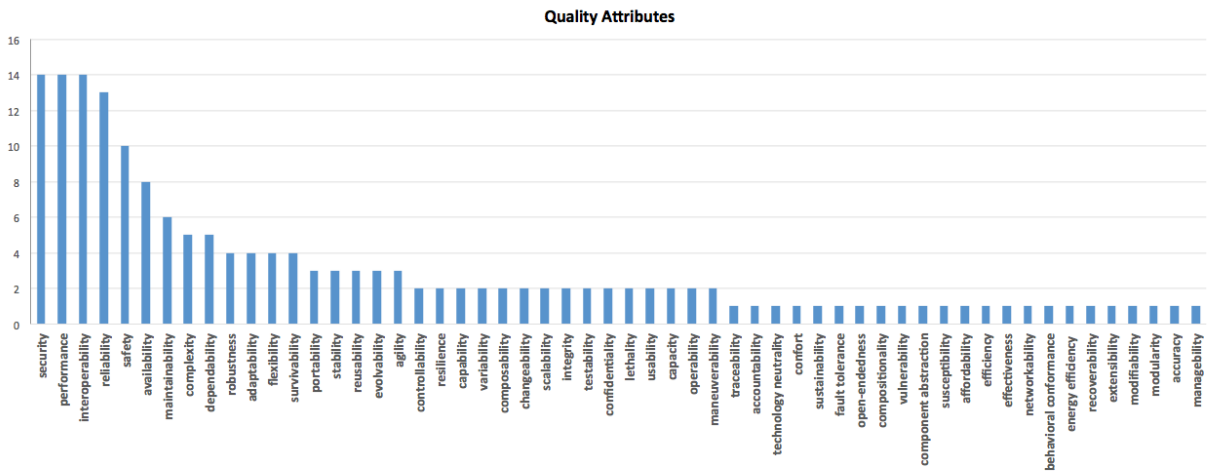

Figure 3.4: Quality Attributes identified in the included studies.

hicles and roadside units that can communicate in *ad hoc* way to exchange information, such as safety warnings and traffic information, in order to avoid accidents and traffic congestion.

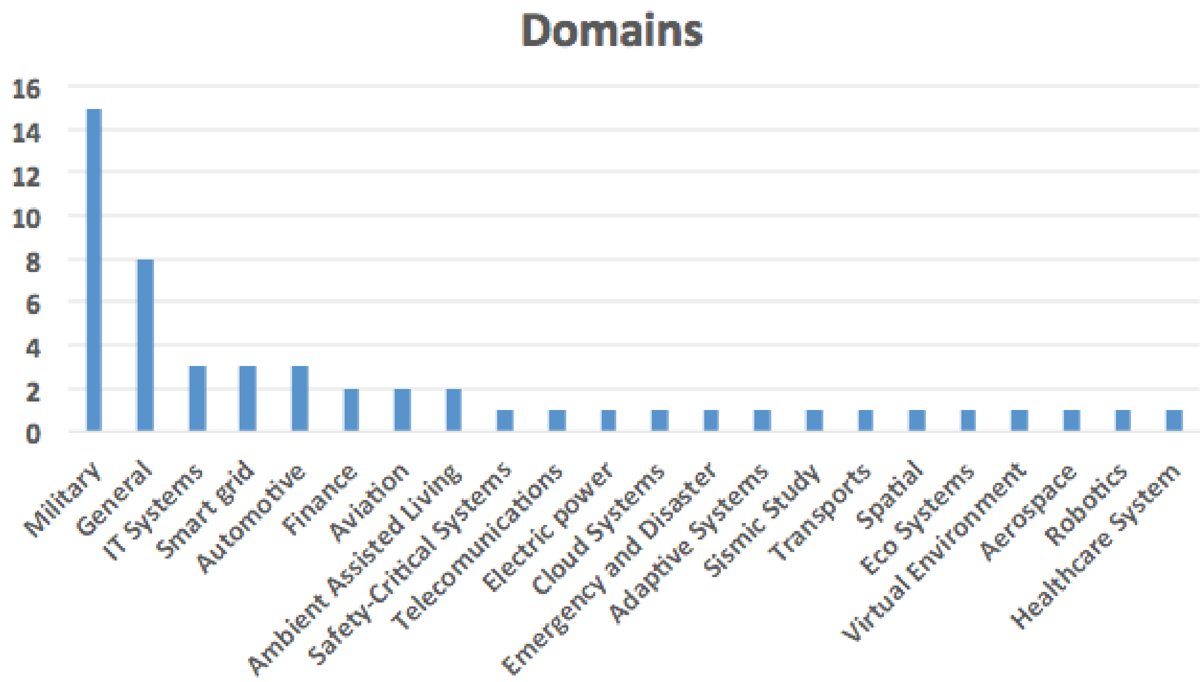

Figure 3.5: Application Domains identified on the included studies.

Considering the RQ3, our results indicates that the domains addressed on RQ2 are concerned with a similar set of quality attributes as can be seen in Figures 3.6, 3.7 and 3.8.

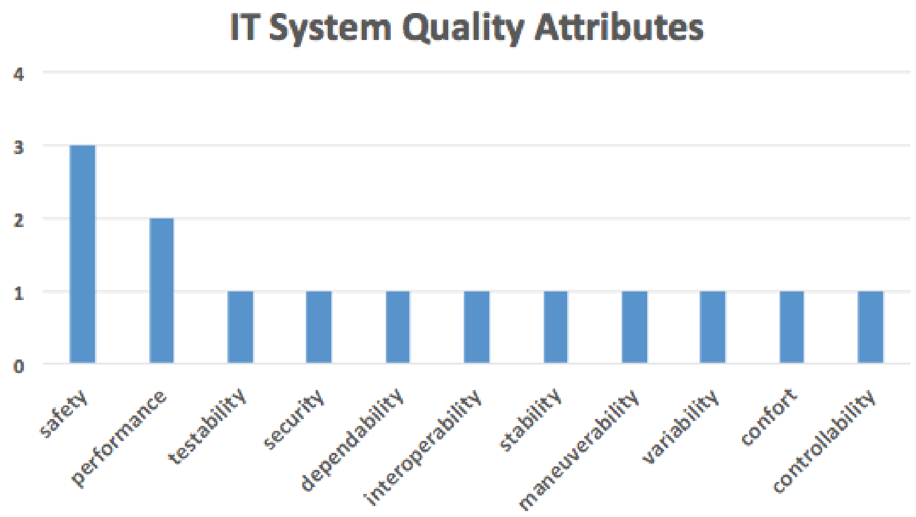

Figure 3.6: Quality attributes identified on the Information Technology domain.

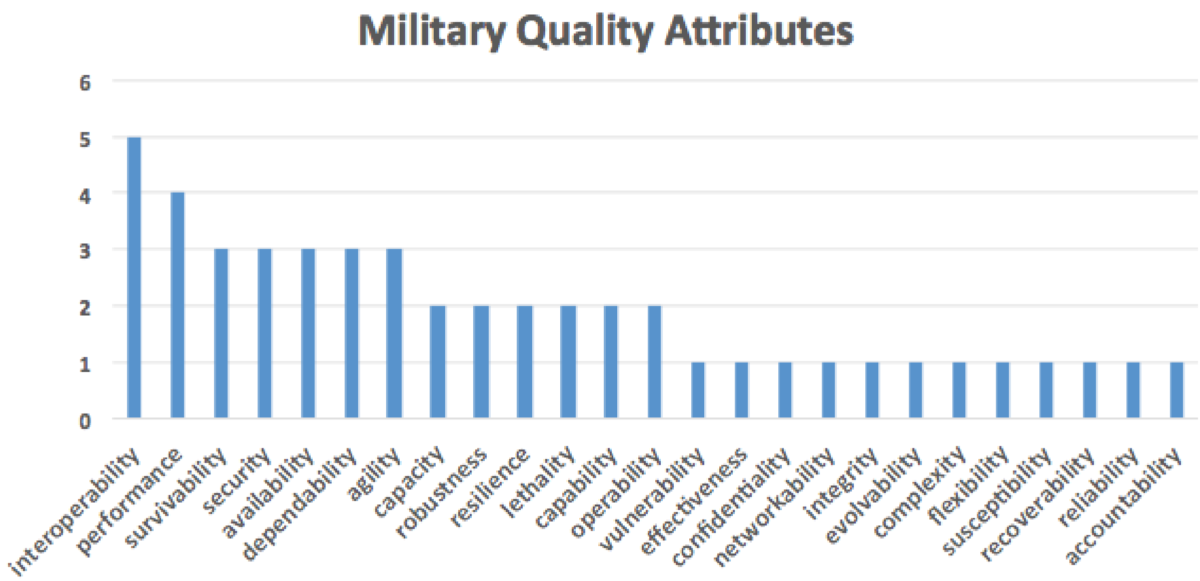

Figure 3.7: Quality attributes identified on the Military domain.

3.4 Threats to Validity

The main threats identified to the validity of our SLR are described as follows:

- **Missing of important primary studies:** The search for studies related to SoS quality attributes was conducted in several publication databases and search engines. According to Dyba et al. (2005) and Kitchenham and Charters (2007), the publication databases that were used are the most relevant sources. Aiming at not missing any important evidence, it was also considered the specialist suggestion and the related works which were presented in the reference list of the selected primary studies. In addition, no limit was placed on the date of publications. During the

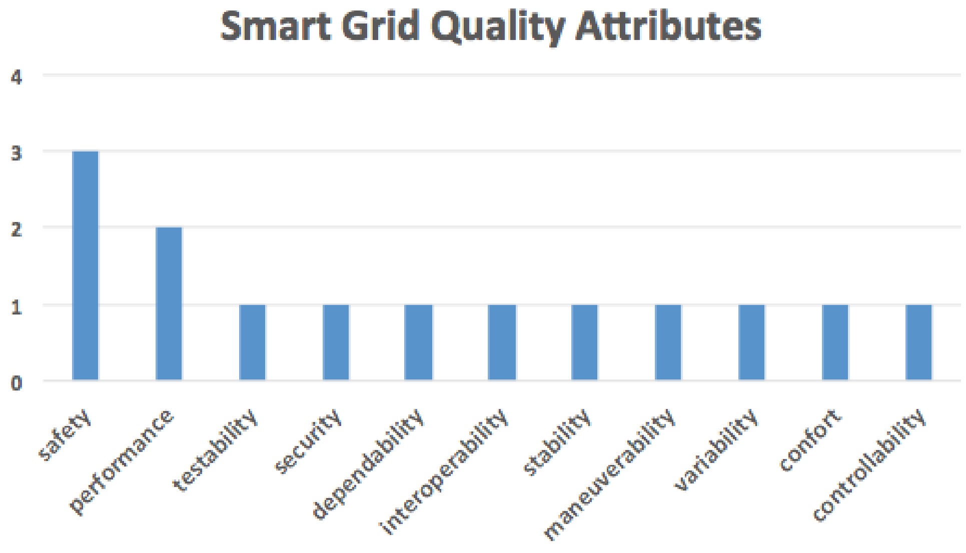

Figure 3.8: Quality attributes identified on the Smart Grid domain.

search, conference papers, journals, and technical reports were also considered. In spite of the effort to included all relevant evidence in this research, it is possible that primary studies were missed;

- **Reviewers reliability:** In our SLR, all reviewers are researchers in the Software Engineering area. Furthermore, none of primary study was published by research group or researchers related to the authors of this work. Therefore, the authors of this work are not aware of any bias that may have been introduced during the analysis process. However, it might be possible that the conclusion about the studies evaluated have been influenced by the opinion of the reviewers;

- **Data extraction:** Another threat to this review refers to how the data were extracted from the primary studies, since not all the information were obvious to answer the research questions and some data had to be interpreted. In order to ensure the validity of our SLR, other sources of information were analyzed, i.e., technical reports, web sites, and manuals, in addition to the primary studies analyzed. Furthermore, in the event of a disagreement between the reviewers, a discussion was conducted to ensure that a full agreement was reached; and

- **Quality assessment:** Since the goal of our SLR was to identify studies related to SoS quality attributes, no quality assessment was performed, as it might restrict the number of primary studies included. It is an agreement that a quality assessment can provide more insights and explanations to the conclusion of this review. Thus, it will be included in a future version of this work.

As it can be observed, the authors of this work are concerned with the validity of this SLR results. In particular, they have dedicated special effort to completely cover this research area as impartial as possible.

3.5 Discussion

This section presents a discussion about the five most relevant quality attributes for SoS found by our SLR: security, interoperability, performance, relialibity and safety. The focus of this analysis is to address how the known quality models, specificly ISO/IEC 25010, can support the specification and the evaluation of these attributes considering the characteristics and challenges of SoS.

Based on our results it is possible to affirm that besides military domain, new application scenarios of SoS have emerged. Some of these scenarios are related with safety-critical systems in which failure may cause death or injury to people, harm to the environment, or substantial finantial losses (Bozzano and Villafiorita, 2010). In these situations, the SoS must be able to react appropriately to dynamic changes in order to assure its behavior and quality (Schneider et al., 2011). Therefore, still according to (Schneider et al., 2011) it is very difficult to assure quality properties like safety, reliability, performance and security for the whole SoS since it is hard to guarantee such properties for all constituent systems. Thus, dependencies, trade-offs and relationships among the quality attributes should become more complex for SoS.

Regarding the quality attributes definitions, this work has found that some well established quality attributes definitions, such as reliability, can not be fully applied in the SoS context. (Garro and Tundis, 2014) highlight that this problem happens because in SoS, the concept of mission failure is not so easily identifiable in comparison with monolithic systems in which failure scenarios and their effects can be clearly identified. (Garro and Tundis, 2014) also exemplifies: consider a military system operation in a rescue mission of a man overboard, if a helicopter fails due to an unexpected breakdown, the SoS is still able to perform its mission once it can contact a ship to perform the same activity even with degraded overall performance. However, only when the performance achieves a unacceptable level it can be faced as an SoS failure. Thus, the SoS reliability is a wider and more flexible concept that should be taken into account considering the flexible and dynamic nature of SoS (Garro and Tundis, 2014).

Considering the "safety" quality attribute, the results of our SLR investigation show that this attribute is an essential quality characteristic or even the main goal of most SoS. According to (ISO/IEC, 2010), safety is the degree to which a product or system mitigates the potential risk to people in the intended context of its usage. In SoS, these potential risks can become hazardous situations due to failures in any of the constituent systems which shows that availability, reliability, performance and security attributes must be properly addressed for each constituent system (Schneider et al., 2011). It is important to note that these quality attributes dependencies and relationships are not properly considered in the hierarchical structure found in ISO/IEC 25010 and other quality models.

Considering the “interoperability” quality attribute, it is noticed that it can be handled as cross-cutting concern that has a unique and crucial coordination role relative to the others quality concerns (Rothenberg, 2008). However, it is important to highlight that an SoS is usually conceived without consider the interoperability of its constituent systems on the early stages of development (Madni and Sievers, 2014). In addition, the metrics established by the current quality models to measure interoperability can not directly deal with the SoS characteristics (Guariniello and DeLaurentis, 2014b). This is explained by the emergent behavior of SoS that makes difficult to capture and evaluate interoperability at the level of constituent systems (Meilich, 2006).

From that, it is important to consider the interdependence between quality attributes in scenarios that cascading failures and bottlenecks could result in a complete SoS blackout (Chiprianov et al., 2014). To do that, the quality attributes needs to be measured and controlled for each constituent system in order to address the impact on SoS quality (Chiprianov et al., 2014; Gorod et al., 2007; Waller and Craddock, 2011).

Therefore, a first analysis about the ISO/IEC 25010 coverage of SoS quality attributes shows that 48% of the quality attributes commonly considered in SoS are not addressed as can be seen in Table 3.4. So, nowadays, SoS have been developed without considering some important criteria on quality attributes evaluation that were not been properly addressed by a standardized quality model.

Table 3.4: ISO/IEC 25010 quality attributes coverage

#	Quality Attributes	Among	Covered by ISO
1	performance	14	yes
2	security	14	yes
3	interoperability	14	yes
4	reliability	13	yes
5	safety	10	yes
6	availability	8	yes
7	maintainability	6	yes
8	complexity	5	no
9	dependability	5	no

continued on next page ...

#	Quality Attributes	Among	Covered by ISO
10	robustness	4	no
11	survivability	4	no
12	flexibility	4	yes
13	adaptability	4	yes
14	agility	3	no
15	reusability	3	yes
16	stability	3	no
17	evolvability	3	no
18	portability	3	yes
19	changeability	2	no
20	operability	2	yes
21	lethality	2	no
22	controllability	2	no
23	composability	2	no
24	integrity	2	yes
25	scalability	2	yes
26	testability	2	yes
27	capability	2	no
28	confidentiality	2	yes
29	variability	2	no
30	usability	2	yes
31	capacity	2	yes
32	resilience	2	no
33	maneuverability	2	no
34	open-endedness	1	no
35	susceptibility	1	no
36	extensibility	1	no
37	fault tolerance	1	yes
38	networkability	1	no
39	behavioral conformance	1	no
40	vulnerability	1	no
41	confort	1	yes
42	technology neutrality	1	no
43	component abstraction	1	no

continued on next page ...

#	**Quality Attributes**	**Among**	**Covered by ISO**
44	sustainability	1	yes
45	affordability	1	no
46	traceability	1	no
47	effectiveness	1	yes
48	effectiveness	1	yes
49	energy efficiency	1	yes
50	managebility	1	no
51	accountability	1	yes
52	accuracy	1	yes
53	recoverability	1	yes
54	modifiability	1	yes
55	compositionality	1	no
56	modularity	1	yes

Chapter

4

Conclusion and Future Work

This study aimed, through a SLR, to identify the most common quality attributes in SoS context, considering different application domains. As a result, 56 quality attributes were identified. Besides that, it is possible to confirm a migration trend from traditional military domain research to new application domains, such as automotive, IT systems, and smart grids. Moreover, from analysis of the five most relevant quality attributes, this study pointed out limitations of the current quality models, specifically, the well established and widely used ISO/IEC 25010. SoS quality attributes have also complex interdependencies that can not be translated by a hierarchical structure. Besides that, some well-established definitions for each quality attribute can not be fully applied in the SoS context due to the flexible, dynamic nature of these systems. As future work, we aim to perform a detailed analysis about the concepts and definitions of the quality attributes found in this study in order to propose a suitable quality model for SoS. With that, we intend that issues related to quality attributes interdependencies can be properly addressed during development, maintenance, and even evolution of SoS.

Acknowledgements

This work is supported by Brazilian funding agencies: FAPESP (Process N. 2014/02244-7), CAPES, and CNPq.

References

Ackermann, C.; Lindvall, M.; Cleaveland, R. Towards behavioral reflexion models. In: *ISSRE 2009*, 2009, p. 175–184.

AL-Badareen, A.; Selamat, M.; A. Jabar, M.; Din, J.; Turaev, S. Software quality models: A comparative study. In: Mohamad Zain, J.; Wan Mohd, W.; El-Qawasmeh, E., eds. *Software Engineering and Computer Systems*, Communications in Computer and Information Science, Springer Berlin Heidelberg, p. 46–55, 2011.

Alghamdi, A.; Hussain, T.; Khan, G. Enhancing c4i security using threat modeling. In: *UKSim 2010*, 2010, p. 131–136.

Allen, M. From substandard to successful software. *CrossTalk*, p. 29–32, 2009.

Aoyama, M.; Tanabe, H. A design methodology for real-time distributed software architecture based on the behavioral properties and its application to advanced automotive software. In: *APSEC 2013*, APSEC '11, IEEE Computer Society, 2011, p. 211–218 (*APSEC '11*,).

Azizian, N.; Mazzuchi, T.; Sarkani, S.; Rico, D. A framework for evaluating technology readiness, system quality, and program performance of u.s. dod acquisitions. *Systems Engineering*, p. 410–426, 2011.

Babar, M.; Chauhan, M. A tale of migration to cloud computing for sharing experiences and observations. In: *SECLOUD 2011*, 2011, p. 50–56.

Balci, O.; Arthur, J. D.; Ormsby, W. F. Achieving reusability and composability with a simulation conceptual model. *Journal of Simulation*, p. 157–165, 2011.

Batista, T. Challenges for sos architecture description. In: *SESoS 2013*, 2013, p. 35–37.

Belloir, N.; Chiprianov, V.; Ahmad, M.; Munier, M.; Gallon, L.; Bruel, J.-M. Using relax operators into an mde security requirement elicitation process for systems of systems. In: *ECSAW 2014*, ACM, 2007, p. 1–4.

Boardman, J.; Sauser, B. System of systems - the meaning of of. In: *SoSE 2006*, 2006.

Boehm, B.; Lane, J. A. 21st Century Processes for Acquiring 21st Century Software-Intensive Systems of Systems. *Crosstalk: Journal of Defence Software Engineering*, p. 4–9, 2006.

Boehm, B. W.; Brown, J. R.; Kaspar, J. R.; Lipow, M. L.; MacCleod, G. Characteristics of software quality. *American Elsevier*, 1978.

Bozzano, M.; Villafiorita, A. *Design and safety assessment of critical systems.* Taylor & Francis, 2010.

Brøndum, J.; Zhu, L. Towards an architectural viewpoint for systems of software intensive systems. In: *SHARK 2010*, 2010, p. 60–63.

Calinescu, R.; Kikuchi, S.; Johnson, K. Compositional reverification of probabilistic safety properties for large-scale complex it systems. *Lecture Notes in Computer Science*, p. 303–329, 2012.

Carbon, R.; Johann, G.; Muthig, D.; Naab, M. A Method for Collaborative Development of Systems of Systems in the Office Domain. In: *EDOC 2008*, 2008, p. 339–345.

Chen, H.-M.; Kazman, R. Architecting ultra-large-scale green information systems. In: *GREENS 2012*, 2012, p. 69–75.

Chiprianov, V.; Falkner, K.; Gallon, L.; Munier, M. Towards modelling and analysing non-functional properties of systems of systems. In: *SOSE 2014*, 2014, p. 289–294.

Colin, C.; Duncan, J.; Lu, L.; Zongyang, L.; David, E.; Jie, X. A scenario-based architecture evaluation framework for network enabled capability. In: *COMPSAC 2009*, 2009, p. 9–12.

Department of Defense *System engineering guide for systems of systems.* Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, 2008.

Department of Defense Dodaf architecture framework version 2.02. [On-line], World Wide Web, 2010.

Dieste, O.; Grimán, A.; Juristo, N. Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, p. 513–539, 2009.

DoD *Systems engineering guide for system of systems.* Technical Report, Department of Defense, 2008.

Dyba, T.; Kitchenham, B. A.; Jorgensen, M. Evidence-Based Software Engineering for Practitioners. *IEEE Software*, p. 58–65, 2005.

Eklund, U. c.; Bosch, J. Architecture for embedded open software ecosystems. *Journal of Systems and Software*, p. 128–142, 2014.

Fang, Z.; DeLaurentis, D.; Davendralingam, N. An approach to facilitate decision making on architecture evolution strategies. *Procedia Computer Science*, p. 275 – 282, 2013.

Farcas, C.; Farcas, E.; Krueger, I.; Menarini, M. Addressing the Integration Challenge for Avionics and Automotive Systems: From Components to Rich Services. *Proceedings of the IEEE*, p. 562–583, 2010.

Fischer, D.; Sarkarati, M.; Spada, M.; Michelbach, T.; Urban, W.; Tueffers, C. An application security framework for soa-based mission data systems. In: *SMC-IT 2011*, 2011, p. 53–60.

Fuchs, A.; Rieke, R. Identification of security requirements in systems of systems by functional security analysis. *Lecture Notes in Computer Science*, p. 74–96, 2010.

Gagliardi, M.; Wood, W.; Klein, J.; Morley, J. A uniform approach for system of systems architecture evaluation. *CrossTalk*, p. 12–15, 2009.

Garro, A.; Tundis, A. On the reliability analysis of systems and sos: The ramsas method and related extensions. *IEEE Systems Journal*, p. 1–10, 2014.

Ge, B.; Hipel, K. W.; Yang, K.; Chen, Y. A data-centric capability-focused approach for system-of-systems architecture modeling and analysis. *Systems Engineering*, p. 363–377, 2013.

Gorod, A.; Gove, R.; Sauser, B.; Boardman, J. System of systems management: A network management approach. In: *SoSE 2007*, 2007, p. 1–5.

Grady, R. B.; Caswell, D. L. *Software metrics: Establishing a company-wide program.* Prentice-Hall, Inc., 1987.

Grondin, G.; Faure, M.; Urtado, C.; Vauttier, S. Mission-oriented Autonomic Configuration of Pervasive Systems. In: *7th International Conference on Software Engineering Advances (ICSEA)*, 2012.

Guariniello, C.; DeLaurentis, D. Communications, information, and cyber security in systems-of-systems: Assessing the impact of attacks through interdependency analysis. *Procedia Computer Science*, p. 720 – 727, 2014a.

Guariniello, C.; DeLaurentis, D. Integrated analysis of functional and developmental interdependencies to quantify and trade-off ilities for system-of-systems design, architecture, and evolution. *Procedia Computer Science*, p. 728 – 735, 2014b.

Guo, F.; Wang, M. Quantitative measurement of interoperability by using petri net. *Journal of Computational Information Systems*, p. 3245–3252, 2012.

Iacobucci, J.; Mavris, D. A method for the generation and evaluation of architecture alternatives on the cloud. In: *SoSE 2011*, 2011, p. 137–142.

ISO/IEC *Iso/iec 25010 system and software quality models.* Technical Report, 2010.

Jamshidi, M. *Systems of systems engineering: Principles and applications.* Taylor & Francis, 2008.

Joyce, H.; Hin, O.; Seng, Y.; Chan, S. A systems assurance perspective towards generic systems engineering. *INCOSE 2011*, p. 742–761, 2011.

Kazman, R. b.; Gagliardi, M.; Wood, W. Scaling up software architecture analysis. *Journal of Systems and Software*, p. 1511–1519, 2012.

Kimura, D.; Osaki, T.; Yanoo, K.; Izukura, S.; Sakaki, H.; Kobayashi, A. Evaluation of it systems considering characteristics as system of systems. In: *SoSE 2011*, 2011, p. 43–48.

Kitchenham, B. A.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Technical Report, Keele University and Durham University Joint Report, 2007.

Lane, J. A. What is a system of systems and why should i care? 2013.

Leuchter, S.; Reinert, F.; MÃ¼ller, W. Assessment of the integration capability of system architectures from a complex and distributed software systems perspective. In: *SPIE 2014*, 2014.

Lin, Q.; Cai, Z.; Wang, Y.; Yang, J.; Chen, L. Adaptive flight control design for quadrotor uav based on dynamic inversion and neural networks. In: *IMCCC 2013*, 2013, p. 1461–1466.

Luzeaux, D.; Ruault, J. *Systems of systems*. ISTE. Wiley, 2010.

Madni, A.; Sievers, M. System of systems integration: Key considerations and challenges. *Systems Engineering*, p. 330–347, 2014.

Maier, M. W. Architecting principles for systems-of-systems. *Systems Engineering*, p. 267–284, 1998.

McCall, J. A.; Richards, P. K.; Walters, G. F. Factors in software quality. *Rome Air Development Center Air Force Systems Command*, 1977.

Meilich, A. System of systems (sos) engineering amp; architecture challenges in a net centric environment. In: *SoSE 2006*, 2006, p. 5 pp.–.

Michael, J.; Riehle, R.; Shing, M.-T. The verification and validation of software architecture for systems of systems. In: *SoSE 2009*, 2009, p. 1–6.

Nair, S.; De La Vara, J.; Sabetzadeh, M.; Briand, L. An extended systematic literature review on provision of evidence for safety certification. *Information and Software Technology*, p. 689–717, 2014.

Nakagawa, E. Y.; Gonçalves, M.; Guessi, M.; Oliveira, L. B. R.; Oquendo, F. The state of the art and future perspectives in systems of systems software architectures. *SESoS 2013*, p. 13–20, 2013.

Oliveira, M.; Pereira, J. Extensible virtual environment systems using system of systems engineering approach. In: *ICAT 2007*, 2007, p. 89–96.

Osmundson, J.; Langford, G. Connections in system of systems. In: *INCOSE 2012*, 2012, p. 831–869.

Petcu, V.; Petrescu, A. Systems of systems applications for telemedicine. In: *RoEduNet 2010*, 2010, p. 208–211.

Ramaswamy, A. b.; Monsuez, B.; Tapus, A. Formal models for cognitive systems. In: *ICAR 2013*, 2013.

Ravichandran, T.; Rothenberger, M. A. Software reuse strategies and component markets. *Commun. ACM*, p. 109–114, 2003.

Rezaei, R.; Chiew, T.; Lee, S. An interoperability model for ultra large scale systems. *Advances in Engineering Software*, p. 22–46, 2014.

Rothenberg, J. Interoperability as a cross-cutting concern. 2008, p. .

Santos, D. S.; Oliveira, B.; Guessi, M.; Oquendo, F.; Delamaro, M.; Nakagawa, E. Y. Towards the evaluation od system of systems softaware architecture. *WDES 2014*, p. 1–6, 2014.

Schneider, D.; Becker, M.; Trapp, M. Approaching runtime trust assurance in open adaptive systems. In: *SEAMS 2011*, 2011, p. 196–201.

Schneider, D.; Trapp, M. Runtime safety models in open systems of systems. In: *DASC 2009*, 2009, p. 455–460.

Schneider, D.; Trapp, M. A safety engineering framework for open adaptive systems. In: *SASO 2011*, 2011, p. 89–98.

Schugerl, P.; Rilling, J.; Witte, R.; Charland, P. A quality perspective of software evolvability using semantic analysis. In: *ICSC 2009*, 2009, p. 420–427.

Shukla, M.; Asundi, J. Considering emergency and disaster management systems from a software architecture perspective. *International Journal of System of Systems Engineering*, p. 129–141, 2012.

Singh, A.; Dagli, C. H. Multi-objective stochastic heuristic methodology for tradespace exploration of a network centric system of systems. In: *Systems Conference 2009*, 2009, p. 218–223.

Stratton, W.; Sibol, D.; Lindvall, M.; Ackermann, C.; Godfrey, S. Developing an approach for analyzing and verifying system communication. In: *Aerospace Conference 2009*, 2009, p. 1–13.

Tsadimas, A.; Kapos, G.-D.; Dalakas, V.; Nikolaidou, M.; Anagnostopoulos, D. Integrating simulation capabilities into sysml for enterprise information system design. In: *SoSE 2014*, 2014, p. 272–277.

Tucker, A.; Dagli, C. Design of experiments as a means of lean value delivery to the flight test enterprise. *Systems Engineering*, p. 201–217, 2009.

Urwin, E.; Venters, C.; Russell, D.; Liu, L.; Luo, Z.; Webster, D.; Henshaw, M.; Xu, J. Scenario-based design and evaluation for capability. In: *SoSE 2010*, 2010, p. 1–6.

Wada, H.; Suzuki, J.; Oba, K. A model-driven development framework for non-functional aspects in service oriented architecture. *International Journal of Web Services Research*, p. 1–31, 2008.

Wagner, S. *Software product quality control.* Springer, 2013.

Waller, A.; Craddock, R. Managing runtime re-engineering of a system-of-systems for cyber security. In: *SoSE 2011*, 2011, p. 13–18.

Walraven, S.; Lagaisse, B.; Truyen, E.; Joosen, W. Dynamic composition of cross-organizational features in distributed software systems. In: *Proceedings of the 10th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, DAIS'10, Springer-Verlag, 2010, p. 183–197 (*DAIS'10*,).

Xia, X.; Wu, J.; Liu, C.; Xu, L. A model-driven approach for evaluating system of systems. In: *ICECCS 2013*, 2013, p. 56–64.

Zafar, N.; Arnautovic, E.; Diabat, A.; Svetinovic, D. System security requirements analysis: A smart grid case study. *Systems Engineering*, p. 77–88, 2014.

Zhu, L.; Staples, M.; Jeffery, R. Scaling up software architecture evaluation processes. *Lecture Notes in Computer Science*, p. 112–122, 2008.

Zuccato, A. b.; Daniels, N.; Jampathom, C.; Nilson, M. Report: Modular safeguards to create holistic security requirement specifications for system of systems. *Lecture Notes in Computer Science*, p. 218–230, 2010.