Approaches for Describing Reference Architectures: A Systematic Mapping Study

Pedro Henrique Dias Valle[1], Lina Garcés[1], Milena Guessi[1], Silverio Martínez-Fernández[2], and Elisa Yumi Nakagawa[1]

[1] Institute of Mathematical and Computer Sciences,
University of São Paulo (ICMC-USP),
São Carlos, SP, Brazil
[2] Universitat Politècnica de Catalunya, BarcelonaTech
Barcelona, Spain
pedrohenriquevalle@usp.br,linamgr@icmc.usp.br,milena@icmc.usp.br,
smartinez@essi.upc.edu,elisa@icmc.usp.br

Abstract. Reference architectures have emerged as an important artifact of software engineering, since they provide knowledge and experience about how to design architectures of software systems in a specific domain, favoring their reuse, and standardization. Describing reference architectures still is a great challenge, since knowledge about the domain and the concrete architectures should be described and documented in a coherent, cohesive, unambiguous, and complete way. However, reference architectures practical use is difficult due mainly to problems in their descriptions. The main goal is evaluate existing approaches for describing RAs, analyzing six key aspects while describing RAs: type, adherence to ISO/IEC 42010, applicability, maturity, interoperability, and dynamism. For this, we conducted a systematic mapping and identified 21 primary studies with approaches to describe reference architectures. We evaluated 21 approaches to describe RAs, of which four approaches (S4, S7, S16, S17) adhered to at least half of the ISO/IEC standard, six (S3, S4, S5, S7, S8, and S19) achieved a medium technological readiness level, one (S4) adopted a standard to achieve interoperability among systems, and only one (S8) considered the dynamism in the description of RA. We highlight the need to propose methods for describing RAs, considering the guides and tools to support the description of RAs.

Keywords: Reference Architecture, Architecture Description, Software Engineering, Systematic Mapping, Secondary Study

1 Introduction

In the last years, the size and complexity of software systems have increased, and consequently, the design, specification, and analysis of overall system structures have become a critical issue [1]. Software architectures have contributed to software systems design, playing a fundamental role in determining the quality of systems, because they considered several characteristics of quality as interoperability, performance, portability, adaptability, and maintainability. According to Bass [2], a software architecture *"is the structure or structures of the system, which is composed of software components, the externally visible properties of those components, and the relationships between them".*

The Reference Architecture (RA) concept has emerged as an important type of software architecture [2]. A RA is defined as an architecture that aggregates knowledge about how to design software architectures of systems in a specific application or technical domain [2]. A RA is a more generic and abstract type of architecture that includes business rules, standards and legislations, domain knowledge, software and hardware elements, architectural styles and patterns, and best practices of software development, among others artifacts [3, 4]. Reference architectures provide templates and guidelines for designing software systems in a particular domain [5]. Therefore, the purpose of a RA is to guide the development, standardization, and evolution of systems [6], at the same that enhances reuse and reduces time spent during software activities. RAs have successfully supported the development of critical systems in industry in domains as automotive, healthcare, industry 4.0, agriculture, among others [7]. A RA can be established for *standardization* of concrete architectures or *facilitation* of the design of such architectures [3]. In this sense, a RA must be described to communicate reliably the knowledge it contains. Architecture descriptions are used to improve communication and co-operation of software teams, enabling them to work in an integrated and coherent fashion, and improve software development through reuse and standardization [4].

The information contained in the architecture description should be accessible to a wide range of stakeholders. The standard ISO/IEC 42010 provides a core ontology for the description of architectures [8] and defines that the architecture description can be used for the following purposes: (i) as basis for system design and development activities; (ii) as basis to analyze and evaluate alternative implementations of an architecture; (iii) as development and maintenance documentation; (iv) for documenting the characteristics, features and design of a system for potential clients, acquirers, owners, operators and integrators; and (v) for sharing lessons learned and reusing architectural knowledge through viewpoints, patterns, and styles. Some studies have been conducted to investigate what approaches have been used for describing software architectures [9] and [10]. However, they are focused on the overall scope of software architecture and do not present specificities of RAs description considering the impact that good descriptions have on the quality and value of RAs in practice.

In this paper, we present an overview of existing approaches for describing RAs. We analyzed their adherence to the standard ISO/IEC 42010, as well as their applicability and maturity. We also analyzed whether such approaches support the description of interoperability and dynamism in RAs, since these quality characteristics are now than ever considered as important for modern systems as those based on microservices architectures, IoT (Internet of Things), software ecosystems, big data, and Systems-of-Systems (SoS). This paper is organized as follows: Section 2 presents the protocol of the SMS conducted in this work. Section 3 describes the results of analyzing the existing approaches for RAs description. Section 4 discusses the important findings of our study. Section 5 states the main threats to the validity of this work. Finally, Section 6 presents the main conclusions and future work.

2 Systematic Mapping Study (SMS)

This SMS was conducted following the guidelines proposed by Petersen *et al.,* [11] and Kitchenham and Charters [12]. In this section, we present the protocol of this systematic mapping, showing the objective research questions and metrics, and strategies for selection criteria, and data extraction and analysis.

Objective: The main objective of this paper is to evaluate the approaches identified for describing RAs regarding their completeness, applicability, maturity, and analyze how interoperability and dynamism have been addressed by such approaches in the description of RAs.

Research Questions: In order to obtain evidence for addressing our objective, we defined six Research Questions (RQs). For each RQ, we established some metrics that were used to support the analysis of the results, as presented following:

RQ_1: Which approaches have been proposed to describe RAs?

RQ_2: Which is the adherence level of approaches to describe RAs to the standard ISO/IEC 42010?

RQ_3: How the approaches support the description of RAs?

RQ_4: Which is the level of maturity of the approaches to describe RAs?

RQ_5: How the approaches address interoperability and dynamism in the description of RAs?

Search Strategy: We selected the search terms, the most suitable words, synonyms, acronyms or alternative spelling within the research field for composed our search string. In the context of our study, we wanted to identify all studies that proposed any approaches for describing references architectures. In this sense, we identified the related terms, considering the plural form of all keywords and related terms. Following, we present our search string:

("*reference architecture?*") ***AND*** (*"software architecture?" or "software structure?" or "software design?" or "system architecture?" or "systems structure?" or "system design?"*)

We executed the search string in the following digital libraries: Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink. For complementary the set of studies, we perform manual searches in relevant conferences in the software architecture area.

Selection Criteria: We established one inclusion criteria and three exclusion criteria for selected the studies that were used to answer our RQs. *Inclusion Criteria*

(IC): (i) the work is mainly focused on an approach to support the description of RAs. *Exclusion Criteria (EC):* (i) the topic of the paper is not focused on a RA for software systems, but in other areas (e.g., enterprise architecture, or hardware architectures); (ii) the work only claims to present an approach to support RAs engineering, but it is not mainly focused on architecture description; and (iii) the work was not peer-reviewed.

Data Extraction and Analysis

We use an online data extraction form[3] for extracting data from each considered study. This form contains information related to each research question and metric. The dataset gathered from these forms supported the synthesis of the results. Data of each study were extracted by one researcher involved in this study, during the data extraction. When there was a case of doubt, discussions with other researchers were conducted. To draw conclusions and answer our research questions, we performed qualitative and qualitative analysis.

3 Results

This study was conducted by researchers with experience in RA, software architecture, software engineering, and systematic literature review. This study was conducted between January 2018 and August 2018. In order to conduct this SMS, we executed the following steps: (i) *perform search string in data libraries.* In this step, we obtained a total of 989 studies; (ii) *removing of duplicate studies.* In this step, we removed 400 duplicated studies, therefore, 589 studies remained for the next step; (iii) *first selection activity*, analyzing the title, abstract, and keywords and apply the selection criteria, 323 studies were excluded, and 266 studies were selected for detailed inspection; (iv) *second selection*, the full text of each study was read and the selection criteria were again applied. As a result, 17 primary studies were selected to be included in this SMS; and (v) *perform snowballing inspection in the related works.* We included 4 relevant works after performed the snowballing inspection. We manual search in the following conferences: IEEE International Conference on Software Architecture (ICSA), European Conference on Software Architecture (ECSA 2020), Brazilian Symposium on Components, Architectures and Software Reuse (SBCARS), and International Conference on Software Engineering (ICSE).

Therefore, we selected a final list with *21 studies*[4] as presented in Table 1. The first column shows the ID for each study, the second column contains the title of studies, the third column shows the distribution of these studies over the years, the fourth column presents the orientation of the approaches. Still on this table, the fifth column presents the types of approaches (i.e., process, method, model, framework, among others) identified, and finally, the sixth column shows the TRL for each analyzed approach.

Overview of approaches for describing RAs

[3] Available on: https://forms.gle/RFnaXExvEyASuNVB7

[4] References of included studies are available on: `https://bit.ly/2kzZ35J`

Table 1. Approaches for describing reference architectures

ID	Title	Year	Orientation	Type of Approaches	TRL	Conformance with ISO/IEC 42010
S1	A reference architecture for control of mechanical systems	1994	Academy	Process	TRL 2	No
S2	NSA's MISSI reference architecture - Moving from prose to precise specifications	1998	Academy	ADL	TRL 3	ADL (H)
S3	PuLSE-DSSA—a method for the development of software RAs	1998	Industry	Method	TRL 4	Model Kind (M)
S4	Describing, instantiating and evaluating a reference architecture: A case study	2003	Academy	Method	TRL 5	View (H), Viewpoint (H), Model Kind (M), Stakeholder (H), Concern(H)
S5	Definition of RAs based on existing systems	2004	Industry	Process	TRL 6	View (M), Viewpoint (L), Model Kind (H)
S6	An Approach to Reference Architecture Design for Different Domains of Embedded Systems	2008	Academy	Method	TRL 2	Model Kind (M)
S7	Architectural Knowledge in an SOA Infrastructure Reference Architecture	2009	Industry	Method	TRL 4	Viewpoint (M), Model Kind (L), Concern (L), Decisions (L), Rationale (L)
S8	Formal modelling and analysis of HLA architectural style	2010	Academy	Architectural Style	TRL 4	ADL (H)
S9	A Methodology for Developing an Agent Systems Reference Architecture	2011	Academy	Process	TRL 2	View (M), Model Kind (L), Stakeholder (L)
S10	A reference architecture for integrated EHR in Colombia	2011	Academy	Process	TRL 2	View (L), Viewpoint (L)
S11	Empirically-grounded RAs: A proposal	2011	Academy	Process	TRL 2	View (L), Stakeholder (L)
S12	A knowledge-based framework for RAs	2012	Academy	Framework	TRL 2	No
S13	A reference architecture template for software-intensive embedded systems	2012	Academy	Document Template	TRL 3	View (L), Stakeholder (M)
S14	RAModel: A Reference Model for RAs	2012	Academy	Model	TRL 3	No
S15	Towards a bottom-up development of RAs for smart energy systems	2013	Industry	Process	TRL 4	View (L)
S16	An approach for capturing and documenting architectural decisions of RAs	2014	Academy	Method	TRL 3	Viewpoint (M), Model Kind (L), Stakeholder (L), Decisions (M), Rationale (M)
S17	Development and Specification of a Reference Architecture for Agent-Based Systems	2014	Academy	Process	TRL 5	View (H), Model Kind (M), Stakeholder (M), Decisions (L)
S18	Modeling and reusing robotic software architectures: The HyperFlex toolchain	2014	Academy	Process	TRL 3	No
S19	Variability viewpoint to describe RAs	2014	Academy	Viewpoint	TRL 4	Viewpoint (H), Model Kind (L), Stakeholder (M)
S20	Design and Evaluation of a Customizable Multi-domain Reference Architecture on Top of Product Lines of Self-driving Heavy Vehicles: An Industrial Case Study	2015	Academy	Process	TRL 4	Model Kind (L), Stakeholder (L)
S21	Quality-based heuristic for optimal product derivation in Software Product Lines	2015	Academy	Process	TRL 4	View (L), Stakeholder (L)

Results presented in this section correspond to RQ$_1$. We observed that the first studies on RA description began to be investigated in the early 1990s and that the community has had a greater interest in this research topic in the last decade. It is worth to highlight that until the conduction of this mapping study (August 2018), the most recent approach was published in 2015. Regarding the venue of publication, 52% (S6, S8, S11, S12, S13, S14, S16, S18, S19, S20, and S21) of papers were published in conferences, while 24% (S1, S2, S3, S9, and S15) of papers were published in workshops. Only 10% (S10, and S17) of papers were published in the journal. Finally, 10% of papers (S4, and S5) were published as technical reports, and only 5% (S7) published as a book chapter.

Approaches orientation: we also checked whether the approaches were proposed in the academic or industry context. For this, we analyzed the authors' affiliation and how was performed the development and evaluation of such approaches. As a result, we found that 19% (S3, S5, S7, and S15) of papers were proposed in the industry. 81% (S1, S2, S4, S6, S8, S9, S10, S11, S12, S13, S14, S15, S16, S18, S19, S20, and S21) of papers were proposed and validated in the academic context.

Approaches domains: we analyzed whether the approaches for describing RAs were proposed for some specific domain or general-purpose. In short, 7 approaches were proposed for a specific domain, while 14 studies were proposed for general purpose. S1 was proposed for mechanical systems, S9 and S18 were proposed for agent-based systems domain, while S10 developed an approach for health domain. In S13, the authors proposed an approach for embedded systems domain, S15 was proposed for the robotic domain, and S20 proposed an approach for the automotive domain. It is worth highlighting that all the studies that were created for help describing RAs for a specific domain were proposed in the academic context.

Approaches types: we identified 8 approaches types that proposed some mechanism to support the description of RAs, namely, processes, methods, architecture description language (ADL), architectural style, framework, model, viewpoint, and document template. Process and methods were the most recurrent approaches found in our study.

Process ($\approx$ 48%): it defines *what* is required to be done, without specifying *how* each task must be performed. We identified 10 processes (S1, S5, S9, S10, S11, S15, S17, S18, S20, and S21) to support the description of RAs. S5 defined a RA using the experience of existing systems, while S15 proposed an incremental development of RA for smart energy systems. In the academic context, S1 built and assessed a RA for control of mechanical systems. S9 presented a process for developing a RA that documents agent-based systems from different system viewpoints, while S10 presented a process for developing a RA for the implementation of an integrated EHR (Electronic Health Record), S11 proposed a six-steps process which helps to systematically design RA. S17 proposed a process for developing agent systems RA, using an approach based on software engineering techniques adapted to study agent frameworks. S18 defined how RA can be exploited for building robotic applications. S20 presented a systematic way to design of RA involving all relevant external and internal stakeholders, development documents, low-level artifacts, and literature. Finally, S21 proposed a process to support architectural decisions in the derivation of specific products of a software product line (SPL) family according to customer requirements, from an optimal feasible solution among a set of feasible solutions.

Method ($\approx$ 24%): we identified 5 studies that proposed methods for supporting the description of RAs. It is important to note that S3 and S7 proposed methods in the industry context. In short, S3 proposed the systematic and iterative development of RAs for SPL, while S7 presented how to harvest architectural knowledge from industry projects to create RA. S4 proposed to describe a RA instantiating it into a software architecture by making implementation decisions and evaluating it for quality attributes. S6 proposed a service-based method for design in RAs. Finally, S16 presented a systematic method for capturing and documenting architectural decisions of RAs. The authors carried out a case study showing how their approach can be used for enhancing the architecture description of RAs.

ADL ($\approx$ 4%): S2 discussed the definition and modeling of RA and the notion of conformance. The authors demonstrated that Rapide ADL can specify interfaces, connections and operational constraints, and also to specify how it is related (or conforms) with other architectures or to implementations.

Architectural Style ($\approx$ 4%): it provides a way to reuse the design decisions and constraints that are applied to an architecture to induce chosen desirable qualities. S8 was the only study in this line proposing a formal model for HLA (High Level Architectural) style.

Framework ($\approx$ 4%): it is a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful. S12 is the only study that presented a framework, named KRAF (knowledge-based Reference Architecture Framework), for providing a better understanding of the contents of RAs, enabling the development of more complete, efficient RAs.

Model ($\approx$ 4%): S14 presented a reference model for RA, named RAModel (Reference Architecture Model), that intends to improve the understanding about what RAs are, as well as their components and relationships, assisting the establishment, use, and evolution of such architectures.

Viewpoint ($\approx$ 4%): it is a representation of one or more aspects of an architecture that illustrates how the architecture addresses the concerns held by one or more of its stakeholders [13]. S19 proposed an architecture viewpoint to represent variability in RA. The authors described how they created such a viewpoint.

Document template ($\approx$ 4%): S13 presented a document template for RAs for the domain of software-intensive embedded systems. The template addresses the somewhat conflicting needs when documenting a RA.

Architectural views and viewpoints: an architecture view expresses the architecture of a system from the perspective of specific system concern. An architecture viewpoint establishes the conventions for the construction, interpretation, and use of architecture views to frame specific system concerns [8]. In our analysis, we selected the *views* that were proposed or used in studies, or only cited by them throughout the paper. We identified 24 views: functional logical view (S4, S5, S11, S17, and S21), process view (S5, S9, S10, and S17), components view (studies S5, S10, and S15), implementation view (studies S9, S10, and S17), scenario view (studies S5, S9, and S17), platform view (studies S5, and S10), technical view (S11), physical view (S5), view (S11), deployment view (S4), model view (S9), context view (S5), informal view (S13), information models view (S10), domain view (S5), New view, interface view (S5), code view (S5), module view (S5), execution view (S5), conceptual view (S5), execution view (S5), new view (S5), filtered view (S5), and augmented view (S5).

Adherence of the approaches with the standard ISO/IEC 42010

The RQ_2 is related to analyze the adherence level of the approaches with the architectural description standard ISO/IEC 42010 [8]. To answer RQ_2, we analyzed the studies to know whether they addressed the proposed elements by the standard (i.e., *views*, *viewpoint*, *model kind*, *stakeholder*, *concern*, *architectural decisions*, *rationale*, and *ADL*). For each element, we consider the level of details, that are: N (no details), L (low level of details), M (medium level of details), and H (high level of details), that were addressed by each approach. A summary of this analysis is shown in Table 1.

In general, the studies considered only a few set of elements (i.e., views, viewpoints, and model kind) proposed by the standard ISO/IEC 42010. It is important to note that, 42% of the approaches (S1, S6, S10, S11, S9, S12, S15, S18, and S20) were described in a high-level abstraction. Regarding the adherence with the standard, only two studies (S11 and S18) followed the recommendations proposed by this standard. Nevertheless, these studies did not provide details on how to establish the artifacts that are part of the architectural description. A discussion for each element is presented as follows.

We identified some viewpoints, and selected the ones that were proposed, used or cited by the authors throughout the paper. We found 25 *viewpoints* to represent and describe RAs. In S3, five viewpoints that were used to represent the RAs: use-case viewpoint, logical viewpoint, deployment viewpoint, implementation viewpoint, and data viewpoint. S5 presents a set of viewpoints, as build-time viewpoint, behavioral viewpoint, execution architecture viewpoint, code architecture viewpoint, module architecture viewpoint, conceptual architecture viewpoint, and feature viewpoint. S7 used three viewpoints, that are: physical viewpoint, scenario viewpoint, decision viewpoint. S10 proposed a process that established five system's perspectives, such as enterprise viewpoint, information viewpoint, computational viewpoint, engineering viewpoint, and technology viewpoint. S16 used four viewpoints: decision detail viewpoint, decision relationship viewpoint, decision chronology viewpoint, and decision stakeholder viewpoint. Finally, S19 proposed a viewpoint to represent the variability in RA. S5 presents a set of viewpoints, as build-time viewpoint, behavioral viewpoint, execution architecture viewpoint, code architecture viewpoint, module architecture viewpoint, conceptual architecture viewpoint, and feature viewpoint. S7 used three viewpoints, that are: physical viewpoint, scenario viewpoint, decision viewpoint. S10 proposed a process that established five system's perspectives, as enterprise viewpoint, information viewpoint, computational viewpoint, engineering viewpoint, and technology viewpoint. S16 used four viewpoints: decision detail viewpoint, decision relationship viewpoint, decision chronology viewpoint, and decision stakeholder. Finally, S19 proposed a viewpoint to represent the variability in RA.

We also identified *models* to model and represent RAs. An architecture model can be a part of more than one architecture view [8]. In particular, S16 used the activity diagram, requirements diagram, parametric diagram, state machine diagram, and use case diagram to represent viewpoints and show the flow of activities for architectural decision-making. S3, S4, S5, S7, and S9 used UML diagrams to represent views and viewpoints, that are: use-case diagram, component diagram, activity diagram, package diagram, workflow diagrams, sequence diagram, state diagram, collaboration diagram, class diagram, and message sequence charts. S6 used the feature model to manage the variability and product derivation, while S19 used the internal block diagram (SysML) to model the variability viewpoint. Finally, S20 used the use case diagram to represent the stakeholders and model the RA. Finally, the studies S8 used the π-ADL to model the HLA architectural style in a formal way, while S2 proposed the Rapide ADL that is a formal language to simulate the behavior of system architectures.

We identified several stakeholders that could be considered for the description of RA. It can be an individual, team, or organization that have an interest in a system [8]. The stakeholders that identified were: architects, system designers, attribute leaders, component designers, project managers, suppliers, developers, domain experts, business-persons, customers, system users, and engineers. The most common stakeholders of these approaches are architects, project managers, and developers. In general, the studies only presented who are the important stakeholders for the architecture description activity, and did not show how to represent and manage these stakeholders. We also found a description of the rationale records explanation, justification or reasoning about architecture decisions that have been made. The rationale for a decision can include the basis for a decision, alternatives, and trade-offs considered, potential consequences of the decision. In this context, we analyzed the architecture decisions on RAs. Only 14% of studies addressed this element. Specifically, S7 represented the architectural decisions in a semi-formal way using architectural patterns, and a meta-model to represent them, while S16, S17 represented these decisions an informal way through text description. S7 addressed the rationale for architectural decision-making through a table using a textual description with the reasons to choose an alternative, while S16 used a meta-model and the textual description to show which are the reasons for choosing an architectural solution.

Applicability of approaches for describing RAs

The RQ_3 aims to analyze how much the approaches are easy to use for the development of RAs description. For this, we analyzed the level of the details of the approaches, the tools suggested by these approaches to facilitating the building of the description of RAs, and the guidelines available to help on the use of these approaches. We classified the level of description of the approaches into four categories: *N* (no details), *L* (low level of details), *M* (medium level of details), and *H* (high level of details), as shown in Table 1. In general, the approaches are described with a high-level of abstraction and they do not present details on how to use the approaches to create an architectural description, and neither steps/tasks that are necessary to the building of such descriptions.

The studies S1, S2, S6, S9, S10, S11, S12, S15, S18, and S20 were classified in the category *L* (low level of details) because these studies proposed generic activities/tasks for the building of description of RAs, without showing *what* and *how* to build the artifacts that are part of an architectural description. In category *M* (medium level of details) the studies S2, S5, S7, S8, S14, S16, S19, S21 were classified because they presented with more details, the views, viewpoints, and models that could be used to build a description of RAs, S4, and S17 were classified as *H* (high level of details), since they offered more information about the views, viewpoints, models, architectural patterns, and examples that could support RAs description. None approaches providing guidelines to facilitate their application were identified, neither we do not find extra material to support their use. We also analyzed whether the approaches suggested tools to facilitate their application in practice. Only S18 suggested HyperFlex toolchain an open-source tool for supporting the design and the reuse of RAs. Therefore, this area still needs to be better explored in a further research tool.

Maturity of the approaches for describing RA

The RQ_4 aims to analyze what is the approaches level of maturity. For this, we used the TRL (*Technological Readiness Level*), which is a type of measurement used to assess the maturity level of a particular technology [14]. There are nine technology readiness levels.

TRL 1 is the lowest maturity level and TRL 9 is the highest one. S1, S6, S9, S10, S11, and S12 were classified as TRL 2. At this level, the approach is in the beginning. Only basic principles are observed the applications are speculative, and there may be no proof or detailed analysis to support the assumptions. S2, S13, S14, S16, and S18 were classified as TRL 3 since they include analytical and laboratory studies to physically validate the analytical predictions of separate elements of the technology. We classified S3, S7, S8, S15, S19, S20, and 21 as TRL 4 because their validation was carried out in the laboratory, and basic technological components were integrated to establish that they will work together, S4 and S17 were classified as TRL 5, because their basic technological components were integrated for testing in a simulated environment. Finally, only S5 was classified as TRL 6, because at this level a model or prototype was developed, representing a nearly desired configuration. Activities include testing in a simulated operational environment or laboratory. In this paper, we considered the studies that had TRL greater or equal than level 4 as a good level of maturity. In this sense, we observed that all the studies that were proposed in the industrial context obtained a good level of maturity, and the most of them (S3, S4, S5, S7, S8, and S19) were proposed for general purpose, as shown in Table 1. It is worth highlighting that no studies were classified with TRL greater than level 6.

Interoperability and Dynamism

The RQ_5 aims to analyze how the approaches have addressed interoperability and dynamism concerns in the description of RAs. We analyzed the level of interoperability (i.e., technical, syntactic, semantic, and organizational) that the approaches considered in the architectural description, and the strategies used to provide the interoperability. In short, few studies addressed the interoperability indeed, since most of them only commented on the text that RAs help to achieve the interoperability in systems, but no evidence to support this was found. In S1, the authors suggested that interoperability among different systems can be achieved through the use of protocols. The only study with concrete action about interoperability is S4, the authors adopted International Learning Technology standards, to achieve interoperability among different systems, which is currently a major issue in the domain of LMS (Learning Management System).

Regarding to dynamism we investigate whether the types of dynamism (i.e., description programmed, self-manageable, and ad-hoc) were considered by approaches and the strategies used to represent the dynamism in such architectures. In this sense, only S8 considered the dynamism in the description of RA proposing a formal model for HLA architectural style using the π-ADL, which is aimed at the description of architecture dynamic evolution. The type of dynamism considered in this study was programmed.

4 Discussion of results

This study aimed to present an overview of how to RAs have been described and represented, showing the main existing approaches for building the description of such architectures. However, we did not find approaches that fully adhered to the ISO/IEC 42010 standard. In this sense, we identified research gaps that could be explored in future work.

The ISO/IEC 42010 specifies how architecture descriptions of systems are organized and expressed, presenting the architecture viewpoints, architecture frameworks, and architectures languages for use in architecture description. In this sense, **adherence to standards** could be considered as a research gap in RAs description. As shown in Table 1, few approaches adhered with this standard as a base for supporting the building the architecture descriptions. The few studies that considered the elements proposed by ISO/IEC 42010, described and represented these elements with few details, and with a high level of abstraction. This scenario makes difficult the building of RAs descriptions of systems, besides difficulties in the standardization of RAs. Therefore, it is interesting to consider the standard ISO/IEC 42010 to propose approaches for RAs description because it is the well-known standard used for representation and documenting software architectures, and we believe RAs could benefit from this. The approaches identified in this work were proposed superficially without giving details on what tasks/activities are needed for building a suitable architecture description. The activities proposed indicate only that something must be done, but they do not indicate how this should be done. In this sense, there is a need to improve or propose new approaches to describe RAs that should be described in detail way, showing for architects what tasks/activities should be performed.

Guidelines and tools also considered as a research gap. When approaches are made available to the scientific community, different stakeholders may use them to describe RAs. It is important to highlight that when these approaches are used for the building of architecture descriptions without support materials (as usage guidelines) this may lead to erroneous interpretations and consequently affect the quality of description of RAs because such guidelines may help understanding better how to use these approaches. Tools can automate some tasks/activities to describe RA, as well as, made easy the representation of such architectures. Despite the importance of the guidelines and tools, only one study proposed a support tool and no approach provided supporting guidelines. In this sense, we note that there is a lack of approaches that provide guidelines and tools to support the description of RAs suitably.

Characteristics of complex systems have been calling the attention of researchers in recent years, since the size and complexity of modern software systems. It is expected that new RAs are proposed for these systems in a short time. Such architectures should consider different characteristics that concern to complex systems, such as **interoperability** and **dynamism**. From the result of this paper, we observed that few approaches address the interoperability and dynamism in current systems. Therefore, we expect that new initiatives will be proposed to support the dynamism and interoperability description in RAs.

We analyzed the validation of the approaches using the TRL. Most studies were classified with a low maturity level because few approaches have been evaluated using real case studies. Therefore, **validation of approaches** should be conducted to evaluate the approaches to describe RAs to apply these approaches in the industry in real situations so that will be analyzed the effectiveness of the approaches to support the architecture descriptions. Regarding the identified studies in partnership with the industry, we found only four approaches that were proposed and/or validated by the industry. We believe that studies with **contribution for the industry** should be conducted since our analysis shows a lack of studies that investigate how the industry has described and represented RAs. Therefore, the academy may be performing research that does not meet the real needs of the software industry. It is also necessary to understand how the software industry represents currently such architectures.

5 Threats to Validity

In order to minimize biases of our study, we discuss some actions that we performed to mitigate threats to construct, internal, external, and conclusion validity [15, 16]:

Construct validity: we applied a mapping study protocol with relevant details (i.e., RA description concepts, research questions, search method, study selection, data extraction, analysis), which was reviewed by researchers. This helped mitigating an imprecise description of the mapping study setting.

Internal validity: a threat to internal validity comes from study selection bias. We piloted the inclusion/exclusion criteria with three researchers to mitigate this threat. This helped to clarify differences and build a common understanding of the inclusion/exclusion criteria. Additionally, we performed researcher triangulation in order to minimize researcher bias when selecting the primary studies.

External validity: the amount and relevance of included studies may be considered as a threat to validity to the generalization of the results.

Conclusion validity: during data extraction, not all the information were obvious to answer the research questions and some data was interpreted (e.g., some elements proposed by ISO/IEC 42010). Therefore, in the event of a disagreement among authors, discussions among the participants were conducted.

6 Conclusion and future work

The current systems have become increasingly complex and larger. Consequently, developing and maintaining such systems have become a critical issue. In response to this scenario, RAs can provide a set of knowledge and experience about how to architect a set of software systems to a specific domain. In this perspective, this secondary study aimed to evaluate the existing approaches to describe RAs, analyzing their adherence to ISO/IEC 42010, applicability, maturity, and checking how the interoperability and dynamism have been addressed in the description of RAs. For this analysis, we identified and selected 21 approaches to describe RAs.

As main results, we identified works to describe RAs. Two approaches (S4, S17) were classified as TRL 5, adhering three elements of the ISO/IEC 42010 (view, model kind, and stakeholders). One approach (S5) was classified as TRL 6, also addressing three elements of the ISO/IEC 42010 (view, viewpoint, and model kind). These approaches were considered the most complete to describe RAs. Other approaches also could contribute to describe RAs as S7, S16 that adhere to at least half the ISO/IEC standard, S3, S8, and S19 achieve a medium TRL, and S8 that considered the dynamism in the description of RA.

As future work, we intend to establish a method for describing RAs considering the elements proposed by standard ISO/IEC 42010, and the concerns of modern software systems. This method will be composed of guides and tools to support the description of such architectures.

Acknowledgment

This work was supported in part by CNPq (Grant: 141602/2017-1, 312634/2018-8), FAPESP (Grant: 2018/07437-9, 2018/24173-5, 2017/22107-2, 2017/06195-9), and CoSmart Spanish project (contract TIN2017-83964-R).

References

1. Garcés, L., Oquendo, F., Nakagawa, E.Y.: Software mediators as first-class entities of systems-of-systems software architectures. Journal of the Brazilian Computer Society **25** (2019) 8
2. Bass, L.: Software architecture in practice. 3rd edn. Addison-Wesley (2013)
3. Angelov, S., Grefen, P., Greefhorst, D.: A classification of software reference architectures: Analyzing their success and effectiveness. In: WICSA/ECSA Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, Cambridge, UK, IEEE (2009) 141–150
4. Martinez-Fernandez, S., Ayala, C.P., Franch, X., Marques, H.M.: Benefits and drawbacks of software reference architectures: A case study. Information and Software Technology **88** (2017) 37 – 52
5. Galster, M., Angelov, S., Martínez-Fernández, S., Tofan, D.: Reference architectures and scrum: Friends or foes? In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017), Paderborn, Germany, ACM (2017) 896–901
6. Martínez-Fernández, S., Ayala, C.P., Franch, X., Martins Marques, H., Ameller, D.: Towards Guidelines for Building a Business Case and Gathering Evidence of Software Reference Architectures in Industry. Journal of Software Engineering Research and Development (JSERD) **2** (2014)
7. Nakagawa, E.Y., Oquendo, F., Avgeriou, P., Cuesta, C.E., Drira, K., Maldonado, J.C., Zisman, A.: Foreword: towards reference architectures for systems-of-systems. In: International Workshop on Software Engineering for Systems-of-Systems (SESoS), Florence, Italy, IEEE Press (2015) 1–4
8. ISO, I.: Iso/iec/ieee 42010: Systems and software engineering–architecture description. Technical report (2011)
9. Guessi, M., Nakagawa, E.Y., Oquendo, F., Maldonado, J.C.: Architectural description of embedded systems: A systematic review. In: Proceedings of the 3rd international ACM SIGSOFT symposium on Architecting Critical Systems, ACM (2012) 31–40
10. Guessi, M., Neto, V.V.G., Bianchi, T., Felizardo, K.R., Oquendo, F., Nakagawa, E.Y.: A systematic literature review on the description of software architectures for systems of systems. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, ACM (2015)
11. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology **64** (2015) 1 – 18
12. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - a systematic literature review. Information and Software Technology **51** (2009)
13. Guessi, M., Oquendo, F., Nakagawa, E.Y.: Variability viewpoint to describe reference architectures. In: Proceedings of the Working IEEE/IFIP Conference on Software Architecture, Sydney, Australia, ACM (2014)
14. Mankins, J.C.: Technology readiness levels. White Paper, April **6** (1995) 1995
15. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)
16. Zhou, X., Jin, Y., Zhang, H., Li, S., Huang, X.: A map of threats to validity of systematic literature reviews in software engineering. In: 23rd Asia-Pacific Software Engineering Conference (APSEC). (2016) 153–160