Aggregating Empirical Evidence about the Benefits and Drawbacks of Software Reference Architectures

Silverio Martínez-Fernández
UPC - BarcelonaTech
Barcelona, Spain
smartinez@essi.upc.edu

Paulo Sérgio Medeiros dos Santos
PESC/COPPE/UFRJ
Rio de Janeiro, Brazil
pasemes@cos.ufrj.br

Claudia P. Ayala, Xavier Franch
UPC - BarcelonaTech
Barcelona, Spain
cayala,franch@essi.upc.edu

Guilherme Horta Travassos
PESC/COPPE/UFRJ
Rio de Janeiro, Brazil
ght@cos.ufrj.br

***Abstract*—Context: Several empirical studies investigated the benefits and drawbacks of acquiring a Software Reference Architecture (SRA) to construct a family of software systems with similar architectural needs. However, these empirical results have not been synthesized by any study yet. Such synthesized evidence is essential to make informed decisions whether or not to adopt an SRA in an organization. Goal: To aggregate existing empirically-grounded evidence about the benefits and drawbacks of SRAs, aiming at supporting organizations' decision making on their adoption. Method: To identify primary studies in the technical literature through a systematic literature review, and then, use the Structured Synthesis Method (SSM) to aggregate qualitative and quantitative evidence through the use of diagrammatic models. Results: From the five identified primary studies, five SRA benefits have considerably increased their belief value after aggregation: interoperability of software systems, reduced development costs, improved communication among stakeholders, reduced risk, and reduced time-to-market. Also, one drawback of SRAs has increased its belief value: the required learning curve for developers. Conclusions: The aggregated results consolidate knowledge and confidence on some of the studied SRA effects. The commonly reported effects showed a clear increment of their belief and pointed out to broader generalization. The effects that did not show any belief increment are important to detect areas requiring further evidence to reach a higher degree of consolidation. Practitioners might benefit from these results to support the decision of adopting an SRA in practice.**

Keywords—software engineering; software architecture; software reference architecture; evidence representation; evidence aggregation; belief functions; Dempster-Shafer theory; research synthesis; secondary study; meta-analysis

I. INTRODUCTION

Many today's organizations face the development and maintenance of big and complex software families, which are composed of many similar software systems. In this context, organizations may introduce a central asset called Software Reference Architecture (SRA) capturing the architecture essence of similar software systems in an application or technology domain [1]. To make informed decisions whether or not to adopt an SRA, an organization needs to analyze the benefits and drawbacks for its own context.

Previous empirical studies have reported the strengths and weaknesses of SRAs in order to ease their industrial uptake [1]–[5]. However, the results of these single empirical studies have not been analyzed together. Therefore, there is not a consolidated and unified evidence about the benefits and drawbacks of SRA adoption.

The main goal of this paper is to strengthen the evidence about the benefits and drawbacks of SRAs by means of performing a research synthesis study of existing evidence through the Structured Synthesis Method (SSM) [6]. The SSM consists of a method able to aggregate qualitative and quantitative evidence through the use of diagrammatic models. Such synthesis helps to gain more confidence on the effects of SRAs that have been reported by more than one empirical study, and to determine the context of those effects that only appear under specific contexts of SRAs. The aggregated evidence of this paper aims to help practitioners to understand and analyze the benefits and drawbacks of adopting and using SRAs in their organizations, and to support researchers to identify areas where further research is needed to consolidate/understand the actual evidence. Besides the aforementioned goal, this study presents a working example of application of the SSM, and discusses its applicability for synthesizing Software Engineering (SE) evidence.

This document is structured as follows. Section II gives a background on SRAs. Section III presents the research methodology of this study: the SSM. Section IV represents evidence that was reported in previous single studies. Section V aggregates the empirical evidence modeled in Section IV. Section VI discusses the results of this study. Section VII discusses the threats to validity of this study. Finally, Section VIII concludes the paper and presents future work.

II. BACKGROUND

Nakagawa et al. define an SRA as "an architecture that encompasses the knowledge about how to design concrete architectures of systems of a given application domain; therefore, it must address the business rules, architectural styles (sometimes also defined as architectural patterns that address quality attributes in the SRA), best practices of software development (for instance, architectural decisions, domain constraints, legislation, and standards), and the software elements that support development of systems for that domain. All of this must be supported by a unified, unambiguous, and widely understood domain terminology" [7]. SRAs mainly appear in organizations where the multiplicity of similar software systems (i.e., systems developed at multiple locations,

by multiple vendors, and across multiple organizations) triggers a need for life-cycle support for all systems [2]. Therefore, SRAs are attractive when organizations become large and distributed in order to develop new software systems or new versions of existing ones. In this context, organizations need to analyze whether or not to acquire an SRA.

Angelov et al. distinguish between different types of SRAs [8]. These different types are classified by the following characteristics:

- Goal: to *standardize* concrete architectures of software systems (aiming at system/component interoperability) or to *facilitate* the design of concrete architectures (aiming at providing guidelines/inspiration for the design of systems).
- Organizations in which the SRA is used: an SRA can be used in a *single organization*, or in *multiple organizations* that share a certain property (e.g., car manufactures).
- Definition type: a *preliminary* SRA is defined when the technology, software solutions, or algorithms demanded for its application do not yet exist in practice whereas a *classical* SRA is defined when these artifacts exist by the time of its design and have been tested in practice.

Examples of well-known SRAs are: AUTOSAR, which is a classical SRA that aims to standardize the software architecture of electronic control units in modern vehicles, and targets multiple organizations (e.g., many car manufactures) [5]; classical service-oriented SRAs, such the one studied in [1] that facilitates the design of variability-intensive service-based applications, and targets many municipalities in the Netherlands using e-government systems.

III. Methodology

We defined our research question as: "*What are the trends on available empirical evidence about the benefits and drawbacks of SRAs for acquisition organizations?*" We focused on the benefits and drawbacks for organizations that introduce an SRA for designing and constructing a family of software systems. Therefore, we focused on the SRA "usage" perspective, rather than other perspectives, e.g., SRA "design".

To answer this research question, we needed to aggregate the research results of previous works. To perform such aggregation, we used the SSM [6]. As both qualitative and quantitative research synthesis method, the SSM briefly depicts the important contextual aspects, and informs the trend of the effects (e.g., positive or negative), as well as a certain estimation about them. Therefore, SSM neither aggregates precise quantitative findings nor rich qualitative descriptions.

We decided to use the SSM because it is able to conceptualize about the context, and to integrate studies' results. Therefore, it has an interesting blend of integrative and interpretive synthesis [9]. In the SSM, interpretative synthesis aspects are concerned with the organization and development of concepts to describe contextual aspects of evidence whereas integrative features are focused on pooling data about *cause-effect* or *moderation* relations. Moreover, studies about SRA benefits and drawbacks report both qualitative and quantitative evidence. The SSM can aggregate these types of evidence, and takes into account the uncertainty estimated for each evidence. Besides, the SSM offers tool support to model and synthesize evidence [10], including facilities for graphical modeling, evidence search, and support for the synthesis. Another important functionality is the evidence model comparison used to aggregate evidence, which has mechanisms for "conflict resolution" between the models. The tool and all the results of the synthesis presented in this paper can be accessed at: http://evidencefactory.lens-ese.cos.ufrj.br/.

The SSM is composed of three main phases. First, papers are identified and selected according to predefined criteria. Existing approaches for study selection (e.g., systematic review search process) can be used for this purpose. Second, information from each paper is extracted, and the pieces of data are organized and put into the same perspective. Evidence are modeled with a diagrammatic representation, which describes the context, *cause-effect* relationships, and their *moderators*. Each model can have more than one *cause-effect*, so multiple outcomes/effects can be analyzed together in the same aggregation. Third, with all evidence under the same perspective, the last phase is dedicated to consolidate and synthesize the results. This synthesis shows what the main trends or conflicts among the analyzed evidence are. The primary interest of the SSM is to combine *cause-effect* relationships from many SE empirical studies.

Next, we report how we applied the SSM method to synthesize the research on SRA benefits and drawbacks.

A. Step 1 of the SSM: Selecting Primary Studies

To select the primary studies, we defined a systematic search strategy, and the inclusion and exclusion criteria.

1) Search strategy: For the search strategy, we considered the same data sources and search string as in a systematic review about SRA engineering that has been conducted in conjunction with the LabES-USP group, whose protocol is available at www.essi.upc.edu/~smartinez/ProESEM15.pdf. Therefore, the following electronic databases were used: Scopus (scopus.com), Web of Science (isiknowledge.com), IEEE Xplore (ieee.org/web/publications/xplore), ACM Digital Library (dl.acm.org), ScienceDirect (sciencedirect.com), and Springer (link.springer.com).

To conduct the search, we used the following search string, which aims to find scientific studies of SRAs for the concrete architecture of software systems:

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

The search was conducted using filters on the titles, abstracts, and keywords of the studies. Besides searching and collecting the papers, we used the forward and backward snowballing strategy for the included papers. Experts or reviewers suggestions were also accepted, which is essential for studies that are not indexed or published yet (i.e., in press).

2) Inclusion and exclusion criteria: The SSM is flexible regarding the type of data collected in studies and the amount of their outcomes. So, as inclusion criteria, we defined any

empirical study reporting findings based on evidence about the benefits and drawbacks of adopting an SRA.

Concerning exclusion, we defined three exclusion criteria: (i) studies whose findings were based on opinions rather than evidence; (ii) studies that were not the primary source of the reported study or data (i.e., papers that reanalyze or review a published study or do not consider data); and (iii) studies that were not reported in English.

3) Study selection: The search string, performed in September 2014, retrieved 492 non-duplicated studies. From these studies, the empirical studies reporting evidence were manually identified. From this list, we looked for those focusing on reporting the benefits and drawbacks of SRAs.

Three papers that report empirically grounded results about SRA benefits and drawbacks were found [2]–[4]. Searching through the references and citations of these three papers, [1] was added to the included studies. Also, [5] was included by convenience, as it was not published yet, but we were aware of its existence because it was conducted by three of the authors.

Finally, we ended up with five included primary studies reporting evidence on the benefits and drawbacks of using an SRA in an organization. The most important details of each of the five papers can be found in Table I.

B. Step 2 of the SSM: Evidence Representation

The SSM uses a diagrammatic representation to support the aggregation of evidence. Following the understanding of most research synthesis methods [9], the idea is that once all evidence are put under the same format their combinability can be better analyzed, and the decision for combination more objective. The representation used in SSM is called *theoretical structure* and, as the name suggests, it is based in the notion of theories, from which the SSM stems its capability of accommodating most diverse types of evidence.

The ten semantic constructs used in the representation are shown in Fig. 1. There are three possible types of *structural* relationships in the representation: *is a*, *part of* and *property of*. All of them have counterparts in UML, respectively: generalization, composition and class attributes. The *is a* and *part of* relationships use the same UML notation for generalization and composition. *Properties* are denoted by dashed connections. The relationships are used to link two types of concepts – *value* and *variable*.

A *value* concept represents a particular variable value, usually an independent variable. Value concepts are represented by rectangles, and they are classified in *archetypes* (the root of each hierarchy), *causes* (indicated by the use of a bold font and a 'C1' following the name denoting that it is the 'cause 1', (e.g., 'Reference Architecture') and *contextual aspects* (e.g., 'Enterprise Software'). The four archetypes – activity, actor, system, and technology – were suggested by Sjøberg et al. [11] in an attempt to capture the typical scenario in SE described by an actor applying a technology to perform activities in a software system.

A *variable* concept focuses on value variations usually associated with a dependent variable. *Variable* concepts are represented by ellipses or parallelograms symbolizing *effects* (e.g., 'communication') and *moderators* (e.g., 'maturity'), respectively. In addition, *effects* are not connected to *cause* using lines as they are assumed to exist when reading the diagram. Lines are also lacking in the link between *moderators* and the (moderated) *effects*. In this case, a textual hint (e.g., 'M1') is shown besides both the moderated effect and moderator. Both relationships, *cause-effects* and *moderation*, are denominated *influence* relationships.

To indicate the effect size, a seven-point Likert scale is used. The scale ranges from strongly negative to strongly positive and is indicated above the ellipse (e.g., '⩓' indicates that 'Flexibility for Suppliers' is positively affected by 'Reference Architecture'). The other type of *variable* concepts, namely *moderators*, indicates that some positive or negative effect is moderated (i.e., reduced) when it increases or decreases. For instance, a moderator is how a 'knowledge repository' influences 'communication'. A last aspect related to *variable* concepts is the association of a *belief value* (ranging from 0% to 100%) to estimate the confidence in the observed effects and moderations. The bar under each element represents the belief value, e.g., 'interoperability' has 35% belief value.

TABLE I. PRIMARY STUDIES.

Study Id.	Study Type: Instruments	Participants	SRA Application Domain	SRA goal[a]	SRA used in[a]	SRA type[a]	Belief[b] & evidence type	Year
S1 [2]	Expert meeting: presentations, discussions	Architects from the System Architecture Forum	Defense and commercial equipment	Standard. & Facilitation	Single & multiple Organizations	Preliminary & classical	0.25+0.10=0.35 qualitative	2010
S2 [3]	Case study: interviews, questionnaires, docs.	28 sw. architects and developers from IT consulting	Banks, insurers, public administration, utilities, and industries	Standard. & Facilitation	Single Organizations	Classical	0.25+0.19=0.44 qualitative & quantitative	2013
S3 [4]	Survey: questionnaires	90 sw. architects and developers from worldwide	n/a	Standard. & Facilitation	Single & multiple Organizations	Preliminary & classical	0.25+0.15=0.40 qualitative & quantitative	2013
S4 [1]	Case study: interviews, docs., meetings	20 sw. architects, managers and experts from local e-goverment	Variability-intensive service-oriented systems	Facilitation	Multiple Organizations	Classical	0.25+0.15=0.40 qualitative	2013
S5 [5]	Survey: questionnaires	51 practitioners from AUTOSAR partners	Automotive systems	Standard.	Multiple Organizations	Classical	0.25+0.17=0.42 qualitative & quantitative	2015

[a] To check the possible values for SRA goal, used in, and type, see Section II. [b] The belief value is calculated as shown at the end of this section.

1) Extracting data to model evidence: The data extraction and evidence modeling activities are intertwined and, together, are very similar to the text coding and analysis process [12]. The major orientation in creating the theoretical structures comes from the *thematic synthesis* and its increasing abstraction level, where text is translated into codes, which are translated into concepts and relations, and, from them, the *theoretical structure* representing an evidence is modeled. The SSM also contains recommendations from *meta-ethnography*, such as how the text should be coded, and papers translated into one to another to identify concepts and relations. The inductive approach from *qualitative comparative analysis*, where concepts are identified inductively from the collection of studies, complements these recommendations. To improve the synthesis reliability, the participation of more than one researcher is recommended as is in *case survey* and many other qualitative methods. A resume of all these research synthesis methods can be obtained in [9]. At last, instructions for identifying *cause-effect* relationships are also included, since they put qualitative and quantitative evidence in the same perspective in the SSM. The instructions are based on the differentiation from [13]: qualitative research "explains individual cases; using the causes-of-effects approach" whereas quantitative research "estimates average effects of independent variables; using the effects-of-causes approach".

Although these heuristics for evidence modeling are used to make the process more systematic and transparent, evidence modeling in SSM is still a subjective process with some influence of the researcher abstraction skills, and knowledge about the topic of interest. Nevertheless, all these orientations were used to model the evidence related to the five studies identified. The first and the second authors divided the five papers into two sets and then individually modeled each evidence. After that, we reviewed the models created by each other, including several meetings to discuss if we have a common understanding about the models. The remaining authors performed a final revision of the models, and the resulting aggregated model.

It is interesting to notice that the identification of concepts and relationships is an iterative process, and the modeling of evidence can be a trigger to review the others. This is important to make concepts and evidence structures more consistent, and particularly important for evidence synthesis, which is described in the next subsection. A last step of data extraction is the evidence quality assessment, which is used for estimating a *belief value* used in the synthesis (also described in next subsection).

C. Step 3 of the SSM: Evidence Synthesis

To aggregate evidence, it is necessary to determine evidence combinability. For that, all *value* concepts (*archetypes*, *cause* and *contextual aspects*) and *structural* relationships (*is a*, *part of* and *property of*) between the different models must match. For instance, if an evidence model describe that 'Enterprise Software' is *type of* 'System' as part of the context, then the other evidence model should have a the same relationship as part of its context description. If there is not a direct correspondence with the use of the same concepts (in the example, 'Enterprise Software'), the researcher can decide if their meaning are similar enough for the aggregation purposes, and still aggregate the evidence. The other option is to keep both concepts separated in the aggregated results – that is, the effects associated with 'Enterprise Software' in an evidence model are not aggregated with the effects associated with the system described in the other model. A third option, when an aspect is only present in an evidence, is to add or remove the concept from the aggregation. Then, the resulting aggregated evidence *value* concepts and *structural* relationships are defined. Next, it is necessary to aggregate *influence* relationships.

After determining which evidence can be combined, and grouping the ones that can, uncertainty formalism is necessary to combine the results – otherwise, a simple vote counting strategy would be used. In the SSM, the Mathematical Theory of Evidence [14] (also known as Dempster-Shafer theory, DST) is the mathematical formalism that enable to combine results. While *value* concepts are used to determine aggregability, the aggregation itself is focused on the *variable* concepts and their relationships (*cause-effect* and *moderation*). DST uses two main inputs to combine two pieces of evidence. One is the hypotheses believed to have a chance to be true – belief value greater than zero – and the other is the belief values themselves. Hypotheses are defined as sets of the powerset of the defined frame of discernment elements, which in the case of SSM is formed by the values of the seven-point Likert scale: Θ = {SN, NE, WN, IF, WP, PO, SP} – the elements values are abbreviations for the Likert scale terms, e.g., SN is 'strongly negative', IF is 'indifferent', and WP is 'weakly positive'. It is interesting to notice that since hypotheses are sets from the powerset, a hypothesis can be a singleton (e.g., {PO}) or a compound set (e.g., {WN, PO} – meaning an imprecision about weakly positive and positive).

The other input is the belief value assigned to each hypothesis. Belief values are estimated using the study type and a quality assessment. First, based on GRADE evidence hierarchy [15], study type level split the 0-1 belief value range into four subranges: unsystematic observations [0.00, 0.25]; observational studies [0.25, 0.50]; quasi-experiments [0.50, 0.75]; and randomized controlled [0.75, 1]. Second, the quality assessment value is translated to the 0.25 subrange. The SSM method proposes to use two checklists to assess the quality of each study, which are explained in [6]. Based on this, the belief values listed in Table I are calculated, e.g., the study S1 was observational (0.25), and in the quality assessment done by the checklists it got 0.10 out of 0.25. The reader is referred to the tool to check the quality assessment for each study.

Once hypotheses and belief values are defined for each evidence, then the *Dempster's Rule of Combination* is applied, see (1). Equation (1) shows that the aggregated belief value for each hypothesis C is equal to the sum of the product of the hypotheses belief values whose intersection between all hypotheses A_i and B_j of both evidence is C.

$$m_3(C) = \frac{\sum_{\substack{i,j \\ A_i \cap B_j = C}} m_1(A_i) m_2(B_j)}{1 - K}, \text{ where } K = \sum_{\substack{i,j \\ A_i \cap B_j = \emptyset}} m_1(A_i) m_2(B_j) \quad (1)$$

When the intersection between two hypotheses is an empty set, we say that there is a conflict. Conflict is, then, redistributed to the aggregated hypotheses – that is the function of 1 - K in the denominator. More details about how DST is used in SSM can be obtained in [6].

IV. Representation of Software Reference Architecture Effects

In this section, we show how we extracted the evidence from the included studies, and created models to represent such evidence. Evidence modeling has a significant interpretation, coding, reasoning and analysis of components. We translated the evidence from text-based studies into evidence models that are diagrams.

Given the restricted amount of space for detailing each model, we describe only one of them in this document. We choose the model associated with the study S1 (see Fig. 1), since it is the shortest (fewer concepts and relationships), and includes all ten semantic constructs detailed in section III.B. In the study S1, the driving forces for SRAs are elicited from the discussions of the System Architect Forum (architectingforum.org). The authors also present real-world SRAs from different domains to help justifying some of the driving forces elicited. Since the results presented in the study S1 are an outcome of an analysis of discussion between professionals with different background, we decided to use general *value* concept for context description including 'Enterprise Software' and 'Acquisition Organization'. In models of other papers where the context is specific, such as the study S5 describing an SRA for the automotive domain, specific value concepts were used to model it (e.g., 'Automotive Software', and 'AUTOSAR partner').

Apart from the context description, the effects were relatively straightforward to identify as they were listed in the paper's text. For instance, 'Terminology Conventions' concept in S1 was identified in the following excerpt: "*Reference Architecture can also serve as a framework and lexicon of terms and naming conventions, as well as structural relationships within a company, industry or a domain*". In fact, this is usually expected since describing the study results is one of the most important parts of any paper. However, moderators were not so unequivocal, since the authors do not report them as moderators, but rather as particular conditions important to augment some effects. For instance, we can see in Fig. 1 that an SRA *is a* technology. Moreover, the 'C1' notation indicates that SRAs are the *cause* of all the *effects* (represented by *ellipses*) in the model. Also, we can see that SRAs have a positive – strongly positive influence on the 'Development Costs' of a 'Software Project' (35% belief). The M1 notation indicates that such influence has a *moderator* (represented by parallelograms), which is 'Reuse'.

All evidence models created for each paper can be found on the tool at the link previously informed. To make this document self-contained, Table II provides a summary of each evidence model with the list of all effects caused by SRAs. For each study, we give the effect intensity in the Likert scale along with the belief value. For instance, 'Interoperability' was reported by study S1 as positive – strongly positive effect of SRAs with a 35% of belief. Studies S3, S4, and S5 also reported 'Interoperability'.

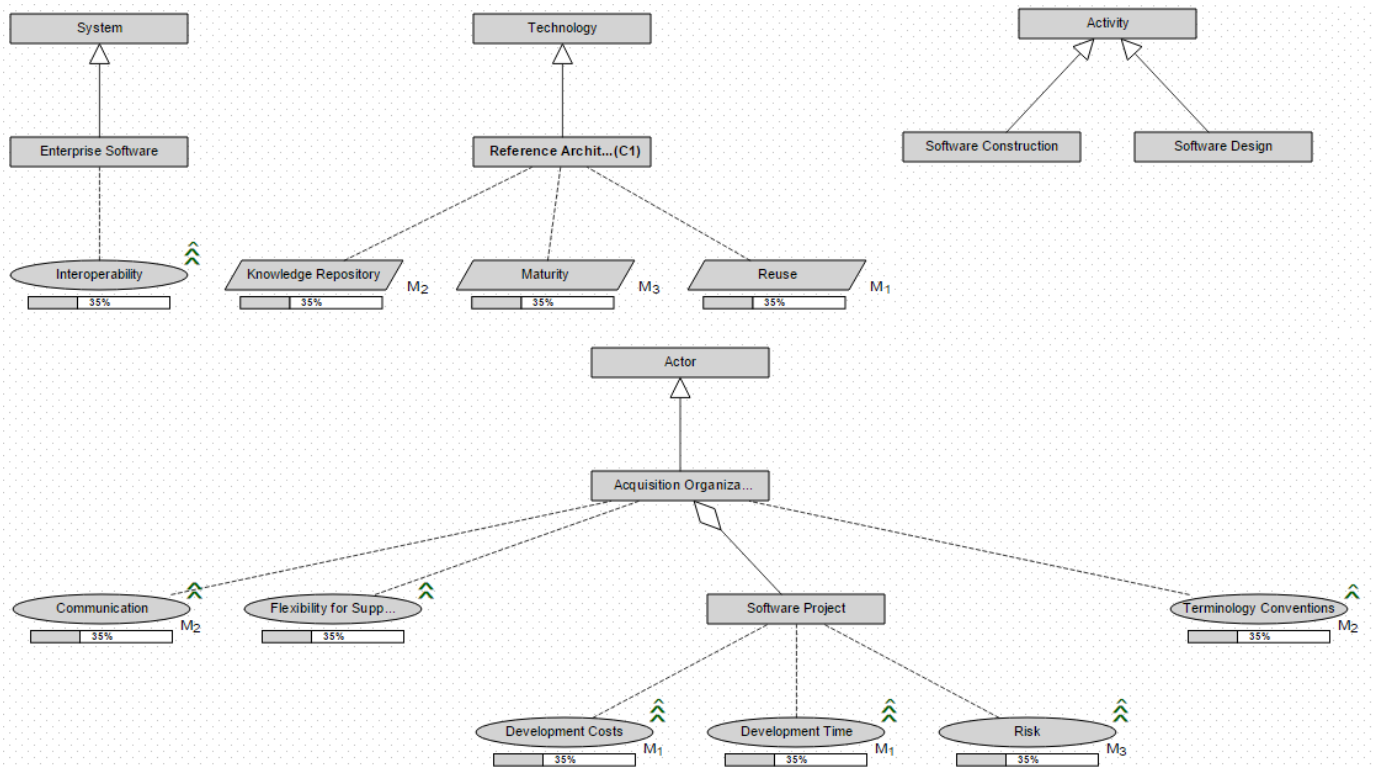

Fig. 1. Evidence model representing the results of the study S1.

TABLE II. SRAS EFFECTS AS REPORTED IN SELECTED STUDIES.

Study / Effect	Representation of evidence from single studies, shown as: intensity (belief value)				
	S1	**S2**	**S3**	**S4**	**S5**
Interoperability	PO, SP (0.35)		PO (0.15)	WP (0.40)	PO, SP (0.22)
Development costs	PO, SP (0.35)	PO (0.36)	PO (0.04)		PO (0.16)
Communication	PO (0.35)		PO (0.09)	PO (0.40)	PO, SP (0.20)
Risk	PO, SP (0.35)			PO (0.40)	PO (0.10)
Best practices			PO (0.31)	PO (0.40)	PO (0.13)
Learning curve		SN, NE (0.36)	NE (0.13)	NE, WN (0.40)	NE (0.22)
Development time	PO, SP (0.35)		PO (0.14)		PO (0.14)
Maintenance cost		PO (0.35)			PO (0.14)
Productivity		PO, SP (0.30)			PO (0.11)
Ease of developing		PO (0.30)	PO (0.07)		WP, PO (0.03)
Alignment		WP, PO (0.19)			WP (0.07)
Restriction		NE (0.13)	NE (0.06)		NE, WN (0.07)
Standardization		WP, PO (0.14)	PO (0.16)	WP (0.40)	SP (0.37)
Latest technologies		WP (0.30)			
Investment					NE (0.25)
Reliability		WP, PO (0.14)			
Dependability		SN, NE (0.09)			NE, WN (0.12)
Reputation					WP (0.06)
Software quality			NE (0.06)		WN (0.04)
Novel design solution			PO (0.05)		WP (0.04)
Complexity		WN (0.06)			SN, NE (0.27)
Terminology conventions	WP, PO (0.35)				NE (0.17)
Flexibility of suppliers	PO (0.35)			WN, IF (0.40)	

Regarding the effect intensity, it is interesting to say that we had to interpret it from the textual descriptions. Thus, if the textual description did not qualify the effect with particular adjectives indicating the intensity, then we chose a default value (e.g., PO), rather than a weak (e.g., WP) or strong value (e.g., SP). Similarly, if the paper gave an ambiguous description we defined a lower range for the effect intensity (e.g., WP, PO). For instance, the 'learning curve' drawback in the study S3 is described as "*additional high or medium learning curve for using the SRA features*". The belief values were based on the study type and the quality assessment as described in Section III.C. Yet, in survey papers (which also report quantitative evidence), we weighted the belief values with the number of respondents that actually perceived the effect as result of SRA usage. This weighting was performed using the DST discount operation. The idea of the discount operation is to adjust the mass distribution (i.e., the belief values assigned to the hypotheses) to reflect the source's credibility – a full discount (discount=1) represents a completely unreliable source. For instance, for survey studies, we used the number of respondents as estimation for the discount value calculated as: (1 - number of respondents for the question / total participants). The DST discount operation is also used in studies considering *p-values*.

The evidence modeling process produces interesting results from the aggregation perspective: the individual results provide a basic understanding to combine results. In addition, since results are already translated into diagrams in a more condensed form, they can be practical for other uses.

V. RESULTS OF THE AGGREGATION

Once we individually processed the evidence of the selected studies in the previous section, we proceed to show how we aggregated the results in this section.

Table III shows the results after performing the aggregation of evidence on the benefits and drawbacks of SRAs. The first column shows the reported effect (i.e., benefit or drawback) caused by the introduction of an SRA in the organization. The second column indicates the number of papers that have reported this effect. The third column shows the aggregated intensity about how the SRA causes such effect (e.g., positive or negative). The fourth column represents the aggregated belief on such effect. This is one of the most interesting results of the aggregation. The individual study with the highest belief for an effect was S4, with 40% belief for the 'Interoperability' effect (see Table II). However, after aggregating the results from single empirical studies, some effects caused by SRAs were reinforced. Table III shows in bold those effects that have higher belief of 40% after the aggregation. The fifth column shows whether there was a conflict while aggregating that effect. This is important to analyze and to characterize different contexts from which the evidence was gathered. Lastly, the sixth column shows the difference between the max value of the belief in individual papers and the gained confidence after the aggregation. Therefore, a positive difference indicates the effects that have been reinforced after the aggregation whereas a negative difference shows that the evidence is somewhat contradictory. The effects are ordered by the difference on the belief after the aggregation.

Aggregation was performed using the Dempster's Rule of combination (1). For instance, Maintenance Cost effect was computed in the following manner: $m_{aggregated}(\{PO\}) = m_{s1}(\{PO\}) \times m_{s2}(\{PO\}) + m_{s1}(\{PO\}) \times m_{s2}(\Theta) + m_{s1}(\Theta) \times m_{s2}(\{PO\}) = 0.049 + 0.301 + 0.091 = 0.441$. This is the value found in Table III. It should be noticed that for cases where more than one intensity is involved, the belief function is used (see [6] for details).

Next, we respectively report the effects that: a) increased, b) slightly increased, c) did not change, and d) decreased their belief after the aggregation.

TABLE III. AGGREGATED EFFECTS OF SRAS (ORDERED BY BELIEF STRENGTHENING).

Effect caused by an SRA	Aggregation Results				
	#Papers	*Intensity*	*Belief*	*Conflict*	*Difference*[a]
Interoperability	4	PO, SP	**74%**	-	34%
Development costs	4	PO, SP	**67%**	-	31%
Communication	4	PO	**65%**	-	25%
Risk	3	PO, SP	**65%**	-	25%
Best practices	3	PO	**64%**	-	24%
Learning curve	4	NE, WN	**60%**	-	20%
Development time	3	PO, SP	**52%**	-	17%
Maintenance cost	2	PO	**44%**	-	9%
Productivity	2	PO	38%	-	8%
Ease of developing	3	PO	35%	-	5%
Alignment	2	WP, PO	24%	-	5%
Restriction	3	NE	18%	-	5%
Standardization	4	WP, PO	43%	-	3%
Latest technologies	1	WP	30%	-	0%
Investment	1	NE	25%	-	0%
Reliability	1	WP, PO	14%	-	0%
Dependability	2	NE, WN	12%	-	0%
Reputation	1	WP	6%	-	0%
Software quality	2	NE	6%	-	0%
Novel design solution	2	PO	5%	-	0%
Complexity	2	SN, NE	26%	0.017	-1%
Terminology conventions	2	WP, PO	31%	0.060	-4%
Flexibility of suppliers	2	WN, IF	31%	0.140	-9%

a. The "Difference" column measures the difference among the max value of belief in previous single papers, and the gained confidence after the aggregation.

A. *Effects of SRAs that Increased their Belief*

Seven effects caused by SRAs increased their belief values after the aggregation. These effects have greater confidence value than any effect before aggregation (i.e., greater confidence level than 40%, see Table II), and have been reported by at least three out of the five studies. Next, we enumerate these seven effects and their moderators.

SRAs positively - strongly positively improve the *interoperability* of the software systems (74% belief). Studies reported that SRAs: aim at "*interoperability to improve compliance for a given context*" [S1]; "*act as communication center for information exchange*" [S5]; and integrate software into (and become part of) an SRA [S4]. As we can see in the last example, *existing software* in the organization proportionally moderates interoperability.

SRAs positively - strongly positively impact the *development costs* of software projects (67% belief). *Reuse* of common assets proportionally moderates development costs from not having to start from scratch [S1]-[S2].

SRAs positively improve the *communication* inside their acquisition organizations (65% belief). SRA stakeholders share the same architectural mindset, fostering an improved communication, i.e., "*people talk the same language*" [S5]. *Organizational thinking* proportionally moderates such communication: "*when a service-based SRA is implemented, different departments within an organization need to a) share information with other departments, but also b) get things from other department*" [S4]. Also, the role of an SRA as a *knowledge repository* proportionally moderates knowledge transfer and communication. To sum up, an SRA aids the understanding of architectural and design principles [S1].

SRAs positively - strongly positively influence the *risk* of software projects (65% belief). The *maturity* of an SRA proportionally moderates its risk. Maturity relates to the degree of formality and optimization of processes, from *ad-hoc* practices, to formally defined steps, to managed result metrics, to active optimization of the processes, e.g., "*a mature architecture follows principles for 'good' design, such as high cohesion, high modularity and low coupling*" [S4]. Risk reduction is achieved through the use of proven and partly prequalified architectural elements. The general maturity and experience level associated with an SRA also bears the promise of a higher quality end-product [S1]. If no mature architecture exists, designing and introducing an SRA is likely to fail [S4].

SRAs positively improve the use of *best practices* inside their acquisition organizations (64% belief). The studies do not report the type of best practices. This is proportionally moderated by the *maturity* of an SRA.

SRAs negatively - weakly negatively influence the *learning curve* of developers (60% belief). Developers that use an SRA need to learn its features [S2]. As a consequence, "*many engineers have difficulty learning*" some SRAs [S5]. *Organizational thinking* indirectly proportionally moderates the learning curve: "*changing organizational thinking in employees is often achieved through training that takes place when introducing SRAs*" [S4].

SRAs positively - strongly positively impact the *development time* of software projects (52% belief). This benefit is also proportionally moderated by *reuse*, which can lead to shorter development cycles. However, it is not the same effect as development costs, because it refers to lower time-to-market of the constructed software [S1], [S2].

B. *Effects of SRAs that Slightly Increased their Belief*

Six effects have slightly increased their belief. Three of these effects have been reported by only two studies: reduced *maintenance costs* of software projects, improved *productivity* of developers, and *alignment* of applications to an organization's business needs. Since the studies agree on them, these effects have increased their value, but more research is needed to corroborate them.

However, the other three effects were reported by at least three studies. In the case of *ease of developing* and *standardization*, we can see in Table II that these effects are stronger for some types of SRAs (see Section VI.B). In the case of regulative SRAs that *restrict* the development on software systems, the percentage is low because the three studies reporting it gave a really low confidence value, thus, it seems that it is not seen as a very important drawback for practitioners.

C. Effects of SRAs that Did Not Change their Belief

Seven effects did not change their belief. Three of them, *dependency* of the software systems over the SRA, propagation of bad *software quality* and wrong decisions of the SRA, and *novel design solutions* are reported by two studies, with different degrees of effect intensity, which did not contribute to increase the evidence level during the aggregation. In fact, the two latter effects have a negligible conflict level of 0.002. We can conclude that the confidence level on these three effects is very low, so they rarely appear in practice, and it seems that they are not considered as fundamental benefits or drawbacks.

The use of *latest technologies*, up-front and migration SRA *investment*, *reliability* of SRA software components used in the software systems, and *reputation* of acquisition organizations have been reported by only one study. It does not mean that these effects caused by SRAs are not important, but that more investigation effort by the research community is needed in other to understand them. Still, some of the effects, as reputation, may depend on the SRA type, e.g., an SRA for a market domain so that other companies may be interested in outsourcing [S5].

D. Effects of SRAs that Decreased their Belief

Three effects have lower confidence level after the aggregation due to contradictory evidence in the single studies. These effects are: a) the *complexity* of the software construction process due to using an SRA; b) how an SRA affects the establishment of *terminology conventions*; and c) how an SRA influences the *flexibility of suppliers* or outsourcing companies that develop software systems based on the SRA. We further discuss the reasons why these effects have decreased their belief in Section VI.B.

VI. Discussions

In this section, we respectively discuss the effects that were present in different SRA contexts, contradictory results, and the utility of the aggregation with respect to SRAs theory.

A. Effects of SRAs Present in Different Contexts

The context varies among different studies (see Table I). Still, we have seen common SRA effects reported in different contexts and application domains. This is the case of improved *interoperability*, reduced *development costs*, better *communication*, and higher *learning curve*, which have been reported in four out of five studies without contradictions. These SRA effects, described in Section V.A, are the strongest results of the aggregation.

B. Contradictory SRA Effects from Different Studies

In all studies, the context was the use of SRAs for the design and construction of software systems. However, these SRAs were of different type, for instance, they had different goals (e.g., standardization and facilitation), targeted several domains (e.g., automotive software and e-government), involved different stakeholders (e.g., vendors and client organizations), and coexisted with different software and constraints (e.g., reference models). Due to their different contexts, there are some effects that are caused by some types of SRAs, but are not present in other types of SRAs. Still, even with those differences, we understood that it was possible to generalize the concept of SRA for software systems, independently from their types, in order to analyze its most prominent effects, and then, examine the conflicting results from the perspective of their contextual differences.

Next, we discuss those effects that have contradictory evidence and form hypothesis to contextualize them.

- Are SRAs *complex* or do they *ease the development* of software systems?

The study S5 reported that AUTOSAR (which is an SRA to standardize automotive software) influences negatively - strongly negatively the *complexity* of the software construction process with 27% belief. However, the study S2 showed that nine SRAs for information systems weakly negatively influence this complexity with 6% belief. Therefore, we can see different SRAs that differently affect to the complexity of software development. For these two studies, related effects to complexity had different effect intensity and confidence level. For instance, AUTOSAR have worse results (i.e., lower intensity and confidence level) for the effects *ease of developing* and *productivity* that the other SRAs.

The reasons that we posit for this conflict is that AUTOSAR has the goal of standardization of concrete architectures (aiming at system/component interoperability), whereas the SRAs of the study S2 focus more on facilitation of the design of concrete architectures (aiming at providing guidelines and inspiration for the design of systems). Also, the study of AUTOSAR mentioned two moderators of this effect: a) the size of the concrete architecture project, e.g., large projects with many developers and highly interconnected functionality is where using AUTOSAR becomes very tough; b) the existence of a tool environment, e.g., tools that help developers while using an SRA. The first moderator, the size of the project, can be present in both contexts (i.e., standardization and facilitation SRAs). However, we can see that facilitation SRAs tend to include more guidelines and a tool environment to facilitate the development of applications, e.g., user manuals, tool prescriptions and plugins, and sample instantiations [16].

Hypothesis: Complexity depends on the type/goal of the SRA (i.e., standardization and facilitation) and on the guidelines that it delivers to facilitate the development. SRAs that aim to standardize tend to be more complex that those that aim to facilitate software systems construction. Providing a tool environment seems to reduce the complexity for both types of SRAs.

- Is it always positive to establish *terminology conventions*?

The study S1 claimed that "*an SRA can serve as a framework and lexicon of terms and naming conventions*" whereas the study S5 stated that "*AUTOSAR practitioners face problems with term confusion*".

One reason we posit for this conflict is that although an SRA could aim to establish term conventions, they do not always reach this benefit. For the case of AUTOSAR [S5], documentation is large (more than 100,000 pages), what could discourage users to completely read these lexicons of terms.

Hypothesis: Although an SRA can define a common lexicon of vocabulary, the success of establishing term conventions depends on the design and documentation size. If documentation is not 'digestible', it may lead to terms confusion when stakeholders are not familiar with those terms.

- Do SRAs allow *flexibility of suppliers*?

In the context of organizations that outsource suppliers to develop their software systems, flexibility of suppliers refers to the capability of an organization to change these suppliers. With the effect "*flexibility of suppliers*" there was also a conflict during the aggregation. In the study S1, the authors state: "*An acquisition program backed up by a strong SRA that ensures interoperability and 'form, fit, and function' compatibility promotes flexibility in the choice of suppliers, as well as a lower risk through multi-sourcing*". However, vendor lock-in moderator of study [S4] shows that "*customers are restricted in changing their system without the involvement of the vendor, despite the use of open standards. Customers try to reduce vendor lock-in, but this is not always possible, given the small market of software vendors in certain domains and the required expertise*". Therefore, SRA adoption does not guarantee flexibility of suppliers, which also depends on other approaches such as the use of open source.

Hypothesis: Despite the use of open standards, mature architectures, and the construction of knowledge repository, outsourcing the construction of SRA-based software may imply vendor lock-in for organizations, jeopardizing the flexibility of suppliers.

C. Contribution of this Aggregation to the Theory on SRAs

Previous studies reported the effects (i.e., benefits and drawbacks) caused by the use of SRAs, as well as the percentage in which they appeared in practice [3]–[5]. Other works have focused on analyzing the practices and constraints of SRA, and qualitatively reported how they moderate or imply the aforementioned effects [1], [2]. By aggregating the results, this is the first study considering the percentages given in previous quantitative studies, and explaining how specific characteristics of SRAs moderate their effects on SRAs.

This work contributes to the body of knowledge of SRAs bringing stronger evidence of their benefits and drawbacks. For most of the effects, the results followed the trends of previous research. However, for three effects these results were contradictory. This observation helped to see that some effects are not general to all types of SRAs, but rather to specific types and contexts. We believe that the aggregated results points to more generalized perceptions and stronger indications of its applicability. Thus, it is expected that practitioners benefit from these indications to support the decision making in practice. Moreover, the stated hypotheses or even the aggregated results themselves can be target of further studies in the future.

VII. VALIDITY

The risks of aggregating diverse evidence (i.e., qualitative and/or quantitative) from different studies have been mitigated by using the SSM method. The SSM method aims to support the SE community to construct and consolidate empirically-grounded knowledge [6]. This section discusses possible threats to validity and emphasizes the mitigation actions used.

To mitigate the threat of missing important primary studies, we systematically searched empirical studies about the benefits and drawbacks of SRAs. We obtained a set of five studies reporting evidence on real SRAs, which is a high number in SE considering that they report the same effects (i.e., benefits and drawbacks of SRAs). During this process, we discarded studies that only reported opinions, rather than empirically-grounded evidence. Even though we found five studies, more studies are needed to reach definitive results.

We are aware that each selected study poses its own validity threats; therefore, we carefully assessed them together with the studies' context to properly interpret their results. Furthermore, while representing empirical evidence from individual studies, researchers can reflect their own opinion and, therefore, bias the representation. To mitigate these subjective issues, the definition and analysis of each individual evidence model from each selected study was first done by a researcher and validated by another one. The studies S2 and S5, conducted by some of the authors, were modelled by the other authors to avoid bias and not to include "extra" knowledge that was not reported in the papers. Then, the aggregated models were assessed and discussed by the whole team. During this process we experienced some semantic issues, meaning that different studies referred to the same concept using different terms. This would lead to a wrong aggregation. To avoid this, we created a glossary of terms that were represented in the evidence models and kept track of the matching terms.

To improve the interpretation of the aggregated evidence, we used some suggested strategies [6]. For instance, given that the SSM method does not consider the different size of sampling of different studies, whenever possible, we refined the confidence level of each effect applying the discount of participants that do not mentioned it, as suggested by the SSM [6]. In addition, we recorded the diverse context of each individual study, so we could better reflect and understand the aggregated evidence. It is important to note that our aggregated results are based on what the authors reported in their papers. Hence, there is always the risk that important information might not be reported. Anyway, in case of doubts we considered the option of contacting the authors to clarify any issues, but this was not the case in this study.

Finally, one of the goals of aggregation in SE is to consolidate empirically-grounded knowledge, to increase whenever possible the generalization of the results and the

understanding of the contexts that might cause any effect. Given the mixed nature (i.e., qualitative/quantitative) of the assessed studies, the aggregated resulting effects could not be statistically but analytically assessed [17]. For this reason, it was highly relevant to properly define the individual contexts of the studies, so we could better understand and interpret the diverse effects of SRAs. We paid special attention to identify the mechanism that produced the studied effects (i.e., moderators). This helped us to explain how the SRA characteristics influence the acquisition organizations. All these strategies have increased the confidence of our results.

Our results show that some effects got higher degrees of belief while others did not. It is important to be aware of the correct interpretation of these results. On the one hand, the effects that got higher belief are potentially those that have been further studied and agreed among the studies. On the other hand, those effects that got lower belief values (or even negative) are those that were just partially approached by the existing evidence (or got contradictory results among the studies). Therefore, these effects are relevant topics that need to be further studied. We highly encourage the SE community to investigate the effects that do not have a high confidence value yet, in order to increase knowledge and consolidation of the benefits and drawbacks of SRA.

VIII. Conclusions

Aggregating evidence of empirical studies helps to increase the confidence with respect to single studies by formulating new theories. Besides, such synthesis reduces the effort of researchers and practitioners that are interested in a particular phenomenon. In order to progress as a discipline, we believe that more aggregation studies are needed in Software Engineering (SE). These studies help to join forces while conducting empirical studies, which becomes more important in SE due to the low number of empirical studies. In this study, we have applied the Structured Synthesis Method (SSM) to aggregate existing evidence about the benefits and drawbacks of Software Reference Architectures (SRA) from five empirical studies. These five studies were identified and selected through a systematic literature review. Although not many studies have used the SSM yet, we understand that the procedures presented in this work can be used by other research peers to evaluate its applicability as a research synthesis method in SE.

As a result of the aggregation process, several benefits of SRAs have considerably increased their belief value: interoperability of software systems, reduced developments costs, improved communication among stakeholders, reduced risk due to previous experiences, use of best practices, and reduced time-to-market. Regarding drawbacks, the studies agree that an SRA weakly negatively - negatively influence the required learning curve for developers before they could construct software systems.

The aggregation also identified three effects of SRAs that were contradictory in the literature: SRA complexity, terminology conventions, and flexibility of suppliers. We have analyzed the context of studies that reported these results, and elaborated hypothesis to explain these contradictions.

The aggregated results point to more generalized perceptions and stronger indications. Thus, practitioners could benefit from them to support decision making in practice. We indicate that future works should focus on gathering more evidence to better understand the conflicting effects or those that have not increased their belief value in this study.

IX. Acknowledgment

This research has been partially supported by CNPq in Brazil, the Spanish grant FPU12/00690, the Spanish project TIN2013-44641-P, and Becas Santander JPI 2014.

References

[1] M. Galster, P. Avgeriou, and D. Tofan, “Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study,” Inf. Softw. Technol., 55 (2), pp. 428–441, 2013.

[2] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone, “The Concept of Reference Architectures,” Syst. Eng., 13 (1), pp. 14–27, 2010.

[3] S. Martínez-Fernández, C. P. Ayala, X. Franch, and H. Marques, “Benefits and drawbacks of reference architectures,” in European Conference on Software Architecture (ECSA), 2013, pp. 307–310.

[4] S. Angelov, J. Trienekens, and R. Kusters, “Software reference architectures-exploring their usage and design in practice,” in European Conference on Software Architecture (ECSA), 2013, pp. 17–24.

[5] S. Martínez-Fernández, C. P. Ayala, X. Franch, and E. Y. Nakagawa, “A Survey on the Benefits and Drawbacks of AUTOSAR,” in Workshop on Automotive Software Architecture (WASA), 2015, pp. 19-26.

[6] P. S. Medeiros Dos Santos and G. H. Travassos, “On the representation and aggregation of evidence in software engineering: A theory and belief-based perspective,” Electron. Notes Theor. Comput. Sci., 292, pp. 95–118, 2013.

[7] E. Nakagawa, P. Antonino, and M. Becker, “Reference architecture and product line architecture: A subtle but critical difference,” in European Conference on Software Architecture (ECSA), 2011, pp. 207–211.

[8] S. Angelov, P. Grefen, and D. Greefhorst, “A framework for analysis and design of software reference architectures,” Inf. Softw. Technol., 54 (4), pp. 417–431, 2012.

[9] D. S. Cruzes and T. Dybå, “Synthesizing evidence in software engineering research,” in ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2010, p. 1.

[10] P. S. Santos, I. Nascimento, and G. H. Travassos, “A Computational Infrastructure for Research Synthesis in Software Engineering,” in XVIII Ibero-American Conference on Software Engineering, 2015, pp. 309–322.

[11] D. I. K. Sjøberg, T. Dybå, B. C. D. Anda, and J. E. Hannay, “Building theories in software engineering,” in Guide to advanced empirical software engineering, Springer, 2008, pp. 312–336.

[12] C. F. Auerbach and L. B. Silverstein, Qualitative data: An introduction to coding and analysis. Qualitative studies in psychology. New York University Press, 2003.

[13] J. Mahoney, “A Tale of Two Cultures: Contrasting Quantitative and Qualitative Research,” Polit. Anal., 14 (3), pp. 227–249, Jun. 2006.

[14] G. Shafer, A Mathematical Theory of Evidence. Princeton University Press, 1976.

[15] D. Atkins, D. Best, P. A. Briss, M. Eccles, Y. Falck-Ytter, S. Flottorp, et al., “Grading quality of evidence and strength of recommendations.,” BMJ, 328 (7454), p. 1490, Jun. 2004.

[16] S. Martínez-Fernández, C. P. Ayala, X. Franch, and H. Marques, “Artifacts of Software Reference Architectures: A Case Study,” in 18th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2014, p. 42:1–42:10.

[17] R. J. Wieringa, “Design Science Methodology for Information Systems and Software Engineering,” Springer, 2014.