Service Oriented Computing and Applications (2020) 14:203–222
https://doi.org/10.1007/s11761-020-00294-x

ORIGINAL RESEARCH PAPER

Enterprise architecture framework evaluation criteria: a literature review and artifact development

Pierre Hadaya[1] · Abderrahmane Leshob[1,2] · Philippe Marchildon[1] · Istvan Matyas-Balassy[1]

Received: 20 February 2020 / Revised: 6 June 2020 / Accepted: 10 June 2020 / Published online: 4 July 2020

Abstract

Selecting an enterprise architecture framework (EAF) that will best address the needs of their organization is a difficult task for enterprise architecture practitioners. The objective of this study is to make this difficult task easier. To do so, this research first conducts a review of the literature on EAF evaluation criteria. Findings from this review show the shortcomings of this literature, most importantly the lack of a comprehensive set of EAF criteria and adequate measures for their operationalization. Based on these findings, and using the design science research approach, this study then designs and tests an EAF evaluation artifact that identifies, elaborates and operationalizes a comprehensive set of 14 criteria. Results of the experiment that followed show that: (i) 90.87% of the criteria were perceived usable, (ii) 97.62% of them were perceived to be applicable and relevant, and (iii) 90.48% were perceived as correct. This study makes several contributions. First, it provides a much-needed and timely overview of the literature on EAF evaluation criteria. Second, this study is the first to present a comprehensive set of criteria that not only synthesizes previously proposed criteria but also includes new criteria (e.g., SOA models and usability). Third, it answers EA practitioners' requests for a tool that is both theoretically sound and practical. Finally, and most importantly, this study is the first to propose objective measures to operationalize EAF evaluation criteria.

Keywords Enterprise architecture · SOA · Literature review · Enterprise architecture framework · Selection criteria · Design science

1 Introduction

To remain competitive in today's highly competitive global markets where change is the only constant, organizations have no choice but to excel at transforming themselves [1]. However, organizational transformations are not easy and more than 50% of them fail to provide expected benefits [2]. Accordingly, practitioners and researchers have proposed a number of best practices and tools to help organizations achieve such a feat despite the numerous challenges it implies. Considered by many as the art of organizational design, the enterprise architecture (EA) approach is one of these practices.

The EA approach rests on three key components: (1) a target enterprise architecture that defines how the organization and its technology assets will need to function in the future, (2) a transformation plan that determines and schedules the transformation projects that the organization will need to execute to implement its target enterprise architecture and (3) an enterprise architecture team who is responsible for creating the target enterprise architecture and the transformation plan [3]. Accordingly, the EA approach allows an organization to transform itself effectively, efficiently and with agility. The EA approach has gained a lot of popularity this last decade, especially since the adoption of the Service-Oriented Architecture (SOA) style and the Business Process Management (BPM) approach that have become its pillars. For the Open Group [4], SOA improves the alignment between the business and information technology communities. It facilitates

✉ Abderrahmane Leshob
leshob.abderrahmane@uqam.ca

Pierre Hadaya
hadaya.pierre@uqam.ca

Philippe Marchildon
Marchildon.philippe@uqam.ca

Istvan Matyas-Balassy
matyas-balassy.istvan@uqam.ca

1 School of Management (ESG), University of Quebec at Montreal (UQAM), Montreal, Canada

2 LATECE Laboratory, University of Quebec at Montreal, Montreal (UQAM), Montreal, Canada

the creation of flexible and reusable assets for enabling end-to-end business solutions. When the SOA style is applied to the EA approach, we talk about Service-Oriented Enterprise Architecture (SOEA).

To help organizations structure and guide their implementation of the EA approach, numerous EA frameworks (EAFs) have been developed over the years. An EAF is defined as a coherent set of principles, methods and models used by practitioners to design, implement and maintain an enterprise's organizational structure, business processes, systems and infrastructure [5]. EAFs provide organizations with (1) one or more metamodels to describe the architecture, (2) one or more methods to design and maintain the architecture and (3) a common vocabulary and optional reference models used as templates or blueprints [6]. EAFs can also be used as tools to access, organize and communicate various architectures that describe key components of the enterprise [7,8].

Yet, despite all the frameworks available, most organizations fail to implement the EA approach. One of the main reasons for such failures is that most enterprise architects are unable to select an EAF that will best address the particular needs of their organization. Indeed, because the selection of an EAF is generally one of the first activities conducted when implementing the EA approach, enterprise architects often lack the knowledge and expertise required to make a good decision. To make matters worst, there are over 25 different EAFs available today and current EAF evaluation tools, which should help organizations select the right EAF, have important limits. First, they compare only a limited number of frameworks. Second, their relevance over time is limited as the EAFs they compared continue to evolve. Third, they rely on somewhat different criteria making it difficult for architects to identify the right set of criteria to guide their EAF selection process [9,10]. Fourth, they limit themselves to the best-known criteria and thus omit several other criteria that may be critical for architects especially when designing the systems and data architectures (e.g., support of the SOA style and the BPM approach). Finally, but not the least, they rely on overly simplistic and subjective operationalizations of the chosen criteria [11].

To help enterprise architects (1) evaluate currently available EAFs, and (2) select the EAF that will best address the particular needs of their organization, the present paper develops, using the design science research (DSR) approach [12,13], an EAF evaluation artifact. DSR aims to create and evaluate artifacts and tools to solve problems identified in organizations. As evaluating and selecting an EAF is one of the most important issues when implementing the EA approach and having established the lack of appropriate tools to help organizations assess currently available EAFs, our proposed EAF evaluation tool, which identifies, defines and operationalizes a comprehensive set of 14 criteria, should allow EA practitioners to more effectively and efficiently select the right EAF and, in turn, help them design and implement an EA approach that will act as a catalyst for the future transformations of their organizations.

In this paper, we extend the work presented in [11], by (1) including the principles and the design of a proof-of-concept prototype that explains how the artifact is concretely operationalized, (2) refining the proposed EAF evaluation criteria through explicit definitions and metrics and (3) reporting the results of an empirical experimentation that validates the relevance, usability and correctness of the proposed evaluation criteria.

The remainder of this paper is structured as follows. Section 2 presents a literature review on EAF evaluation criteria. Section 3 describes each step and related outputs of the methodology we used to design, develop and evaluate our EAF evaluation criteria. The evaluation addressed three aspects, namely the relevance, usability and correctness of the proposed criteria. Section 4 concludes the article by highlighting the study's contributions, limitations and directions for future research.

2 Literature review

Among the different types of literature reviews available, we relied on a scoping review as we wanted to examine the extent, range and nature of research activities on EAF evaluation criteria and grids [14] while focusing more on the breadth of coverage of the literature than the depth of the coverage [15]. The findings presented in this section are a summary of a previously published conference article entitled Enterprise Architecture Framework Selection Criteria: A Literature Review ([11]). Our scoping literature review enabled us to identify nine criteria that are generally used to evaluate and select an EAF.

Table 1 presents a definition of each criterion as well as the articles that discuss it.

Taken as a whole, the literature on EAF evaluation criteria has some strengths as well as important shortcomings. On the plus side, this literature identifies and defines several criteria that might be useful when evaluating and selecting an EAF. On the minus side, past efforts do not provide a comprehensive set of evaluation criteria and most importantly do not provide appropriate scales to evaluate EAFs along these criteria. Indeed, currently available evaluation tools often limit themselves to very few 'best known' criteria and thus omit several other criteria that may be critical for organizations, especially when designing the systems and data architectures (e.g., support of the Service-Oriented Architecture style and Business Process Management approach). In addition, among the 18 articles that identified EAF selection criteria and proposed EAF comparison matrices, only nine of them provided corresponding operationalizations

Table 1 Synthesis of main EAF evaluation criteria in the literature

Criterion/Objective	References
Taxonomy	[7,8,10,16–23]
Evaluates how an EAF defines, describes and classifies all the models that compose the enterprise architecture	
Metamodel	[6,17,20,24]
Evaluates how an EAF defines (i) enterprise architecture constructs in terms of attributes, methods, and associations, and (ii) the modeling rules	
Accelerators	[6,7,10,16,17,20–23]
Evaluates if the EAF comprises specialized software tools, procedures, generic models, templates, patterns or blueprints to accelerate the development of the enterprise architecture	
Development process	[6–8,10,16–18,20–23]
Evaluates if the EAF includes a step-by-step method to design an enterprise architecture	
Maintenance and evolution process	[6,8,16–18,22,23,25]
Evaluates if the EAF comprises processes for maintaining the enterprise architecture and to making it evolve to keep up with the recent changes in the IT/business landscape of the organization	
Principles	[10,16,17,21–23,25]
Evaluates if the EAF provides general rules and instructions to guide the design, implementation and evolution of the enterprise architecture	
Governance process	[7,10,16,17,22,26,27]
Evaluates if the EAF includes a process to carry out the review of the various architecture and maintenance projects to ensure their compliance with the architecture principles while enabling business-IT alignment	
Architecture practice	[7,10,17,20,28]
Evaluates if the EAF promotes the creation of a coherent set of services, processes and human actors with the right roles and responsibilities to design, maintain, make evolve and evaluate the enterprise architecture	
Simplicity	[7,10,22,29]
Evaluates if the EAF is usable by internal resources with limited EA and IT knowledge without needing the help of external experts	

[7,8,10,16,17,20–23]. In other words, solely 9 articles provided actual scales to help EA practitioners evaluate currently available EAFs in regard to these criteria. While these operationalizations certainly represent a step in the right direction, these scales are very simplistic and thus fail to provide any real support when evaluating an EAF. Specifically, the operationalization of the selection criteria in these 9 articles is based on a subjective assessment rather than an objective instantiation or threshold. Take for example the comparison matrix proposed by [10] which ranks the main EAFs available at the time along several criteria using a scale ranging from 1 (very poor) to 4 (excellent). While each EAF is given a score, no objective reason is given as to why any given framework receives a high or a low score on a given criterion. The scores are only based on a subjective assessment of the researchers. As such, although past studies have ranked current available EAFs along certain criteria, they make it very difficult to understand the scores and ranking of each EAF. Accordingly, these evaluation criteria do not allow EA practitioners to make their own assessment of an EAF. This important limit, in turn, undermines the relevance over time of these criteria. Indeed, since an EAF continues to evolve over time, it is impossible to know whether modifications brought to an EAF at certain points in time would bring a different evaluation and, most importantly, which score on which criteria would be improved or worsen.

3 Methodology and results

The DSR approach is adopted here to develop and test our new artifact: an EAF evaluation tool. Based on the work of [13], our approach comprised four steps: (1) problem identification and motivation, (2) definition of the objectives for a solution, (3) design and development and (4) evaluation. During steps 1 and 4, members of the industry and experienced enterprise architects were consulted and asked to provide key suggestions, comments and feedback in order to help us develop and evaluate our new artifact. More precisely, experienced architects were consulted to clarify the artifact's objectives and to assess its relevance, usability and correctness [12,30]. Semi-structured interviews were used to collect their suggestions, comments and feedback during these steps. The following paragraphs detail what was done and the key findings of each step.

3.1 Step 1—problem identification and motivation

During step 1, exploratory interviews were conducted with seven experienced practitioners to assess (1) whether the EA approach was a preoccupation for them, (2) the challenges they faced in regard to implementing such an approach, (3) if they knew which EAF they would use to guide their efforts and (4) if an EAF evaluation tool could help them select an EAF that will best address the needs of their organization. Participants included senior enterprise architects from large organizations and other experienced EA and BPM specialists that had already implemented the EA approach. Among key findings from this step, transcripts from the exploratory interviews indicated that all respondents agreed that digital transformations have rendered the EA approach a central organizational preoccupation. All participants mentioned that some of their initiatives to implement the EA approach had failed in the past. For example, EA consultants explained that their customers' first attempt to implement the EA approach often failed because they did not know where to start and lacked the knowledge and experience required to properly coach all the stakeholders involved. In addition, most participants mentioned that leaders had a hard time determining which EAF they should use to guide their efforts when implementing the EA approach. These participants emphasized that selecting an EAF is usually one of the first key decisions to make when implementing the EA approach and that having to make such an important decision, at such an early stage, without enough time to get familiar with the strengths and weaknesses of the different EAFs, was extremely difficult. They also mentioned that although they came across a few EAF comparison matrices, these were of little value as they did not provide scales which would have allowed them to make their own assessment of currently available EAFs. Finally, all participants mentioned that an objective EAF evaluation artifact like ours would be of great value to enterprise architects. They perceived our DSR effort as a way to obtain such a tool and a means by which they could make an informed decision when selecting an EAF.

3.2 Step 2—definition of the objectives for a solution

During step 2, findings from our interviews conducted in step 1 and the knowledge gathered during our review of the relevant literature was used to infer the objective of our artifact. In broad terms, we wanted to develop an artifact to help enterprise architects evaluate and select an EAF that best addresses their needs. Specifically, our objective was threefold. First, our artifact had to identify and define a comprehensive set of criteria that not only assess the ' best known' characteristics of EAFs but also assesses the more practical and technical aspects of EAFs. Second, our artifact had to provide objective scales for the criteria. Third, our artifact had to allow the assessment of currently available EAFs (e.g., TOGAF, DODAF, FEAF). With this aim and objective in mind, we elected to design an EAF evaluation tool.

3.3 Step 3—design and development

During step 3, because no such tool had been previously developed, we used a ' general solution' strategy to develop the first version of our EAF evaluation artifact [44]. To do so, we first anchored our efforts on the literature we reviewed previously as well as the interviews we conducted during step 1. In addition, considering the important limits of the EAF evaluation criteria literature, we conducted additional literature reviews on several closely related EA topics (e.g., EA frameworks, metamodels, SOA, BPM). After doing so, we were confident that our EAF evaluation artifact would not omit any important criteria and that it would be compatible with existing EAFs. Since none of the existing EAF evaluation tools identified in the academic and practitioner literature provide a comprehensive set of EAF evaluation criteria and appropriate scales, missing criteria and scales were then developed.

3.3.1 Design of the artifact

To design these criteria and their respective scale, existing EAF criteria and EAF comparison matrices were used. Specifically, we first analyzed different components and particularities of existing EAFs as well as the various EAF evaluation criteria and scales provided in already existing comparison matrices. As mentioned above, we also conducted additional literature reviews on several closely related EA topics to ensure that we would not omit any important criteria. In total, 14 criteria were developed. Furthermore, for each of these criteria, we provided the definitions of key terms required to properly assess them, their metrics, their overarching assessment logic and their respective objective scale. The 14 identified criteria are specified in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15.

3.3.2 Prototype architecture: a first sketch

This section discusses the design and implementation of a proof-of-concept prototype that supports the proposed

Table 2 Taxonomy—architecture layers

Definitions	*EA taxonomy* Defines, classifies and summarily describes the various models that make up the EA [7,17]. EA taxonomies can be grouped into two types: (1) those only containing architecture layers, and (2) those containing architecture layers and aspects.
	Architecture layers Represent the hierarchical structures of an organization's components. We consider that there are four core layers:
	1. Business: the representation of an organization's capabilities, functions, processes, organizational units and assets [3]
	2. Application: the representation of an organization' s application systems [4]
	3. Data: the representation of an organization' s logical and physical data assets and data management resources [4]
	4. Technology: the representation of an organization's hardware and software technologies (e.g., IT infrastructure, middleware, networks, communications, processing, standards) [4]
	Other layers include but are not limited to: i) the implementation and ii) the strategy (i.e., the representation of an organization's values, mission, vision and stakeholder value propositions [3])
Overarching assessment logic	A taxonomy of layers of an EA framework F (denoted T_{lf}) is considered satisfactory if the set of its architecture layers, denoted F_{al}, is composed of the four core architecture layers (i.e., business, application, data and technology). Thus
	T_{lf} is satisfactory $<=>$ $F_{al} = \{businesslayer, applicationlayer, datalayer, technologylayer\}$
Metrics and operationalization	
Value	
0	$F_{al} = \{\}$
	The framework has no architecture layers.
1	$F_{al} \neq \{\}$
	and
	$\neg(F_{al} \supseteq \{businesslayer, applicationlayer, datalayer, technologylayer\})$
	The framework does not comprise the four core architecture layers.
2	$F_{al} = \{businesslayer, applicationlayer, datalayer, technologylayer\}$
	The framework comprises the four core architecture layers.
3	$F_{al} \supset \{businesslayer, applicationlayer, datalayer, technologylayer\}$
	The framework comprises the four core architecture layers and other layers.

artifact. This first sketch of the artifact was designed and developed to help enterprise architects evaluate EAFs by computing the metrics and assigning a score to each of its criterion.

We designed the core components of the artifact using the SOA style [36]. More precisely, the SOA services were designed as RESTful web services. The SOA services were implemented with the Eclipse Modeling Framework™ (EMF). EMF is a Java-based modeling framework that implements EMOF (Essential Meta-Object Facility). As shown in the UML components diagram of Fig. 1, the artifact is based on an Eclipse plugin, *eaf.selection.criteria*, representing criteria assessor services.

The plugin consists of four sub-packages (ea.models, ea.metamodel, ea.taxonomy, ea.process) and fourteen components (SOA services). Each sub-package is designed as an SOA-based Eclipse plugin. Each component provides functions to calculate metrics for the corresponding evaluation criterion. Note that the metamodel complexity component, ComplexityAssessor, uses the Archimate metamodel as a reference model to compute the complexity metrics (See Table 4). To do so, it uses EMF Refactor, an Eclipse open-source tool that supports metrics reporting, smell detection and models refactoring. On the other hand, the Usability Assessor component depends on external libraries to compute usability metrics. To assess the completeness of an EAF (i.e., if the EAF supports the core EA layers and aspects), the Completeness Assessor component uses the sub-package 'ea.taxonomy' components (i.e., EA Layers Assessor and EA Aspects Assessor). Finally, the Reference Models Assessor component provides functions to the SOA Assessor through its interface in order to evaluate the SOA criterion.

3.4 Step 4—evaluation

The evaluation of the proposed artifact required conducting an experiment to assess the relevance, the usability and the correctness of the EAF evaluation criteria. More precisely,

Table 3 Taxonomy—architecture aspects

Definitions	*EA taxonomy* see Table 2.
	EA Architecture aspects Aspects are abstractions on a set of models aimed at stakeholders or aimed at a set of concerns (e.g., time, motivation, structure) [31].
	We consider that there are three core aspects:
	1. Structure: the set of entities (active vs. passive and/or internal vs. external) that display some behavior (e.g., business actors, devices, or application components) or those on which actions are conducted (e.g., information or physical objects) [32]
	2. Behavior: the processes, functions or activities performed by entities [32]
	3. Motivation: the context and motives (e.g., goals and objectives) behind the architecture [32]
	Other aspects include but are not limited to: (i) Location (i.e., conceptual or physical place or position where structures and/or behaviors are located or performed) and (ii) Time (i.e., the points in time and the duration of behaviors) [33]
Overarching assessment logic	A taxonomy of aspects of an EA framework F (denoted T_{af}) is considered satisfactory if the set of its architecture aspects, denoted F_{aa}, is composed of the three core architecture aspects (i.e., structure, behavior and motivation). Thus,
	T_{af} is satisfactory $<=> F_{aa} = \{structure, behavior, motivation\}$
Metrics and operationalization	
Value	
0	$F_{aa} = \{\}$
	The framework has no architecture aspects.
1	$F_{aa} \neq \{\}$ and
	$\neg(F_{aa} \supseteq \{structure, behavior, motivation\})$
	The framework does not comprise the three core architecture aspects.
2	$F_{aa} = \{structure, behavior, motivation\}$
	The framework comprises the three core architecture aspects.
3	$F_{aa} \supset \{structure, behavior, motivation\}$
	The framework comprises the three core architecture aspects and other aspects.

this experiment aimed to verify whether the evaluation criteria shown in Table 16 are:

1. *relevant* when instantiated in the context of EAFs (i.e., criteria 'instantiation' are meaningful within the context of EAFs),
2. *usable* when instantiated to evaluate EAFs. In this context, we want to assess whether the criteria are effective and do not require advanced technical skills in enterprise architecture, software design and architecture to be assessed in the context of EAFs.
3. *correct* (i.e., allow architects to evaluate and classify EAFs effectively).

To conduct this experiment, we followed the Goal Question Metric (GQM) approach [45]. GQM defines three levels [45]: (i) the goal of the experiment (conceptual level), (ii) the set of questions used to characterize the way to attain the specific goal (operational level) and (iii) the set of metrics that provides the needed information to answer the questions (quantitative level). In addition, because the goal of this experiment is to assess the *relevance, usability* and *correctness* of the criteria, we needed experts with the appropriate expertise to evaluate the fourteen criteria.

The rest of this section is organized as follows. Sub-section 3.4.1 describes the experiment setup. It presents the participant selection (i.e., the experts) and the set of selected EAFs used as experimental objects. Sub-section 3.4.2 discusses the experimental design to evaluate the relevance, usability and correctness of the criteria. Sub-section 3.4.3 presents the experimental operation and execution. The results of the evaluation are presented in Sect. 3.4.4. Finally, Sect. 3.4.5 discusses the issues that might have affected the validity of the experiment.

3.4.1 Experimental subjects and EAFs

Participant Selection: Three senior enterprise architects volunteered to participate in the experiment and assess the usability, relevance, and correctness of our fourteen EAFs criteria. As a group, the participants had several years of experience in enterprise architecture, software architecture, business architecture and a solid BPM expertise. Table 17 summarizes the profile of the participants.

Table 4 Metamodel—complexity

Definitions	*Metamodel* A model that defines the language for expressing models. It defines the constructs of the language (e.g., classes, attributes, associations) and the modeling rules. A metamodel is also used to model arbitrary metadata (e.g., software configuration or requirements metadata) [34]. *Complexity* The extent to which the metamodel is understandable, analyzable and modifiable [35]. As suggested in [35], we selected the metamodel of Archimate as the most representative among EAF metamodels. We also eliminated redundant metrics for measuring size and structure complexity. Thus, we use the following metrics [35]: 1. *Number of Classes (NC)*: The total number of classes 2. *Number of Associations (NAssoc)*: The total number of associations and aggregations 3. *Number of Generalization Hierarchies (NGenH)*: The total number of generalization hierarchies in a class diagram 4. *Maximum DIT (MaxDIT)*: The maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalization hierarchy is the longest path from the class to the root of the hierarchy To measure the metamodel complexity, we calculate the normalized value (between 0 and 1) of the NC, NAssoc, NGenH and MaxDIT. Then, we calculate the distance with Archimate normalized values. For example, the distance of a metric A for a metamodel M is calculated as follows: $Distance(A) = \|N_{ma} - N_{aa}\|$ where N_{ma} is the normalized value of the metric A for the metamodel M and N_{aa} is the normalized value of the metric A for Archimate.
Overarching assessment logic	The complexity of an EA framework is considered satisfactory if the distance for each metric (i.e., NC, NAssoc, NGenH, and MaxDIT) is less than or equal to 0.5. Thus, $Distance(NC) <= 0.5$ and $Distance(NAssoc) <= 0.5$ and $Distance(NGenH) <= 0.5$ and $Distance(MaxDIT) <= 0.5$
Metrics and operationalization	
Value	
0	$Distance(NC) > 0.5$ and $Distance(NAssoc) > 0.5$ and $Distance(NGenH) > 0.5$ and $Distance(MaxDIT) > 0.5$
1	$Distance(NC) <= 0.5$ and $Distance(NAssoc) <= 0.5$ and $Distance(NGenH) > 0.5$ and $Distance(MaxDIT) > 0.5$
2	$Distance(NC) <= 0.5$ and $Distance(NAssoc) <= 0.5$ and $Distance(NGenH) <= 0.5$ and $Distance(MaxDIT) <= 0.5$
3	$Distance(NC) <= 0.5$ and $Distance(NAssoc) <= 0.5$ and $Distance(NGenH) <= 0.25$ and $Distance(MaxDIT) <= 0.25$

Experimental Enterprise Architecture Frameworks: To carry out the experiment, we first studied 12 EAFs. From these EAFs, we selected the six that are the most widely used in practice and cited in the literature, namely Zachman, TOGAF (The Open Group Architecture Framework), FEAF (Federal Enterprise Architecture Framework), EAP (Enterprise Architecture Planning), The Enterprise Architecture IT Project (Urbanization approach) and DoDAF (Department of Defense Architecture Framework). All the experts had experience and a strong knowledge of the experimental EAFs in their CURRENT- 1 versions (i.e., latest version and previous stable versions). Table 18 lists the experimental EAFs.

Table 5 Metamodel—completeness

Definitions	*Metamodel* See Table 4
	Completeness The extent to which the metamodel defines the concepts tied to all the architecture layers and aspects of an EAF (see Tables 2 and 3).
Overarching assessment logic	A metamodel M of an EA framework F (denoted M_f) is considered satisfactory if it includes the concepts, denoted F_{mc},tied to the four core architecture layers (i.e., business, application, data and technology) and the three core architecture aspects (i.e., structure, behavior and motivation). Thus,
	M_f is satisfactory $<=> F_{al} = \{businesslayer, applicationlayer, datalayer, technologylayer\} \cup \{structure, behavior, motivation\}$
Metrics and operationalization	
Value	
0	$F_{al} = \{\}$
	The framework does not define a metamodel.
1	$F_{al} \neq \{\}$ and
	$\neg(F_{al} \supseteq \{businesslayer, applicationlayer, datalayer, technologylayer\} \cup \{structure, behavior, motivation\})$
	The metamodel does not include concepts tied to each of the core EA layers and aspects.
2	$F_{al} \supseteq \{businesslayer, applicationlayer, datalayer, technologylayer\} \cup \{structure, behavior, motivation\}$
	The framework includes concepts tied to each of the core EA layers and aspects.
3	$F_{al} \supset \{businesslayer, applicationlayer, datalayer, technologylayer\} \cup \{structure, behavior, motivation\}$
	The metamodel includes concepts tied to each of the core EA layers and aspects as well as other layers and / or aspects.

3.4.2 Experimental design

According to the GQM paradigm [45], the *goal* of this experiment is to evaluate the 14 criteria of the artifact (See Table 16) with the purpose of assessing their *relevance*, *usability* and *correctness* from the point of view of the experts. To achieve this goal, we encoded three questions:

1. *RQ1:* Are the criteria *relevant* to evaluate EAFs?
2. *RQ2:* Are the criteria *usable* to evaluate EAFs?
3. *RQ3:* Do the criteria evaluate EAFs *correctly*?

These questions allowed us to gain empirical evidence on whether the identified criteria are relevant, usable and correct in the context of different EAFs.

The context of the experiment is determined by: (i) the six selected EAFs (see Table 18), (ii) the 14 criteria to be applied for evaluating EAFs (see Table 16), and (iii) the three experts that evaluate the criteria being applied to the EAFs (see Table 17).

To conduct the experiment, we presented the 14 criteria and six EAFs to the experts and asked them to judge whether each criterion (e.g., Reference models), as instantiated for a particular EAF (e.g., TOGAF), was perceived as usable, relevant and correct.

For each aspect (i.e., relevance, usability and correctness) to validate, we had 14 variables that represent the criteria in table 16. When applied to an EAF, these variables can have two possible values (i.e., the usability, the relevance and the correctness): 0, meaning that the criterion was found not usable/irrelevant/incorrect in the context of the EAF; 1, meaning that the criterion, as instantiated, was found usable/relevant/correct for the EAF in question.

The experiment had 42 hypotheses:

$H1i_0$: The criterion Ci was found to be irrelevant/ $H1i_a = \neg H1i_0$; For i=1 to 14
$H2i_0$: The criterion Ci was found to be not usable/ $H2i_a = \neg H2i_0$; For i=1 to 14
$H3i_0$: The criterion Ci was found to be incorrect/ $H3i_a = \neg H3i_0$; For i=1 to 14

3.4.3 Experiment operation and execution

The experiment was conducted in one session. Several documents and instruments were designed to introduce the participants to the context of our research project. The materials included: (1) training slides with an overview of the artifact and the proof-of-concept prototype, (2) the description of the criteria and (3) a questionnaire for gathering the data.

The experiment took place in a single room, and no interaction between subjects was allowed. First, the conductors briefly trained the subjects on the evaluation criteria and the

Table 6 SOA models

Definitions	*Service-oriented architecture (SOA)* An architectural style that provides a way of describing and understanding organizations, communities, processes and systems to maximize agility and interoperability [36] — According to the Open Group [4], SOA provides insights, patterns, and the building blocks for integrating fundamental elements of an SOA into a solution or enterprise architecture — SOA has been advocated as an evolutionary step in software architecture to help architects overcome the challenge of building effective software to support enterprise business processes [37,38] — SOA models consist of logical models (e.g., SoaML diagrams) and solutions models (e.g., SOA Rest services, SOAP services)
Overarching assessment logic	The SOA reference models of an EA framework F (denoted SOA_f) are considered satisfactory if the set of its SOA models, denoted F_{soa}, covers generic and specific SOA models for the application and data layers. Generic models are domain independent while specific SOA models are domain or industry specific. Thus: SOA_f is satisfactory <=> $F_{soa} \supseteq \{genericSOAmodels, specificSOAmodels\}$
Metrics and operationalization	
Value	
0	$F_{soa} = \{\}$ The framework does not provide any SOA models.
1	$F_{soa} = \{genericSOAmodels\}$ The framework provides only generic SOA models.
2	$SOA_f \supseteq \{genericSOAmodels, specificSOAmodels\}$ The framework provides generic and domain specific SOA models for the application and data layers.
3	$F_{soa} \supseteq \{genericSOAmodels, specificSOAmodels, solutionblocks\}$ The framework provides generic and specific SOA models for the application layer along with solutions/implementation services (e.g., REST services).

SoaML (Service-Oriented architecture Modeling Language) is a language that extends the UML metamodel to explicitly support service modeling and requirements for SOA [36]

artifact and answered their questions. Then, a slot of 120 minutes without a time limit was given to participants to evaluate the three aspects (i.e., relevance, usability and correctness) of each criterion in the context of each experimental EAF. For that, the experts had to, a priori, compute the metrics of each criterion (e.g., *Number of Classes*, *Number of Associations*, *Number of Generalization Hierarchies* and *Maximum DIT* of the criterion *Metamodel Complexity)* in the context of each of the selected EAFs (e.g., TOGAF). To assess the usability criterion (Table 15), experts were asked to compute the metrics (i.e., effectiveness, efficiency and satisfaction) based on their most recent experience with the selected EAFs. Questions that arose during the session were clarified by the conductors.

For each criterion of each EAF, the questionnaire was structured to enable experts, after computing a metric, to: (i) give it a rating between 0 and 3, and (ii) answer binary questions (i.e., Yes/No) related to the aspects to validate (i.e., relevance, usability and correctness). Additional space was available for the experts to add comments explaining their answers to each criterion of each EAF.

Evaluation of the relevance The goal of first step of the experiment is to evaluate, from the expert's perspective, the relevance of the 14 criteria in the context of each of the selected EAFs. A criterion is relevant for a particular EAF when: (i) it is applicable (i.e., can be instantiated) and (ii) is meaningful when instantiated.

For this evaluation, we presented to the experts the 14 questions (Is the criterion *Ci relevant* to evaluate EAFs? For i=1 to 14) we encoded Q1 to Q14. Then, we asked them to answer the 14 questions in the context of each EAF. For each question (Q1 to Q14), a bi-valuated variable was used to enable participants to provide their perception on the relevance. As mentioned above, the bi-valuated variables had two possible values: 0, which means that the expert finds the criterion irrelevant, and 1, which means that the expert finds the criterion relevant when instantiated in the context of a particular EAF. To collect accurate information about their perception, we included an open-ended question to enable participants to add comments explaining why they think that the criterion is relevant or not.

Table 7 Reference models

Definitions	*AE Reference models* Models used as references or patterns to accelerate the development of an architecture [4,17]
	— Reference models are reusable artifacts for the architecture, such as business services for the business layer and SOA services for the application and data layers
	— References models consist of logical models (e.g., UML diagrams, BPMN models) and solutions models (e.g., platform-specific components)
Overarching assessment logic	The reference models of an EA framework F (denoted RM_f) are considered satisfactory if the set of its reference models, denoted F_{rm}, covers the four core architecture layers (i.e., business, application, data and technology). Thus
	RM_f is satisfactory <=>
	$F_{rm} = \{businesslayer, applicationlayer, datalayer, technologylayer\}$
Metrics and operationalization	
Value	
0	$F_{rm} = \{\}$
	The framework does not provide any reference models.
1	$F_{rm} \neq \{\}$ and $\neg(F_{rm} = \{businesslayer, applicationlayer, datalayer, technologylayer\})$
	The framework does not provide reference models for the four core architecture layers.
2	$F_{rm} = \{businesslayer, applicationlayer, datalayer, technologylayer\}$
	The framework provides reference models for the four core architecture layers.
3	$F_{rm} \supset \{businesslayer, applicationlayer, datalayer, technologylayer\}$
	The framework provides reference models for the four core architecture layers as well as other layers (e.g., Implementation and Strategy), aspects, and/or industries (e.g., financial services and government).

Evaluation of the usability The goal of the second step of the experiment is to evaluate the usability of the 14 criteria in the context of each of the selected EAFs. Through this evaluation, we wanted to know, for a particular EAF, whether or not the 14 criteria of our artifact require advanced technical skills in enterprise architecture, software design and architecture to be measured.

For this evaluation, we presented to the experts the 14 questions (Is the criterion *Ci usable* to evaluate EAFs? For i=1 to 14) we have encoded Q15 to Q28. Then, we asked them to answer the questions in the context of each EAF. For each question (Q15 to Q28), a bi-valuated variable was used to enable participants to provide their perception of the usability criterion. The bi-valuated variables have two possible values: 0, which means that the expert finds the criterion not usable, and 1, which means that the expert finds the criterion usable when instantiated in the context of a particular EAF. In addition, we included an open-ended question to enable participants to add comments explaining why they think the criterion is usable or not.

Evaluation of the correctness The goal of the third step of the experiment is to evaluate, from the expert's perspective, the correctness of the 14 criteria in the context of selected EAFs. A criterion is correct for a particular EAF when the associated metrics are (i) correct and (ii) allows architects to evaluate and classify EAFs effectively.

For this evaluation, we presented to the experts the 14 questions (Does the criterion *Ci* evaluate EAFs *correctly*? For i=1 to 14) we have encoded Q29 to Q42. Then, we asked them to answer the questions in the context of each EAF. For each question, a bi-valuated variable was used to enable participants to provide their perception of the correctness criterion. The bi-valuated variables have two possible values: 0, which means that the expert finds the criterion incorrect, and 1, which means that the expert finds the criterion correct when instantiated in the context of a particular EAF. In addition, we included an open-ended question to enable participants to add comments explaining why they think that the criterion is whether correct or not.

3.4.4 Analysis of the results

Analysis of the relevance assessment

Table 19 depicts the assessments of the relevance aspect in the context of the experimental EAFs. The results show that, overall, the experts found the proposed criteria relevant. Indeed, for Expert 1 and Expert 2, the 14 identified criteria were deemed relevant. However, Expert 3 found the criterion Metamodel-Completeness (C3) irrelevant. He argued that the criterion C3 is based on two other criteria: Taxonomy-Architecture Layers (C1) and Taxonomy-Architecture Aspects (C2). Thus, enterprise architects could assess the completeness of the EAF

Table 8 Development process

Definitions	*Development process* A well-defined process that describes each step of the design and implementation of the enterprise architecture[5,6,39]. According to Curtis *et al.* ([40]), a process is a partially ordered set of tasks or steps undertaken to attain a specific goal and its complete definition requires four distinct views:
	1. The functional view represents the activities being performed and the dependencies between them, such as producer-consumer dependencies
	2. The dynamic view provides sequencing and control dependency information between the activities
	3. The organizational view describes who performs each process activity, where in the organization it is performed, and the communication mechanisms
	4. The informational view includes a description of the entities that are manipulated by the process (e.g., artifacts, products, etc)
	Technique A precisely described procedure to accomplish a task [41]
	Template A structured model used to document and describe various architecture artifacts [42]
Overarching assessment logic	The development process of an EA framework F (denoted DP_f) is considered satisfactory if the set of its process views, denoted PV_f, contains the functional and dynamic views. Thus:
	DP_f is satisfactory $<=>$
	$PV_f \supseteq \{functionalview, dynamicview\}$
Metrics and operationalization	
Value	
0	$PV_f = \{\}$
	The framework does not have a development process.
1	$PV_f \neq \{\}$ and $\neg(PV_f \supseteq \{functionalview, dynamicview\})$
	The EAF development process does not have the functional and dynamic views.
2	$PV_f \supseteq \{functionalview, dynamicview\}$
	The EAF development process has the functional and dynamic views.
3	$PV_f \supset \{functionalview, dynamicview\}$
	The EAF development process has the functional and dynamic views. It also includes other views (e.g., organizational view) with techniques and/or templates for the process tasks.

metamodel from the criteria C1 and C2 as an EAF metamodel must implement all the specified layers (C1) and aspects (C2). In other word, Expert 3 perceived that it was not necessary to compute the metrics for the criterion METAMODEL-COMPLETENESS (C3) because they could be inferred. Thus, he concluded that C3 was irrelevant. While we agree with Expert 3 that the criterion C3 metrics depend on C1 and C2, we argue that C3 is still going to be used by: (i) EA tools providers when implementing EAF metamodels, and (ii) EA architects who need to instantiate the metamodel to create architecture models. Going a step further, EA architects could also infer SOA MODELS (C6) metrics from those of REFERENCE MODELS (C5).

These observations allowed us to reject the null hypotheses $H1i_0$ for i in {1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14} and to accept their alternative hypotheses, meaning that all criteria, except C3, were perceived by the experts as relevant.

Analysis of the usability assessment

As shown in Table 20, the experts found that the criterion METAMODEL-COMPLEXITY (C4) was not usable. The experts argued that its metrics are difficult to compute despite the use of measurement tools, including the proof-of-concept prototype. Experts pointed out that the metrics used to assess the metamodel complexity are even more complex to compute when EAF metamodels are described textually or in less formal language. It is worth noting, however, that Expert 3 found that the metrics were usable in the context of the TOGAF framework as its metamodel is well specified.

In addition, Expert 2 found that the criterion USABILITY (C14) was not easy to measure in the context of the six EAFs. A closer analysis of his comments revealed that he perceived the metrics used to assess the USABILITY criterion as too complex. He suggested using solely the SATISFACTION attribute and discarded the other two (i.e., EFFECTIVENESS and the EFFICIENCY).

Overall, we believe that the ‘poor performance’ on the criteria C4 and C14 was due to a combination of factors: (1) the metrics do not appear to be ‘intuitive’ and easy to compute, (2) the unavailability of measurement data (UML metamodel for C3 and statistical data for C14), and (3) the conductors explanations during the experiment lacked clarity.

Table 9 Governance process

Definitions	*Governance process* A well-defined process that reviews architecture projects to make sure they conform with the established architecture principles while enabling business-IT alignment [17]
	According to Curtis *et al.* ([40]), a process is a partially ordered set of tasks or steps undertaken to attain a specific goal and its complete definition requires four distinct views (see Table 8)
Overarching assessment logic	The governance process of an EA framework F (denoted GP_f) is considered satisfactory if the set of its process views, denoted GV_f, contains the functional and dynamic views. Thus:
	GP_f is satisfactory <=>
	$GV_f \supseteq \{functionalview, dynamicview\}$
Metrics and operationalization	
Value	
0	$GV_f = \{\}$
	The framework does not have a governance process.
1	$GV_f \neq \{\}$ and $\neg(GV_f \supseteq \{functionalview, dynamicview\})$
	The EAF governance process does not have the functional and dynamic views.
2	$GV_f \supseteq \{functionalview, dynamicview\}$
	The EAF governance process has the functional and dynamic views.
3	$GV_f \supset \{functionalview, dynamicview\}$
	The EAF governance process has the functional and dynamic views. It also includes other views with techniques and/or templates for the process tasks.

Table 10 Supporting software

Definitions	*Supporting software* Software applications designed to create, store, integrate, structure, analyze and communicate EA related information.
	Generic software Software created for a non-EA-related domain but has functionalities useful for EA (e.g., Microsoft Visio, Microsoft SharePoint)
	Incomplete EA software Software created specifically for EA that enables the creation of models tied to some but not all EA architecture layers and aspects.
	Complete EA software Software created specifically for EA that enables the creation of models tied to all EA architecture layers and aspects directly or via connections to other software
Overarching assessment logic	To be considered satisfactory, an EAF needs to be supported by complete EA software.
Metrics and operationalization	
Value	
0	The framework is only supported by generic software.
1	The framework is supported by incomplete EA software.
2	The framework is supported by complete EA software.
3	The framework is supported by complete EA software that also provides additional functionalities (e.g., automation and monitoring).

Based on the results, we rejected the null hypotheses $H2i_0$ for i in {1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13} and accepted their alternative hypotheses, meaning that all the criteria, except C4 and C14, were perceived by the experts as usable.

Analysis of the correctness assessment

Table 21 depicts the assessments of the correctness aspect of the proposed criteria in the context of the experimental EAFs. The experts found that, among the 14 criteria, Taxonomy-Architecture Layers (C1), Metamodel-Completeness (C3), Metamodel-Complexity (C4), and Usability (C14) were not always correct.

First, Expert 1 and Expert 2 indicated that, although several EAFs combine the data and application layers into one, the metrics for the criteria C1 and C3 required a clear separation between the data and application. As such, they assessed that these two criteria were incorrect. Second, Expert 2 found

Table 11 Availability of free information and supporting software

Definitions	*Free information* The primary and secondary information can be found freely
	Primary information Information on the framework provided by its creator(s)
	Secondary information Information on the framework provided by other parties
	Free supporting software The supporting software can be found freely
Overarching assessment logic	For an EAF to be considered satisfactory, free primary and secondary information on the framework must be available.
Metrics and operationalization	
Value	
0	There is no free information and supporting software.
1	There is free primary information but no free supporting software.
2	There is free primary and secondary information but no free software.
3	There is free primary and secondary information as well as free supporting software.

Table 12 Architecture practice guidelines

Definitions	*EA practice guidelines* A set of coherent roles, responsibilities and skills for the people who participate in the creation, maintenance and evolution of the EA
Overarching assessment logic	The architecture practice guidelines of an EA framework F (denoted APG_f) is considered satisfactory if the set of guidelines, denoted F_{apg} , includes roles, responsibilities and skills. Thus APG_f is satisfactory $<=> F_{apg} \supseteq \{roles, responsibilities, skills\}$
Metrics and operationalization	
Value	
0	$F_{apg} = \{\}$
	The framework does not include architecture practice guidelines.
1	$F_{apg} \neq \{\}$
	and
	$\neg(F_{apg} \supseteq \{roles, responsibilities, skills\})$
	The architecture practice guidelines do not include roles, responsibilities and skills.
2	$F_{apg} \supseteq \{roles, responsibilities, skills\}$
	The architecture practice guidelines include roles, responsibilities and skills.
3	$F_{apg} \supset \{roles, responsibilities, skills\}$
	The architecture practice guidelines include roles, responsibilities, skills and other guidelines.

the metrics for metamodel Complexity (C4) and Usability (C14) not really accurate. A closer analysis of his comments showed that he proposed other metrics, which he found more accurate for C4 and C14. More precisely, he proposed to use the object-oriented design coupling metrics from [46] to compute model complexity and the single and summated usability metric proposed in [47]. On the other hand, Expert 1 and Expert 3 found that the metrics used for C4 and C14 were correct, but not easily applicable to other frameworks (value *1 with textbold background* in table 21). However, unlike expert 2, they did not propose other metrics.

Taking into account the experts comments, we realized that: (1) the metrics for C1 and C3 must be refactored to allow architects to compute them when taxonomy layers are merged, and (2) the metrics for C4 and C14 must be improved to allow precise and generic measurement of the metamodel complexity and usability.

Based on these observations, we reject the null hypotheses $H3i_0$ for i in {2, 5, 6, 7, 8, 9, 10, 11, 12, 13} and accept their alternative hypotheses, meaning that criteria C2, C5, C6, C7, C8, C9, C10, C11, C12 and C13 were perceived by the experts as correct.

Figure 2 summarizes the results of the evaluation of the criteria from the point of view of the EA experts. Among the 756 evaluations (14 criteria x 6 EAFs x 3 experts x 3 aspects to evaluate), participants confirmed that: (1) 90.87% of the criteria were perceived usable, (2) 97.62% of them were perceived to be relevant, and (3) 90.48% of them were perceived as correct.

Table 13 Principles

Definitions	*Principles* Defines the general rules and instructions for the deployment of resources to design and implement the EA *Complete principle* A complete principle provides a *Justification* (i.e., highlights the business benefits of adhering to the principle) and its *Implication* (i.e., highlights the requirements, for both the business and IT for carrying out the principle in terms of resources, costs, and activities/tasks) [4,25]
Overarching assessment logic	The principles of an EA framework F (denoted P_f) are considered satisfactory if the set of its principles, denoted F_p, contains a Justification and Implication. Thus P_f is satisfactory $<=> F_p \supseteq \{justification, implication\}$
Metrics and operationalization	
Value	
0	$F_p = \{\}$ The framework does not include principles.
1	$F_p \neq \{\}$ and $\neg(F_p \supseteq \{justification, implication\})$ The framework includes incomplete principles.
2	$F_p \supseteq \{justification, implication\}$ The framework includes complete principles.
3	$F_p \supset \{justification, implication\}$ The framework includes complete principles along with the stakeholders affected by them.

Table 14 Adaptability

Definitions	*Adaptability* The extent to which the EAF can be adapted according to the specific needs/contexts of the organizations. The adaptation may be related to any part of the framework (e.g., methodology, principles, practice guidelines)
Overarching assessment logic	An EAF is considered adaptable, if it is possible to use parts of the framework and these parts can be changed.
Metrics and operationalization	
Value	
0	The framework can only be used in its entirety.
1	It is possible to use parts of the framework.
2	It is possible to use parts of the framework and these parts can be adapted by architects.
3	It is possible to use parts of the framework and these parts can be adapted. In addition, the framework provides complete guidelines and tools to adapt these parts.

3.4.5 Threats to the validity

In this section, we explain the main issues that may have threatened the validity of the experiment. We consider threats to internal, external and construct validity as discussed in [48].

Internal validity The threats to internal validity compromise the confidence to confirm a relationship between the independent and dependent variables. This is relevant when the study's goal is to establish a causal relationship between variables. In the particular context of our experiment, threats to internal validity were mainly related to participants' experience and possible information exchange between them which both might impact EAF criteria evaluation. To mitigate the threat related to the profile of experts, we defined a minimum skill set to be met by participants. The selection of experts was based on their strong professional experience and knowledge background of selected EAFs. To mitigate the impact of information exchange, the experiment took place in a controlled environment in which the participants were not allowed to communicate with each other.

External validity Threats to external validity might compromise the confidence to determine that the results of an experiment can be generalized. The primary external threat arises from the possibility that selected EAFs could be non-representative of other EAFs. To address this issue, we

Table 15 Usability

Definitions	*Usability* According to the ISO 9241-11 standard [43], usability is " the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". For ISO 9241-11 standard (see [43]): — *Effectiveness* the accuracy and completeness with which users achieve specified goals. For effectiveness, we use the success rate(SR) $SR = \frac{number\ of\ tasks\ completed\ successfully}{total\ number\ of\ tasks}\ X\ 100$ — *Efficiency* the resources (e.g., time, human effort, costs and materials) used in relation to the results achieved. For Efficiency, we use the average time (AT) taken by the architect to successfully complete the EA work $AT = \frac{standard\ time}{amount\ of\ time}\ X\ 100$ — *Satisfaction* the extent to which the user experience that results from actual use meets the user's needs and expectations. For satisfaction, we use the Single Ease Question (SEQ) metric. SEQ is a 7-point rating scale (from 'very difficult' to 'very Easy') to assess how difficult architect find an EAF
Overarching assessment logic	An EA framework is satisfactory if: $SR >= 70\ and\ AT >= 70\ and\ SEQ >= 5$
Metrics and operationalization	
Value	
0	$SR < 70\ and\ AT < 70\ and\ SEQ < 5$
1	$SR >= 70\ and\ (AT < 70\ or\ SEQ < 5)$
2	$SR >= 70\ and\ AT >= 70\ and\ SEQ >= 5$
3	$SR >= 90\ and\ AT >= 90\ and\ SEQ >= 5$

Table 16 Proposed evaluation criteria

Id	Criterion
C1	Taxonomy-architecture layers
C2	Taxonomy-architecture aspects
C3	Metamodel-completeness
C4	Metamodel-complexity
C5	SOA models
C6	Reference models
C7	Development process
C8	Governance process
C9	Supporting software
C10	Principles
C11	Architecture practice guidelines
C12	Availability of free information and software
C13	Adaptability
C14	Usability

analyzed 12 EAFs and selected the six that were the most used in the industry and literature. Nonetheless, our results might be valid only for the experimental EAFs, and further replications are needed to improve the generalizability of the results.

Construct validity Construct validity refers to the extent to which the observations or measurement tools actually represent or measure the construct being investigated. In this paper, one possible threat to construct validity arises from the metrics used to compute the criteria, especially

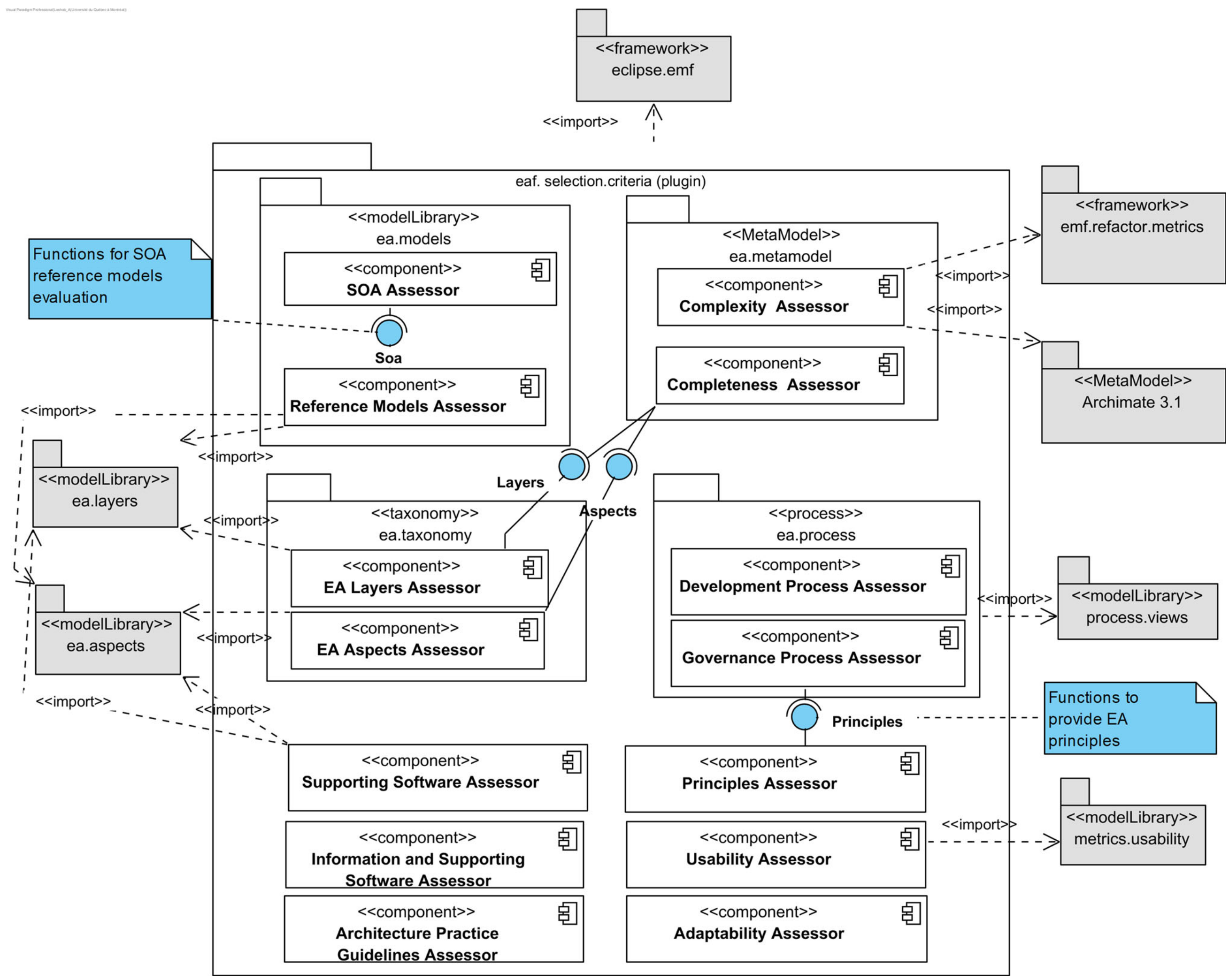

Fig. 1 Architecture of the proposed artifact

Table 17 Profile of the experts

Participant	Profile
Expert 1	
	—More than 30 years of experience in IT and EA
	—EA collaborator with researchers from the Carnegie Mellon and New York Universities
	—Extensive experience with business process management (BPM) and SOA
	—Participated in the EAF selection process of several organizations
Expert 2	
	—More than 20 years of experience in IT and EA
	—Extensive experience with software architecture and object-oriented design
	—Worked with large, medium and small organizations
	—Participated in the EAF selection process of several organizations
Expert 3	
	—More than 30 years of experience in IT, EA and Business Architecture
	—Solid SOA expertise
	—Participated in the EAF selection process of several organizations

Table 18 Selected enterprise architecture frameworks

Id	EA Frameworks	Type
F1	Zachman	Proprietary
F2	TOGAF	Consortia-developed
F3	FEAF	Government
F4	EAP	Proprietary
F5	The enterprise architecture IT Project	Proprietary
F6	DoDAF	Government

Table 19 Evaluation of the relevance criterion

Questions	Expert 1						Expert 2						Expert 3					
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6
Q1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q3	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
Q4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 20 Evaluation of the usability criterion

Questions	Expert 1						Expert 2						Expert 3					
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6
Q15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q18	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Q19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q28	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1

Table 21 Evaluation of the correctness criterion

Questions	Expert 1						Expert 2						Expert 3					
	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6	F1	F2	F3	F4	F5	F6
Q29	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1
Q30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q31	0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1
Q32	**1**	**1**	**1**1	**1**	**1**	**1**	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**
Q33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q35	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q36	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q37	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q38	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q42	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**

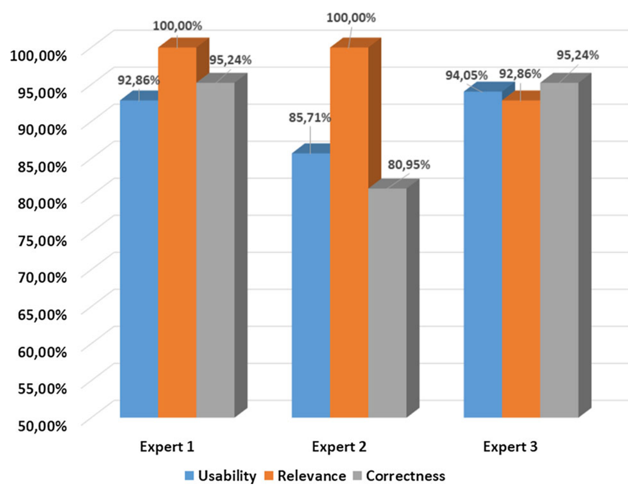

Fig. 2 Results of the evaluation

the metrics for metamodel complexity and EAF usability as raised by Expert 2. Therefore, conclusions obtained from our correctness evaluation might not be representative of other evaluation methods. To mitigate this concern, mature measurement techniques were used when available.

4 Conclusion and future work

Following the DSR approach, this study designed and tested an EAF evaluation artifact that identifies, elaborates and operationalizes a comprehensive set of 14 criteria. Overall, results of the experiment show that: (i) 90.87% of the criteria were perceived usable, (ii) 97.62% of them were perceived to be relevant, and (iii) 90.48% of them were perceived as correct.

This study contributes to the EA literature in several ways. First, through our review of the literature on EAF evaluation criteria, we present a much-needed and timely overview of the literature on this topic. Indeed, given the growing number of EA studies and the disparity of the EA literature, it was important to conduct such a review to induce and summarize past contributions. Most importantly, our review of the literature allowed us to identify key gaps that could explain why EA practitioners still fail to identify the EAF that best addresses the needs of their organization. Our literature review is thus similar but complementary to other EA-related literature reviews [49–52] that have brought order and meaning to studies on other aspects of the EA approach (e.g., benefits, methodologies, IT alignment).

Second, through the development of our EAF evaluation artifact, we answer EA practitioners' requests for a tool that is both theoretically sound and practical. Indeed, while previous studies did present evaluation criteria, our artifact is the first to present a comprehensive set of criteria that not only synthesizes already proposed criteria but also includes new criteria (e.g., SOA models and usability). It is important to mention here that the development of these new criteria was made possible by reviewing other research streams complementary to the one on EA evaluation criteria.

Finally, and most importantly, our artifact is the first to operationalize each of its criteria via a set of objective measures and scales. Indeed, in contrast to previous EAF evaluation studies and available EAF comparison matrices, our measures and scales do not rank EAFs based on subjective assessments of key experts. Instead, our measures and scales rank EAFs based on their objective and tangible characteristics. As such, by using our artifact, EA practitioners can now make their own objective assessment of candidate EAFs. They can thus make a decision that is informed by their own needs and context rather than by the subjective and out of context assessments of unknown experts. Thus, our artifact, by giving EA practitioners much-needed autonomy, goes one step further than the ones proposed in previous studies.

While we believe that our artifact handles the most important criteria to effectively select an EAF, for our tool to be functionally useful and usable, more research will be needed to (i) develop a web tool that supports the enterprise architects in measuring the criteria, (ii) refine certain metrics by taking into account the comments of the experts, especially for the metamodel complexity and usability criteria, and (iii) extend the set of criteria to support the concepts of business architecture as described in [3].

Acknowledgements The authors would like to thank all the experts for their involvement in this research. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Hadaya P, Gagnon B (2019) Succeed in your organizational transformations using a business architecture based framework. Technical report, ASATE Group, Montreal
2. Tallon PP, Pinsonneault A (2011) Competing perspectives on the link between strategic information technology alignment and organizational agility: insights from a mediation model. MIS Q 35(2):463–486
3. Hadaya P, Gagnon B (2017) Business architecture: the missing link in strategy formulation. Implementation and Execution, ASATE Group
4. The Open Group (2011) TOGAF Version 9.1
5. Lankhorst M (2009) Enterprise architecture at work: modelling, communication and analysis. Springer, Berlin
6. Winter R, Fischer R (2006) Essential layers, artifacts, and dependencies of enterprise architecture. In: 2006 10th IEEE international enterprise distributed object computing conference workshops (EDOCW'06), p. 30
7. Sessions R (2007) A comparison of the top four enterprise-architecture methodologies. Microsoft Developer Network Architecture Center, pp. 1–31
8. Urbaczewski L, Mrdalj S (2006) A comparison of enterprise architecture frameworks. Inf Syst J 7(2):18–23
9. Chen D, Doumeingts G, Vernadat F (2008) Architectures for enterprise integration and interoperability: past, present and future. Comput Ind 59(7):647–659
10. Cameron BH, McMillan E (2013) Analyzing the current trends in enterprise architecture frameworks. J Enterp archit 9(1):60–71
11. Hadaya P, Matyas-Balassy I, Marchildon P (2019) Enterprise architecture framework selection criteria: a literature review. In: Proceedings of the ninth international conference on business intelligence and technology, (Venice), p. 8
12. Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q Manag Inf Syst 28(1):75–105
13. Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24(3):45–77
14. Rumrill PD, Fitzgerald SM, Merchant WR (2010) Using scoping literature reviews as a means of understanding and interpreting existing literature. Work (Reading, Mass.) 35(3):399–404
15. Arksey H, O'Malley L (2005) Scoping studies: towards a methodological framework. Int J Soc Res Methodol 8(1):19–32
16. Abdallah S, Galal-Edeen G (2006) Towards a framework for enterprise architecture frameworks comparison and selection. In: The fourth international conference on informatics and systems
17. Franke U, Hook D, Konig J, Lagerstrom R, Narman P, Ullberg J, Gustafsson P, Ekstedt M (2009) EAF2-A framework for categorizing enterprise architecture frameworks. In: 10th ACIS international conference on software engineering, artificial intelligences, networking and parallel/distributed computing, (Daegu, South Korea), pp. 327–332, IEEE
18. Jonkers H, Lankhorst MM, ter Doest HWL, Arbab F, Bosma H, Wieringa RJ (2006) Enterprise architecture: management tool and blueprint for the organisation. Inf Syst Front 8(2):63
19. Kaisler SH, Armour F, Valivullah M (2005)"Enterprise architecting: Critical problems. In: Proceedings of the 38th annual Hawaii international conference on system sciences, (Big Island, HI, USA), pp. 224b–224b, IEEE
20. Leist S, Zellner G (2006) Evaluation of current architecture frameworks. In: Proceedings of the 2006 ACM symposium on applied computing, pp. 1546–1553, ACM
21. Lim N, Lee T-g, Park S-g (2009) A comparative analysis of enterprise architecture frameworks based on EA quality attributes. In: 10th ACIS international conference on software engineering, artificial intelligences, networking and parallel/distributed computing, pp. 283–288, IEEE
22. Rouhani BD, Mahrin MN, Nikpay F, Nikfard P (2013) A comparison enterprise architecture implementation methodologies. In: 2013 international conference on informatics and creative multimedia, pp. 1–6, IEEE
23. Tang A, Han J, Chen P (2004) A comparative analysis of architecture frameworks. In: 11th Asia-pacific software engineering conference, (Busan, South Korea), pp. 640–647, IEEE
24. Braun C, Winter R (2005) A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In: International workshop on enterprise modelling and information systems architectures, pp. 64–79, Ges. für Informatik
25. Winter R, Aier S (2011) How are enterprise architecture design principles used?. In: IEEE 15th international enterprise distributed object computing conference workshops, (Helsinki), pp. 314–321, IEEE
26. Boh WF, Yellin D (2006) Using enterprise architecture standards in managing information technology. J Manag Inf Syst 23(3):163–207
27. Winter R, Schelp J (2008) Enterprise architecture governance: the need for a business-to-IT approach. In: Proceedings of the 2008 ACM symposium on applied computing, (Fortaleza, Brazil), pp. 548–552
28. Sobczak A (2013) Methods of the assessment of enterprise architecture practice maturity in an organization. In: International conference on business informatics research, pp. 104–111, Springer
29. Bernaert M, Poels G, Snoeck M, De Backer M (2014) Enterprise architecture for small and medium-sized enterprises: a starting point for bringing EA to SMES, based on adoption models. In: Devos J, van Landeghem H, Deschoolmeester D (eds) Information systems for small and medium-sized enterprises. Progress in IS. Springer, Berlin
30. Gregor S, Hevner AR (2013) Positioning and presenting design science research for maximum impact. MIS Q 37(2):337–355

31. Steen MWA, Akehurst DH, Doest HWLter, Lankhorst MM (2004) Supporting viewpoint-oriented enterprise architecture. In: Proceedings. Eighth IEEE international enterprise distributed object computing conference, 2004. EDOC 2004., (Monterey, CA, USA), pp. 201–211, IEEE
32. The Open Group (2019) ArchiMate 3.1 Specification
33. Sowa J, Zachman J (1992) Extending and formalizing the framework for information systems architecture. IBM Syst J 31(3):590–616
34. Object Management Group (OMG) (2016) OMG meta object facility (MOF) core specification
35. Genero M, Piattini M, Calero C (2002) Empirical validation of class diagram metrics. In: ISESE 2002- proceedings, 2002 international symposium on empirical software engineering
36. Object Management Group (OMG) (2012) Service oriented architecture modeling language (SoaML) Core Specification
37. Blal R, Leshob A (2017) A Model-driven service specification approach from BPMN models. In: Proceedings-14th IEEE international conference on e-business engineering, ICEBE
38. Blal R, Leshob A, Gonzalez-Huerta J, Mili H, Boubaker A (2018) From inter-organizational business process models to service-oriented architecture models. SOCA 12(3):227–245
39. Schekkerman J (2004) How to survive in the jungle of enterprise architecture frameworks: creating or choosing an enterprise architecture framework. Trafford, Bloomington
40. Curtis B, Kellner MI, Over J (1992) Process modeling. Commun ACM-Special issue on analysis and modeling in software development 35:75–90
41. Kettinger WJ, Teng JTC, Guha S (1997) Business process change: a study of methodologies, techniques, and tools. MIS Q 21(1):55–98
42. Buckl S, Ernst AM, Lankes J, Schneider K, Schweda CM (2007) A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftinformatik Proceedings 2007, pp. 145–162
43. ISO (2018) Ergonomics of human–system interaction (ISO 9241-11)
44. Iivari J (2015) Distinguishing and contrasting two strategies for design science research. Eur J Inf Syst 24(1):107–115
45. Basili VR, Rombach HD (1988) Tame project: towards improvement-oriented software environments. IEEE Trans Softw Eng 14(6):758–773
46. Chidamber SR, Kemerer CF (1994) A metrics suite for object oriented design. IEEE Trans Softw Eng 20(6):476–493
47. Sauro J, Kindlund E (2005) A method to standardize usability metrics into a single score. In: CHI 2005: technology, safety, community: conference proceedings-conference on human factors in computing systems, (Portland, Oregon, USA), pp. 401–409, ACM
48. Cook TD, Campbell DT (1979) Quasi-experimentation: design and analysis issues for field settings. Houghton Mifflin, Boston
49. Gorkhali A, Xu LD (2017) Enterprise architecture: a literature review. J Ind Integr Manag 2(02):1750009
50. Plessius H, Slot R, Pruijt L (2012) On the categorization and measurability of enterprise architecture benefits with the enterprise architecture value framework. Lecture notes in business information processing. Springer, Berlin, pp 79–92
51. Rouhani BD, Mahrin MNZ, Nikpay F, Ahmad RB, Nikfard P (2015) A systematic literature review on enterprise architecture implementation methodologies. Inf Softw Technol 62(1–20):2015
52. Zhang M, Chen H, Luo A (2018) A systematic review of business-it alignment research with enterprise architecture. IEEE Access 6:18933–18944