International Journal of Electronic Engineering and Computer Science
Vol. 4, No. 1, 2019, pp. 1-9
http://www.aiscience.org/journal/ijeecs

Validation Methodologies for RAs: A Case Study of RAFAALS2.0

El Murabet Amina*, Abtoy Anouar, Tahiri Abderahim

Computer Engineering Department, National School of Applied Sciences of Tetuan, Tetuan, Morocco

Abstract

To affirm the completeness, the success and the validation of RAs (Reference Architectures), a chain of validation process should be taking into account in the end of each assessment. The need behind the validation reveals from the necessity of achieving accuracy between the different components of the system, as well as, achieving a high level of abstraction. In the result of achieving a unique version of the suggested architecture to be considered as a reference for multiple projects. The existing methodologies, which allow driving the process of validation, are rare and incomplete. In this paper, we present three different approaches. Each one of them is described and applied into the case study, with the objective of extracting the loopholes to value the strength of each methodology. Finally, we investigate the results and draw the feedback to be used in the enhancement and development of new validating approaches related to RAs in software engineering.

Keywords

Reference Architecture, Validation Methodologies, Software Engineering

Received: March 2, 2019 / Accepted: April 8, 2019 / Published online: April 29, 2019

1. Introduction

With all the efforts invested in creating RAs [1] in software engineering to be considered as a background for developing software systems in specific domains, it is important to evaluate the suggested architectures to make sure they capture all required features. Many questions are raised while choosing to design new systems relaying on existing infrastructures, in matter of investments on the adoption of a RA, once adopted, what could be the suitability of a RA for deriving concrete system architecture, does it answer to the standards of RAs? Would these visions serve as a standardized one or would it be only an inspiration element for designing concrete architecture or does it lack other elements to achieve completeness…etc. All these questions had driven the investigation presented in this paper.

It is clear that the main objective of using a RA is to provide a mutual background for the architects in a specified field [2]. If the RA can afford an understanding from different perspectives of the targeted field than the evaluation will be an easy task as it would concern the specifications of that scope [3-4].

The ambition of having an accepted and widely used RA should be closured by its validation. Which drives us to investigate the existing approaches with the ambition of verifying the quality level of RA. As a case study, we have chosen for this paper RAFAALS2.0 [5] as it presents a new vision of reference architectures. To do so, we will investigate the system in the context of the envisioned work processes as the questions of formalization and automation of the work processes can become subjects of further elaborations.

Generally, the correctness and the completeness of a RA is determined by a number of attributes, depending on the validation Framework [6]. Each one of the presented methodologies of validation embraces its own strategies and values to evaluate the design. We can judged that all of them concentrate on the homogeneity between the goal from

* Corresponding author
E-mail address: elmurabet.a@gmail.com (El M. Amina)

designing the RA and its capabilities of answering to the requirements of the system that embraces it. If the design allows meeting between the functionalities and the constraints than the design of the RA is highly approved. A formal evaluation is the architect's best hope to estimate the completeness of the work.

One of the critical success factors is the design process that should be embraced while constructing the RA is the respect of the different views [6]. A good RA must allow the concrete architectures to be complete and meet its requirements. The general design specification is also an important factor for detecting the completeness of the RA. If the concrete architectures are capable of meeting their objectives by representing the components of the system in a reliable and a multifunctional way, than the RA achieves a large scale of success.

The existence of different views of the RA is also a supportive element of its completeness. The various levels of abstraction gives a sophisticated infrastructure for building the concrete architectures [2]. In [7] authors suggested that the two elementary views that have to be presented by the RA are the structural and the behavioral ones as illustrated in Figure 1. The details over the components is also a mandatory descriptive way of the RA to achieve the complete analysis structure of the targeted field, which means validating a RA should start by approving the existence of these three views.

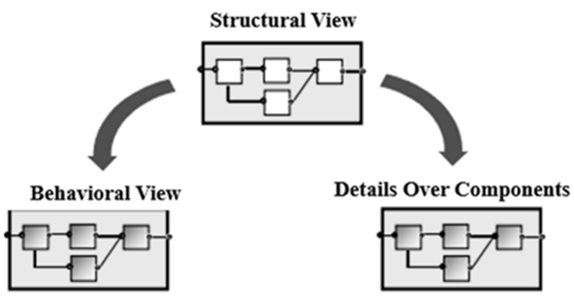

Figure 1. Different views that have to be supported by RAs [7].

In this paper, we first start by introducing the concept of the validation methodologies. In the second section, we described the Angelov's framework and we applied it to the case study RAFAALS2.0. In the third section, we introduced the Santos' Checklist to reveal its advantages and disadvantages in the process of validating a RA. The fourth section was devoted to the Graaf's Framework. Finally, we studied the problems facing the validation methodologies to be more mature and to offer reliable strategies of validation for designers and developers to ease their mission while proposing new approaches of RAs. At the end of this paper we conclude and we encourage new opportunities of developing mature strategies of validation.

2. Methodologies of Validation

The main objective of the validation methodologies disposed in the spectrum of software RAs is highlighting the weakness of the proposed designs in order to give a chance for designers to fix their approaches during the early development stages [2], which would cost less and prevent problems in the post development phase. The validation certainly detects the conflicts between the conception and the requirements [8, 9]. The validation approaches focus also on ensuring the enforcement of quality attributes and their derivation of constraints to make the RA adaptable to further emerging requirements.

Along in years the only methodology of validating RAs was done with the help of answering manually to some particular questions as in the case of Caracciolo [10]. Some current methods offer a list of factors such as Kumar's [6]. All with the aim of reducing the gap between the specifications and the design [11-12]. In the presented study, we figured that the uniformity of the design is a mandatory issue discussed by the verification tools.

Many frameworks adapt the idea of validating the RA as they become more and more popular over the years [13]. Designing software systems over reliable architectures gives power for more survival period for these systems [14]. The survival point is also related to the strength of the concrete architecture, as it should be built over solid RA. Each one of the methodologies proposed to validate the RA has its own nature. Some of these approaches require validation by frameworks while others propose checklists. In the presented paper, we studied ones from different natures to ensure a complete overview of the field.

For this purpose, we launched an investigation searching for the most reliable methodologies of validation. We used a literature overview of architecture evaluation methods in selecting the presented ones [9]. As of the extracted knowledge the Angelov's [15] framework for validating the design of RAs is a very reliable one according to [16]. To approve the validity of our approach we have chosen two other tools which are the Santos' Checklist [17] in addition to Graaf's Framework [9]. To ensure architectural compliance, regular checks are essential [6]. As a first overview, we judged that the complete process of validation takes in consideration the system design requirements, the general design specification, and the detailed design specification.

In the end of the processes, some approaches give an indicator of validation. The results may be: i) Validated: if the RA answers to all the constraints of the Framework of validation. ii) semi-validated: if the RA is appropriate to some criteria, thus it needs some modifications and adjustments to be complete. iii) non-validated: if the framework judged the RA as unqualified to be adapted. Some other frameworks give other indicators such as according a type to the RA as we will see in details in the validation by Angelov's Framework section.

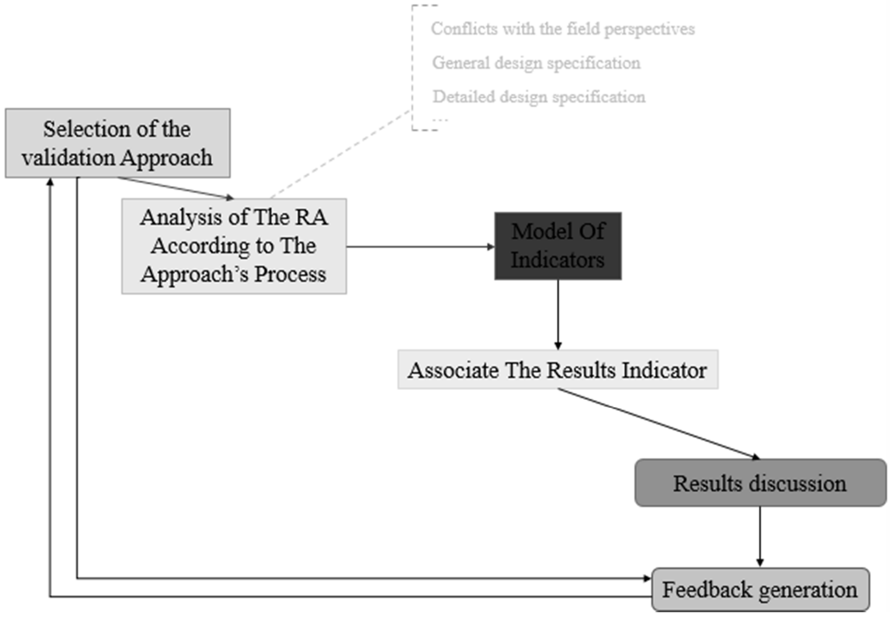

Figure 2. General overview of the validation processes.

In our vision, we believe that the general flow of validating RAs begins with the selection of the validation approach as illustrated in Figure 2. Then, according to the analysis chain of the approach we discuss any existing conflict, the general design specification, the detailed design specification…etc. and many other aspects. Each one of the validating methodologies has a model of indicators associated to the level of completeness and coherence of the evaluated RA. Results are discussed after the association of the indicators and feedback is generated.

To be able to detect the advantages and the disadvantages of the validation processes we embraced RAFAALS2.0 as a case study. In the following steps, we consider as an objective to validate this architecture according to each methodology. Finally, we draw conclusions to be used in further developments of validation frameworks as there is a lack of them on the conceptual field.

3. The Angelov's Framework

Angelov's Framework [15] is considered a reliable validating framewok for RAs [5]. It consists of multi-dimensional classification space and of five types of RAs, which are formed by combining specific values from multi-dimensional classification space. If a RA can be classified as one of the proposed types, it has a better chance of success. It uses analysis for the level of congruence of existing RAs. The framework has proven a success by validating more than 23 RA approved by the experts' opinions.

In the Angelov's perspective, the coherence between the context, the goal and the design is the vital element for a survival RA as each one of them affects the others. While validating, it is recommended to investigate these concepts and the respect of their relationships by the RA. If the goals of the RA are relevant for the context and its design properly reflects the architecture's context and goals than this RA is a congruent one according to it.

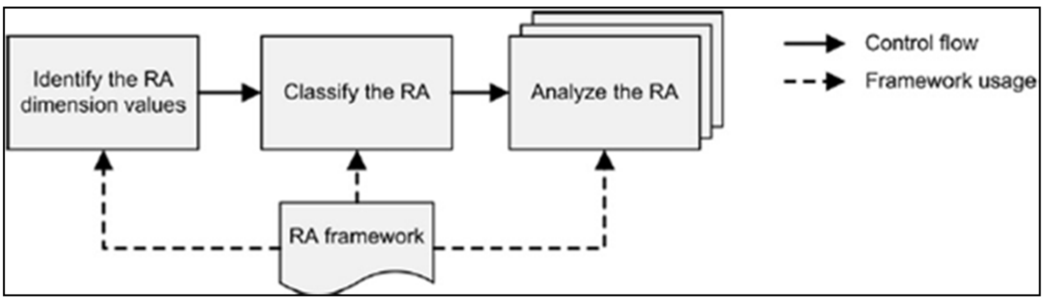

Figure 3. Angelov's Framework usage in the analysis of a RA.

As it is illustrated in Figure 3, Angelov's Framework usage process starts by identifying the RA dimension values. It classifies the RA based on several types and finally analyses the results. By starting with associating the dimensions values to the RA, we will use the same control flow.

3.1. The Identification of Dimension Values

Agelov defines a multi-dimensional space for the classification of RAs that supports the analysis in terms of relationship between three dimensions context, goal and design. Each of these dimensions relies on sub-dimensions that address aspects of the dimension.

Context dimension: represents aspects of the context that may affect the goals and design of the RA. Its sub-dimensions are Stakeholders, Development Team and Timing. For each one of these sub-dimension we associate a value.

Stakeholders: RAFAALS2.0 has described its stakeholders by being any contributors of AAL systems' design and implementation. In the process of constructing the RA, we offered its benefits for any organization that might find it useful to be adapted. Intended recipients of RAAFALS2.0 are architects, engineers and manufacturers searching for a background and formal template to build their designs on. As a result, our RA is intended to serve multiple organizations that share the property of constructing AAL systems. We can then associate to this sub-dimension the value ''multiple organization''.

1. Development Team: in the process of designing the RA, software designers, software users, software researchers were involved. We can overcome with the value of user as organizations that were responsible of RAFAALS2.0 construction.

2. Timing: RAFAALS2.0 was designed after the existence of commercial systems and after the experience that have been accumulated from previous systems. By the time of design multiple technologies, software and algorithms related to AAL systems exists and have been tested in practice. The team has considered these experiences while designing this approach. As a result, we can assume that RAFAALS2.0 is a classical RA.

3. Goal dimension: Angelov defines the possible goals of the RA by this dimension. There in only one-sub-dimension defined that represents the general goal of a RA that has to be aligned with the architecture context and design.

4. Goal: RAFAAALS2.0 approach has as goal to standardize the concrete system architectures aiming to accomplish the system/component interoperability. As a result, we can associate the value: Standardization to its goal.

5. Design dimension: in this perspective the framework evaluates the RA in terms of its operational side (design and specification). The sub-dimensions addresses the content of the RA, its level of detail, its correctness and the techniques used for presentation.

6. Description: RAFAALS2.0 lists all the elements that can be defined in a RA. it describes the components and the connectors, the interfaces, the policies as well as the guidelines for implementation which are referred to by texture. It associates the value: Components/Connectors/guidelines to this sub-dimension.

7. Level of details: there are three levels of details associated with Angelov's Framework. detailed, semi-detailed, and aggregated specification of the elements of a RA. These detail levels are believed complex to judged. RAFAALS2.0 is detailed in many levels. The team described even some details of the depth functioning of some elements such as the Situation Reasoner. We assume that linking the details to this sub-dimension seem to be convenient.

8. Concreteness: it is related to multiple levels of abstraction. In term of technology, applications, vendors…etc. RAFAALS2.0 approach presents a high level of abstraction as it is very generally specified. The software components are data management module that is why we can assign the value: Abstraction to this sub-domain.

9. Presentation: the levels of formalization are described as informal, semi-formal and formal. As the RAFAALS2.0 presentation is based on formal, well-defined notations such as concept maps, components and connectors …etc. it gives a reasonable compromise between the clarity and precision. We believe that it is a semi-formal presentation.

As a conclusion we summaries the values of each dimension in:

a. Context: stakeholder: multiple-organization

b. Development Team: User organizations

c. Timing: classical reference architectures

d. Goal: Standardization

e. Design: Description: Components/connectors/guidelines

f. Level of details: detailed

g. Concreteness: Abstract

h. Presentation: Semi-formal

3.2. Classifying the RA

Angelov's Framework identifies a number of types of RAs by combining the values of the sub-dimensions that should satisfy a number of constraints. The basic combination of values are used to define a type of variations. For the sake of classifying RAFAALS2.0 we will use a structured table where we can associate values to dimensions and then to the defined types of the Framework.

Table 1. Classification of RAFAARLS2.0 according to Angelov's types.

Dimension	Sub-Dimensions	Values	Type1	Type2	Type3	Type4	Type5
Context	Stakeholder	Multiple-Organization	x		x		x
		Single Organization		x		x	
	Development Team	Standardization-Organization					x
		Software Organizations		x	x		x
		User Organizations	x				
		Software design groups				x	
	Timing	Classical RA	x	x	x	x	
		Preliminary					x
Goal	Goal	Standardization	x	x			
		Facilitation			x	x	x
Design	Description	Components	x	x	x	x	x
		Connectors	x		x	x	
		Guidelines	x	x		x	
		Algorithms					x
	Level Of Details	Detailed	x			x	
		Semi-Detailed		x	x		x
		Aggregated					
	Concreteness	Abstract	x				x
		Concrete					
		Semi-Concrete		x	x	x	
	Presentation	Formal					x
		Semi-Formal	x		x	x	
		Informal		x			

In Table 1, we presented all the suggested dimensions, sub-dimensions and their values proposed by Angelov's Framework. The intersection between the dimension values and the values presented in this table came up with a full match with type1 RA.

According to the Framework, RAFAALS2.0 is a classical, standardization architecture, design to be implemented by multiple organizations.

4. Validation Using Santos' Checklist

Santos' [17] has introduced a checklist approach for the evaluation of RAs. It helps to uncover defects in an early stage of the RA life cycle. This approach was evaluated by experts in software architectures and was successfully applied [18]. The construction of this checklist was driven by the literature available of RA and software architectures. It incorporated the characteristics and the required elements.

The checklist has demonstrated to be an effective approach to catch defects in the early stages of designing RAs [17]. Experts consider this checklist as adequate for use. It is flexible and adaptable to many RA domains [6].

The authors recommend the checklist application during the elaboration of the RA where it can catch defects while the modification is cheaper than in a last stage of its life cycle. Nevertheless, it is highly effective to use it during the final evaluation as illustrated in Figure 4, which is our case.

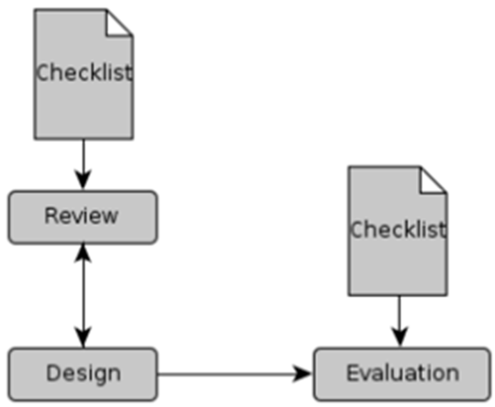

Figure 4. The checklist application points in a general life cycle.

The checklist proposed a four-stage structure of evaluation. Starting by general Information, Raising discussion, general analysis and finally the domain specifics. As it is illustrated in Table 2, the first stage consists of two sets with 9 and 16 questions sequentially. In this stage, the evaluation focuses on the views, the models, the stakeholders and the special concerns of the evaluated RA.

Stage 2 has one set including 52 question related to the implementation issue. In stage 3, the checklist analyses some general elements. At the fourth and last stage is divided into two sets evaluating the hardware and the software elements.

Table 2. Subjects and number of question in each stage.

Stage	Set	Questions	Subjects covered
1	1	9	Overview information, views, models, stakeholders and concerns
	2	16	Design and development issues, domain specific issues, quality attributes
2	1	52	Specific questions over the implementation
3	1	3	Conclusion of the general analysis
4	1	11	Embedded hardware
	2	3	Embedded software

We will answer these questions to validate the case study: RAFAALS2.0. The checklist presented in next table is elaborated using Factors and associated values. Three possibilities are presented:

‘’ ✓” Confirmed, if the factor is supported by the RA that is in the process of evaluation.

‘’+” Need Improvements, it indicates the factors that need some further improvements to achieve a satisfactory level.

‘’x” Missing, if the factor is not supported by the RA than we assume it is absent. Here comes the strength of validating the RA.

If the checklist confirms the presence of 80% of the factors than the RA is considered a strong RA from Santos' viewpoint. Between 60% and 80% is considered a reliable one with a need of some modifications. Less than that, is associated with poor successes of the proposed RA.

We achieved 82% of the factors required by Santos' checklist while this study was droven. Nevertheless, this validation process has opened many other motivations for us to select and improve the quality of the validating processes. Some of the needed improvements are highly recommended to accomplish a higher level of maturity. However, the missing factors highlighted some new guidance to cover the absent elements that did not manifest to our investigations' processes.

This validation process has accumulated a new experience of evaluating the factors presented by the RA. We could clearly see the benefits out of comparing the prescribed components and the factors imposed by the checklist to validated the RA. We will certainly be influenced by these factors for our further improvements.

5. Validation Using Graaf's Framework

Graaf's Framework [9] has his own perspective on evaluating RAs. It believes that calculating the costs and benefits out of adopting a RA can reflect the level of its maturity. The framework uses components to calculate the costs and software metrics to examine the benefits. On one hand, the number of components required by the RA allows the estimation of costs of the implemented architectures. On the other hand, metrics help analyzing the benefits that can be found in the RA.

For this purpose, Graaf's Framework introduce a chain of phases to complete the validation procedure presented in Figure 2. We will pursue the chain of phases to validated RAFAALS2.0. As we believe this verification procedure is different from the previous ones, we assume that we will gather further information in favor of developing new validation processes.

5.1. Associating Benefits to the Metrics for Classification

The first stage of the process consists of describing the case study as illustrated in Figure 5. We consider the previous descriptions present in previous sections. In this step, we will directly move to the association between Metrics and benefits. The first recommendation is to start by describing the RA, its objectives, characteristics, the major design…etc. The description is followed by extracting the benefits of this RA. Then comes the phase of associating the benefits to the metrics and classifying the RA according to the classification model associated with the framework. Extraction of the results and drawing feedbacks are the last steps.

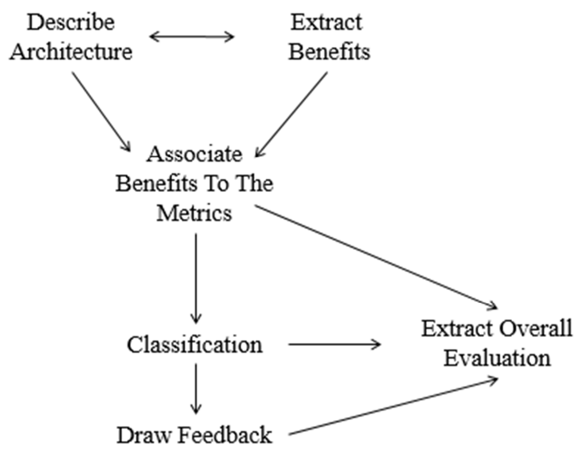

Figure 5. Graaf's procedure of evaluating RAs.

The overall evaluation resulted in a set of strong and weak points. The set of strong points includes the aforementioned use of the RA and its flexibility. Most of the evaluated metrics have to be supported. The set of weak points includes a design flow that prevents support for multiple views, which is required for strengthening the understanding of the RA.

The RA is considered incomplete if it misses the details over its major components. Another weakness may be revealed form the fact that variations in design are not allowed. Missing the structural design raises the weak points. Conformance to the domain and the deviations from it are also considered. Maintainability issues can raise the strong points to a very high-level as it is one of the most required benefits.

5.2. Overall Evaluation: Set of Strong and Weak Points

In this step, the framework suggested evaluating the strong and the weak points of the RA. The strong points are related to the flexibility as well as the high support of the metrics presented previously. As we can extract from the earlier phase RAFAALS2.0 nearly answers to the major metrics presented.

Therefore, evaluating the weak points is a bit harder as the RA is still in its very early stage of the life cycle. The set of weak points includes the design flow, details of the major components, allowing variation, the structural design and conformance to the domain of concretization. To evaluate them, we will discuss the presence or the absence of each one of them.

Starting by the design flow, a weakness can reveal from the unique view offered of the RA's design. In the contrary, RAFAALS2.0 has been described using multiple views. The overall design was the first presentational element presented in the process of design construction, followed by the structural and the behavioral views. The case study presented the context-awareness extraction view and detailed the goal orientation process. We can assume that this number of views is enough to prevent this weakness.

Details over the major elements are produced whenever we estimated a high demand of illustrations. The architecture describe the essential elements with depth and precision such as in the case of the Situation Reasoner. We cannot be conclusive in the matter of details. As some people might judged the level of details as a complete one where another might find difficulties in understanding the behavior and the nature of a component.

The RA presents a standardized design, aiming of working as a template for upcoming concrete architectures oriented to the AAL domain. As it is a template, the design can be adapted according to the needs, requirements, nature and investments offered to create the system. These adjustments in matter of requirements will affect the overall design, as it is context-aware and goal-oriented. As a result variation in the design might appear. Thus, these variations should not overcome the major concepts nor ignore the essential components.

The RA is targeted to the AAL domain. It was built with a dedication to support the implementation of concrete architectures of AAL systems. It derived investigations, comparisons, reviews and studies over this domain to come up with its challenges and its scalability issues and develop a new approach to overcome its limitation and enhance its qualities. The RAFAALS2.0 is certainly conformed to the domain of implementation.

To summaries, the strong points of the RA have overcome the number of the weakness we found in the design. The majority of the metrics were associated with according benefits. As a result, we believe that this validation process was in favor of the ‘’ quality ‘’ of RAFAALS2.0.

The RAFAALS2.0 approach shows no conflict between the existing elements of the AAL environment, its classification nor its essential pillars. It assists the understanding of the field and influences the suitability of the concrete architecture to the predefined requirements. Moreover, it eases the communication between concrete architecture's designer and developers of the end-system. By establishing the rules of our approach, developers can maintain compliance with the system's requirements during the implementation process.

6. Discussion of the Validation Methodologies

We faced several problems when selecting the frameworks of validation as we met with papers full of analysis and no means of validation. We believe that this area of research lacks a lot of work to enhance the quality and ensure the yield from these processes. No clear procedures were offered along with these attempts.

According to the review over the validation tools that we adopted in our selection [10], the three methodologies we present in this paper, are the most effective ones. As you may notice, there are some similarities between them in the matter of the attributes of validation.

The validation frameworks have different processes to analyze the RA. Angelov uses dimensions, sub-dimensions and associated values, where the checklist concerns factors and the Graaf's framework uses the metrics in which we associate the benefits. All of these methods were valuable to our studies as we extracted many advantages of them. We could examine the strong points of the case study and we had the opportunity to know the weak ones to be considered in future contributions.

RAFAALS2.0 passes the three methods of validation, as it has many benefits. It reduces the development time and faster the delivery of the end systems. It increases the productivity as it offers abstract background useful to start with. The RA introduces a documentation of the field; by this mean, it provides a useful path for the non-specialists. For the system's builders it works like guidelines. Complexity is reduced using the RA as it solved the functionality part. Finally, the integration mission is easier with the legacy systems and the external ones. Most important of them all is cost reduction, in the process of creation as well as in the maintenance one. Furthermore, both architectural knowledge and common elements benefit the reduction of expenses.

Homogenizing the systems developed in the AAL domain is the main concern. We believe that by using more effective validating processes we can increase the control over the suggested designs and improve the quality of the implemented systems through the homogeneity between the backgrounds and the real time designs. The RAs allow adding or changing functionalities using the described components. The risk will also be reduced as the experience will show. Many quality attributes are promoted by the RAs. The harvested knowledge improves the success as well as the survival of the architecture.

Several benefits are accumulated from driving the validation processes. Perusing the chains will reduce complexity by systematically reducing the diversity when dealing with introducing new visions of RAs. It will greatly increase the speed of contributions in the field as it reduces the operational expenses and gives a solid background to start. Encouraging the validation will enhance the productions of new RAs that will serve as backgrounds to lower the complexity, as well as, increase the investments and encourage the reuse.

The validation offers a new opportunity to conduct construction of mature versions of RAs where the designers can overcome all the missing points and update their RA. The need of reliable validation methodologies in the field of software developments is an emergency and a necessity to increase the quality and insure the coherence between the designs and the products.

7. Conclusion

In this paper, we conducted three different methodologies to validate a RA case study introduced by RAFAALS2.0. Each one of them was illustrated, described and applied separately with the objective of highlighting its advantages and disadvantages with the ambition of using these loopholes and strengths in the development of new validating approaches for RAs. Furthermore, we highlighted the weak points and the missing metrics to be enhanced in future versions. We believe that the validation of a proposed RA opens a new opportunity for designers and developers to work over the missing points that might influence the survival of their approaches in long terms.

References

[1] P. B. Hongyu, "A Survey and Analysis of Reference Architectures for the Internet-of-things," presented at the The Twelfth International Conference on Software Engineering Advances, Västerås, Sweden, 2017.

[2] S. Martínez-Fernández, C. P. Ayala, X. Franch, and H. M. Marques, "Benefits and drawbacks of software reference architectures: A case study," *Information and Software Technology*, vol. 88, pp. 37–52, Aug. 2017.

[3] E. Y. Nakagawa, M. Guessi, J. C. Maldonado, D. Feitosa, and F. Oquendo, "Consolidating a Process for the Design, Representation, and Evaluation of Reference Architectures," in *2014 IEEE/IFIP Conference on Software Architecture*, 2014, pp. 143–152.

[4] P. O. Antonino, D. Schneider, C. Hofmann, and E. Y. Nakagawa, "Evaluation of AAL Platforms According to Architecture-Based Quality Attributes," in *Ambient Intelligence*, Springer, Berlin, Heidelberg, 2011, pp. 264–274.

[5] A. El murabet, A. Abtoy, and A. Tahiri, "A Context-aware Reference Architecture for Ambient Assisted Living Information Systems," *MECS press*, vol. 11, no. 2, pp. 9–20, Feb. 2019.

[6] N. Kumar, "Software Architecture Validation Methods, Tools Support and Case Studies," in *Emerging Research in Computing, Information, Communication and Applications*, Springer, 2016, pp. 335–345.

[7] J. Magee, J. Kramer, and D. Giannakopoulou, "Behaviour analysis of software architectures," in *Software Architecture*, Springer, 1999, pp. 35–49.

[8] B. Graaf, H. van Dijk, and A. van Deursen, "Evaluating an Embedded Software Reference Architecture," in *Ninth European Conference on Software Maintenance and Reengineering*, 2005, pp. 354–363.

[9] A. Holzinger, G. Searle, A. Auinger, and M. Ziefle, "Informatics as Semiotics Engineering: Lessons Learned from Design, Development and Evaluation of Ambient Assisted Living Applications for Elderly People," in *Universal Access in Human-Computer Interaction. Context Diversity*, 2011, pp. 183–192.

[10] A. Caracciolo, M. F. Lungu, and O. Nierstrasz, "How do software architects specify and validate quality requirements?," presented at the European Conference on Software Architecture, 2014, pp. 374–389.

[11] R. Allen and D. Garlan, "A Formal Basis for Architectural Connection," *ACM Trans. Softw. Eng. Methodol.*, vol. 6, no. 3, pp. 213–249, Jul. 1997.

[12] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, "Software Quality Requirements: A Systematic Mapping Study," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, 2013, vol. 1, pp. 231–238.

[13] D. Salvi, M. Colomer, J. Bautista, M. T. Arredondo, B. Prazak-Aram, and C. Mayer, "A framework for evaluating Ambient Assisted Living technologies and the experience," *Journal of Ambient Intelligence and Smart Environments*, vol. 7, no. 3, pp. 329–352, Jan. 2015.

[14] C. A. Gunter, E. L. Gunter, M. Jackson, and P. Zave, "A reference model for requirements and specifications," *IEEE Software*, vol. 17, no. 3, pp. 37–43, May 2000.

[15] S. Angelov, P. Grefen, and D. Greefhorst, "A framework for analysis and design of software reference architectures," *Information and Software Technology*, vol. 54, no. 4, pp. 417–431, Apr. 2012.

[16] A. El murabet, A. Abtoy, A. Touhafi, and A. Tahiri, "Ambient Assisted living system's models and architectures: A survey of the state of the art," *Journal of King Saud University - Computer and Information Sciences*, Apr. 2018.

[17] J. F.. Santos, M. Guessi, M. Galster, D. Feitosa, and E. Nakagawa, "A checklist for evaluation of reference architectures of embedded systems," presented at the Proceedings of the International Conference on Software Engineering and Knowledge Engineering, 2013, pp. 451–454.

[18] L. Garcés, A. Ampatzoglou, F. Oquendo, and E. Yumi Nakagawa, "Reference architectures and reference models for ambient assisted living systems: a systematic literature review," University of São Paulo (USP), Tech. Rep. 414, Feb. 2017.