SOAQE
Service Oriented Architecture Quality Evaluation

Riad Belkhatir[1], Mourad Oussalah[1] and Arnaud Viguier[2]
[1]*Department of Computing, University of Nantes, Rue de la Houssinière, Nantes, France*
[2]*BeOtic, Rezé, France*

Keywords: SOA, Quality, Evaluation, Attributes.

Abstract: This paper presents a semi-automated method for evaluating SOAs called SOAQE, correcting defects observed so far with existing methods such as lacks of pertinence and accuracy for evaluation results. SOAQE takes as a starting point the McCall model, describing software quality, which led to an international standard for the evaluation of software quality (ISO/IEC 9126-1, 2001). This model is organized around three types of quality attributes (factors, criteria and metrics). The SOAQE method consists in decomposing the whole architecture and evaluating it according to the McCall model, i.e. a list of quality factors arising from business needs grouping criteria composed by metrics. Our experimentations led us to quantify numerically a first determining factor for SOAs, the 'dynamism' and some attributes of its structure: namely the 'loose coupling' criterion and its constituent metrics ('physical, semantic and syntactic').

1 INTRODUCTION

We can distinguish a few architectural paradigms for distributed systems, and, among the most noteworthy ones, three have contributed to the evolution of the concerns. These are chronologically, object oriented architectures (OOA), component based architectures (CBA) and s**ervice oriented architectures (SOA).**

First developers were quickly aware of code repetitions in applications and sought to define mechanisms limiting these repetitions. **OOA** is focused on this concern and its development is one of the achievements of this research. OOA provides great control of the **reusability** (*reusing a system the same way or through a certain number of modifications*) which paved the way to applications more and more complex and consequently to the identification of new limits in terms of granularity. These limits have led to the shift of the concerns towards the **composability** (*combining in a sure way its architectural elements in order to build new systems or composite architectural elements*). Correlatively, the software engineering community developed and introduced **CBA** to overcome this new challenge and thus, the CBA reinforces control of the composability and clearly formalizes the associated processes. By extension, this formalization establishes the base necessary to automation possibilities. At the same time, a part of the software community took the research in a new direction: the **dynamism** concern (*developing applications able to adapt in a dynamic, automatic and autonomous ways their behaviors to answer the changing needs of requirements and contexts as well as possibilities of errors*) as the predominant aspect. In short, **SOA** has been developed on the basis of the experience gained by objects and components, with a focalization from the outset on ways of improving the **dynamism.**

The SOAQE method presented in the present paper allows evaluating SOA by combining the computerized approach and the human intervention to eliminate lacks identified with past methods such as ATAM, SAAM or ARID (Clements,Kazman and Kein, 2001). The main idea resides in automating the process to avoid time-wasting evaluations. The process consists in three principal stages detailed later in this paper. In the current paper, we first relate in the section 2 the interests of the evaluation, we analyze the existing works and we present the SOAQE method in the section 3; and finally we relate the experimentation which has been done by the lab team in the section 4. We conclude this paper

Belkhatir R., Oussalah M. and Viguier A..
SOAQE - Service Oriented Architecture Quality Evaluation.
DOI: 10.5220/0003997701990202
In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2012), pages 199-202
ISBN: 978-989-8565-13-6

with a conclusion about the perspectives of our work.

2 MEASURING THE QUALITY

The scientific community utilizes models of quality introducing what we call the software attributes decomposition.

2.1 Mc Call Model

One of the models that have been published is the McCall model in 1977 (*see figure 1*) decomposing quality attributes in three stages. This model led to the IEEE standard*: ISO/IEC 9126* (ISO/IEC 9126-1, 2001). A certain number of attributes, called external (*applicable to running software*), are considered as key attributes for quality. We call them **quality factors** (Clements,Kazman and Kein, 2001). These attributes are decomposed in lower level attributes, the internal attributes (*which do not rely on software execution*), called **quality criteria** and each criterion is associated to a set of attributes directly measurable and which are called **quality metrics**.

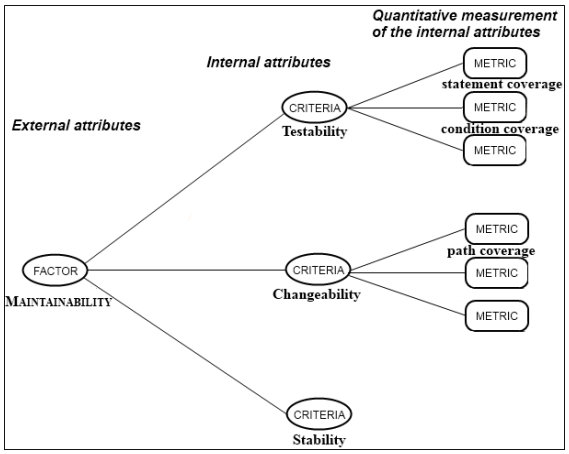

Figure 1: Mc Call model applied to the "Maintainability".

2.2 Lack of Precision

Current methods of evaluation stop the quality attributes decomposition at the *"quality factors step"* and remain too vague when it comes to giving accurate measures to quality. These methods are not precise because they cannot go further in the decomposition and consequently they cannot be automated to the point of defining a finite value for each attribute. They lead inevitably to the establishment of a brainstorming between stakeholders (*see section 3*) for the purpose of the institution of a utility tree (SEI, 2010). Because stakeholders do not have tools to quantitatively measure each quality factor chosen, they shall set up scenarios aiming to respectively solicit separately each criterion, and factor. An approximated evaluation of the architecture is then realized after having studied the system behavior while carrying out the scenarios set.

3 SOAQE

It is precisely where our work differs from those existing insofar as we wish to obtain a precise quantitative measurement for each quality factor with the SOAQE method. We especially aim to automate the process in order to avoid hand-operated evaluations pushing to solicit stakeholders for the whole evaluation.

3.1 Principle of the Method

The process consists in three principal stages corresponding each to a decomposition step of our quality attributes.We first identify relevant quality factors for our architecture.Then we isolate the quality criteria defining them.And finally we define quality metrics composing each criterion in order to quantify them numerically.

3.2 Steps in More Detail

The main idea of the SOAQE process is to evaluate in three steps the whole architecture from every metric to the set of quality factors obtained after having previously identified the business objectives. Our work is based on the architect point of view and the attributes selected are the ones considered as the most relevant among all existing.

3.2.1 Quality Factors

The CBA is defined with *reusability* and *composability* (Crnkovic, Chaudron and Larsson, 2006). Basing on previous analysis, we define the SOA with the *Reusability,* the *Composability* and the *Dynamism.*These three attributes, that we identified as **quality factors** for **SOA** represent the qualitative quintessence which has directed the definition of the object, component and service paradigms.

The **first step** of the SOAQE method consists in choosing a first quality factor to study in depth and there exist a lot which could come out after the

analysis of the business objectives. But we have naturally chosen to work on those identified as the qualitative quintessence for SOA. These three quality attributes (*dynamism, composability and reusability*) define each of our three paradigms to varying degrees. Moreover, there exist a hierarchical ranking propelling "dynamism" on top of SOA concerns, and this is precisely why we chose to especially focus deeply on this quality factor. We may record that each of the two others attributes is of major importance for the three paradigms considered (object, component or service oriented architectures).

3.2.2 Quality Criteria

With regards to our work and after having identified the determining factor quality for SOA (i.e. the dynamism), we were interested in the **second step** of the SOAQE method, namely, discovering the criteria defining the factor on which we have gone through.

Further down the road, any factor is composed by a set of criteria that could be looked at as part of our work. These criteria belong to different families of attributes and a few of them are common to the three quality factors chosen for our paper. But we deliberately concentrated our work on technical criteria of each of the quality factors studied because we adopted the point of view of an architect that is itself a technical stakeholder. In this light, we identified six criteria common to each of our three factors. These technical criteria gather elements having significant impacts on global quality, from the development process to the system produced: the ***loose coupling*** (potential of dependences reduction between services), the ***explicit architecture*** (paradigm ability to define clear architectural application views), the ***expressive power*** (potential of paradigm expression in terms of creation capacity and optionalities), the ***communication abstractions*** (paradigm capacity to abstract services functions communications), the ***upgradability*** (paradigm ability to make evolve its services), and the ***owner's responsibility*** (corresponds to the responsibilities sharing out between services providers and consumers). Each of these quality criteria is given varying degrees of consideration according to the quality factor in question. Our previous works (Hock-Koon, 2011) allowed organizing hierarchically (under three distinct levels) these quality criteria for each of the three quality factors (dynamism, reusability and composability).

Consequently, we obtain the triptych of the figure 2 while considering all paradigms. While focusing on the dynamism, our previous work (Hock-Koon, 2011) allowed to conclude that the "**loose coupling**" criterion is of biggest importance for this factor *(see figure 2)*, this is why we chose to concentrate in depth, on it; whereas, the criterion "expressive power" is of less importance.

3.2.3 Quality Metrics

The coupling is relatively well known by the community, and thus, our lab team members decided to look into the matter (Hock-Koon, 2011).

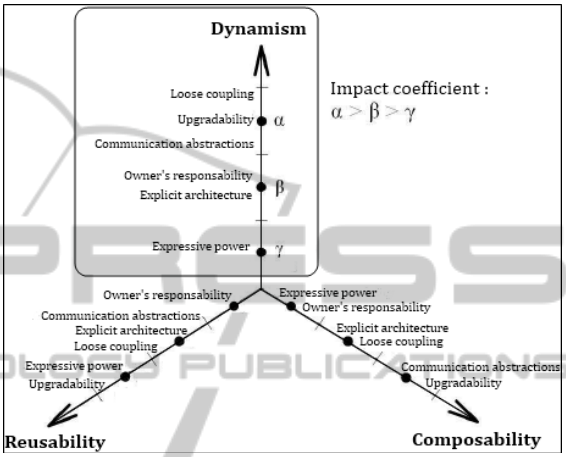

Figure 2: Expression of reusability, composability and dynamism perspectives.

We found three quality metrics for the latter which must be considered for the **third step** of the SOAQE method (the semantic coupling: {high, low or non-predominant} *based on the high-level description of a service defined by the architect,* the syntactic coupling: {high, low} *measures dependencies in terms of realization between abstract services and concrete services* and the physical coupling: {γ, β and α with $0 \leq \gamma \leq \beta \leq \alpha$} *focusing on the implementation of the service*). This implementation corresponds to a particular instance of the service where a choice has been made concerning concrete services to use. A unique solution has been chosen to fulfill each of the needs expressed by abstract services. It reuses existing researches (Perepletchikov, Ryan, Frampton and Tari, 2007) and it is based on measurements such as methods calls, messages exchanged, the number of linked services, commune objects and so forth. These metrics shall make it possible to identify physical dependencies between concrete services.

4 EXPERIMENTATION

For the experimentation, we tempted to quantitative-

ly measure the key quality attributes discussed in the previous sections of this paper; notably, the quality factor "dynamism", the "loose coupling" criteria and the "physical, syntactic and semantic coupling" metrics. That being said, it is important to note that the SOAQE method must be reproduced for every quality factor identified after having analyzed the objectives of the company and the set of criteria and metrics belonging to that quality factor.

4.1 Loose Coupling

Taking as a starting point an existing formula of the field of "Preliminary analysis of risks" (*see formula 3.1*) (Mortureux, 2002) our works led to the identification of a mathematical formula (*see formula 3.2*) combining the three couplings studied: semantic, syntactic and physical.

NB: The simplified formula (see formula 3.1) usually used in the automotive industry, makes it possible to measure the default risk of a car component A is the Criticality of the car component, B is the Probability of occurrence of a failure on this component and C is the Probability of non-detection of this failure.

We associate this concept of risk with our vision of the coupling. Correlatively, the quintessence of the coupling is the expression of the dependences which can exist between two elements and the principle of dependence defines that one element cannot be used without the other. Reducing the risk that the role defined by a service cannot be assured anymore is decreasing the dependence of the application in relation to this service and thus reducing its coupling. The calculation of this risk takes into account all the characteristics influencing the coupling by redefining the three variables A, B and C according to the semantic, syntactic and physical couplings. The global coupling corresponds to the sum of the three couplings calculated individually beforehand. The lower this result is, the more the coupling is weak.

NB: The criticality A∈[(a),(b),(c)] is affiliated to the semantic coupling. 'a' if the service is only associated to non predominant couplings, 'b' for non predominants and low couplings and 'c' for non predominants,low and high couplings, while 'Ps' is the probability of failure of a service.

5 CONCLUSIONS

The finality of our work is to design a conceptual framework and, in fine, a semi-automated prototype *(based on past methods, such as ATAM or SAAM)* which could quantify with an accurate value the quality of the whole service oriented architecture. Another pursued goal consists in bringing to the customer "less abstract" documents than those proposed today. The quality concept remaining a relative one, we will target the sectors requiring a special attention by directly addressing the various development lab teams charged with the relevant functions.

REFERENCES

Crnkovic, I., Chaudron, M., Larsson, S., 2006. Component-based development process and component lifecycle, *ICSEA'06, International Conference on Software Engineering Advances*.

Clements, P., Kazman, R., Klein, M., 2001. Evaluating Software Architectures: Methods and case studied, published by Addison-Wesley Professional.

Hock-Koon, A., 2011. *Contribution à la compréhension et à la modélisation de la composition et du couplage faible de services dans les architectures orientées services*. Thesis (*PhD). University of Nantes.*

Mortureux, Y., 2002. Preliminary risk analysis. Techniques de l'ingénieur. Sécurité et gestion des risques.

Perepletchikov, M., Ryan, C., Frampton, K., Tari, Z., 2007. Coupling metrics for predicting maintainability in service-oriented designs. *ASWEC'07. Australian Software Engineering Conference*

SEI. *The Carnegie Mellon Software Engineering Institute.* 2011 [Online]. Available From: www.sei.cmu.edu/architecture/start/glossary/

ISO/IEC 9126-1:2001: *Software Engineering – Product Quality – Part 1: Quality Model. International Organization For Standardization*, Geneva, Switzerland [Online]. Available From: Http://Www.Iso.Org/Iso/Iso_Catalogue/Catalogue_Tc/Catalogue_Detail.Htm?Csnumber=22749

APPENDIX

$$R = A * B * C \quad (3.1)$$

$$Coupling = \left\{\{(a),(b),(c)\}\right\} * \left\{\left(\left(\prod_{k=1}^{\lambda}\prod_{i=1}^{N}\sum_{j=1}^{s}((P_s)_{kij})^{\alpha_{kij}}\right) * C_{phys}\right) * \left\{P_{ndetect}\right\} \quad (3.2)$$

Figure 3: Defaut risk of a car component (3.1) and global coupling of an architecture (3.2) formulas.