Towards the Evaluation of System-of-Systems Software Architectures

Daniel Soares Santos[1], Brauner Oliveira[1], Milena Guessi[1],

Flavio Oquendo[2], Marcio Delamaro[1], Elisa Yumi Nakagawa[1]

[1]ICMC - University of São Paulo (USP)
São Carlos - Brazil.

[2]IRISA - Université de Bretagne Sud (UBS)
Vannes - France.

{danielss, brauner}@usp.br, {milena, delamaro, elisa}@icmc.usp.br, flavio.oquendo@irisa.fr

Abstract. *Evaluation of software architectures is an important activity to the quality of software systems, as it verifies conformance and completeness of such architecture regarding requirements and goals. In another perspective, Systems-of-Systems (SoS) have emerged as a new class of software systems, which aggregates independent and heterogeneous constituent systems for performing new, emergent capabilities. Likewise, evaluation of SoS software architectures is also important for ensuring that important quality attributes are met in the SoS. The main contribution of this study is to present current challenges for evaluating SoS software architectures and point out important perspectives of research in that direction. We observe that despite the several proposals for evaluating software architectures of SoS, there is still no consensus on exactly what should be evaluated in such architectures. Moreover, there are several difficulties that need to be overcome.*

1. Introduction

Software architecture is essential to the success of software intensive systems and plays a key role in determining the quality of software systems. Decisions made at the architectural level directly enable, facilitate, or interfere in the achievement of business goals as well as of functional and quality requirements [Bass et al. 2012]. In this context, the architectural evaluation can be used for comparing and identifying strengths and weaknesses of different architectural alternatives. Evaluation can also guide the maintenance or indicate new opportunities for enhancing software architectures. Finally, evaluation is essential for ensuring that software architectures meet desired quality attributes [Bosch 2000].

Several approaches for evaluating software architectures can be found in the literature. Bosch (2000) identify four main groups for these evaluation approaches: (i) experience-based methods, which are based on previous experience and domain knowledge of consultants or developers; (ii) simulation-based methods, which typically rely on a high level implementation of the software architecture for evaluating its performance and accuracy; (iii) mathematical modeling methods, which use mathematical proofs for evaluating operational quality requirements, such as performance and reliability; and (iv) scenario-based methods, which evaluate a particular quality attribute by creating scenario profiles. Each scenario describes an intended use of the system by means of a concrete description of the quality requirement. These scenarios help to identify architectural risks and its potential consequences through an efficient and scalable way.

In parallel, software-intensive systems have become increasingly large and complex, with their considerable dissemination in various application domains. In this context, Systems-of-Systems (SoS) result from the integration of several constituent systems that operate independently and could potentially be developed using different technologies and platforms. An adequate integration has been more and more necessary to promote cooperation among these independent systems in order to provide more complex functions, which could not be provided by any system working separately. SoS has been proposed for different domains, in particular, for critical embedded systems, such as medical systems, airport systems, robotic and automotive [Nakagawa et al. 2013].

Besides interoperability, several other quality attributes are critical for SoS (e.g., performance, reliability and flexibility). However, it is quite challenging to meet these quality attributes in SoS, as their constituents are often developed and maintained by different organizations. Moreover, these organizations may have their own stakeholders, development teams, and processes, which collaborate for increasing this challenge. In this context, the evaluation of SoS software architectures could ensure that these quality attributes are satisfied from the early stage of the SoS lifecycle. An early evaluation of the software architecture quality also aids in validating architectural decisions.

In this scenario, the main goal of this paper is to present the main challenges in evaluating SoS software architectures. For this, it was based on results of a Systematic Literature Review (SLR)[1]. In this SLR, we surveyed the difficulties and challenges that are inherent to SoS evaluation as a way of pointing out new, important research perspectives in the software architecture area. Overall, 16 primary studies were included in this SLR. The following discussion builds upon these studies.

The remainder of this paper is organized as follows. Section 2 presents a brief introduction about current research on evaluation of SoS software architectures and elaborates on the main quality attributes that have been addressed for such architectures. Section 3 discusses the main challenges and difficulties for performing this evaluation. Finally, Section 4 presents our final considerations and perspectives to future work.

2. Evaluation of SoS Software Architectures

Evaluation of software architectures usually occurs after the design of such architectures but before implementation starts. Nonetheless, an architecture can be evaluated at any stage of its life cycle [Clements et al 2002]. In particular, for SoS software architectures, due to their characteristics, we have observed that most works have proposed application of evaluation methods in the design phase, as well as in architectures already established, intending to analyze their flexibility and ability to evolution. In our SLR, we also identified several evaluation methods for SoS. Moreover, we also identified the most common quality attributes considered during the evaluation of SoS software architectures. The following sections summarize our findings.

[1] Available at http://goo.gl/PU12iQ (last accessed on 07/13/2014)

2.1. Evaluation Methods

The processes for evaluation of SoS software architectures are typically supported by different methods and techniques. These methods and techniques usually are adapted and enhanced to create a new proposal for SoS software architectures. Overall, six of the 16 works propose or use mathematical-modeling evaluation methods, four works propose or use simulation-based methods, and six works propose or use scenario-based methods. Among the scenario-based approaches, Architecture Trade-off Analysis Method (ATAM) by [Kazman 2000] is the most popular one. This method deals with multiple quality attributes, their relationships, and trade-offs at the architecture level in order to gain insight about the compliance of the architecture implementation regarding quality requirements and business objectives. In our SLR, it was not observed a convergence in using a specific type of evaluation method. Besides that, we did not find works that consider experience-based methods for evaluating SoS.

The suitability of the methods and techniques usually has been assessment through expert opinion and experiences in real projects, proof of concept or demonstration, case study, and application in industry. It is important to highlight that all works selected to evaluation in the industrial context have proposed to use scenario-based evaluation methods. This result shows that these evaluation methods have been well accepted by industry or at least that they could be scalable to SoS.

2.2. Quality Attributes

Evaluation methods can either focus on single or several quality attributes. Through results of our SRL, all proposed scenarios-based methods do not focus on specific quality attributes. The main reason is that scenarios-based methods usually focus on identifying trade-off among different qualities attributes instead of measuring each quality attribute. However, simulation-based and mathematical modeling methods usually focus on one or a few tangible quality attributes. The most common quality attributes considered by these methods are reliability, performance, operability, complexity, and flexibility. However, we have observed that evaluation methods for SoS should take into account several quality attributes. Moreover, these methods should also be able of measuring and classifying these quality attributes in order to support an accurate comparison among architectural alternatives. This may be possible through the use of simulation-based in combination with scenarios-based approaches.

Finally, the use of quality models for evaluating SoS architectures would be relevant, as they would provide standardization for quality attributes of SoS, as well as establishment of relationships among such attributes. However, none of the works included in our SLR discusses the use of quality models during architectural evaluation.

3. Challenges

We have also observed that there are several challenges for an adequate evaluation of SoS software architectures. The following discussion focuses on the main challenges.

The reliability of the communication among constituent systems is an important factor to the success of SoS [Urwin et al. 2010]. According to Stratton (2009), it is difficult to ensure reliable communication through an architecture evaluation for several reasons: (i) constituent systems are usually developed independently by different teams at different places; (ii) specification of communication requirements is ambiguous; and (iii) communication issues are often subtle and remain hidden for a long time. Moreover, the complex interdependencies that exist among constituents make it difficult to foresee the behavior of SoS due to an unexpected loss of one of their constituents. In the worst, SoS could collapse or trigger a cascading failure among their constituents. These consequences cannot be fully understood only through an architectural evaluation of the independent systems, as SoS require an evaluation of the effect of interdependence among constituents on the entire system [Guariniello, C. and DeLaurentis 2014a].

Regarding the evolutionary and decentralized nature of SoS, it becomes difficult to ensure, for instance, reliability, security, or performance, using architecture evaluation methods, which focus exclusively on structural characteristics but ignore behavior compliance. This could be a problem, as a simple divergence in the implementation of one of the constituents often reduces performance and reliability of the entire SoS [Chen et al. 2012, Ackermann et al. 2009 and Zhu et al. 2008]

Finally, an important step to an adequate architectural evaluation involves identification of metrics to measure features of systems. However, metrics used to evaluate individual systems can not directly deal with the characteristics of the SoS [Guariniello and DeLaurentis 2014b]. This happens because the emergent behavior of SoS usually cannot be captured and evaluated by evaluation approaches that address the level of constituent systems [Meilich 2006].

4. Conclusion

This position paper briefly presents the most important results of our SLR about SoS software architecture evaluation. Despite the number of initiatives to evaluate such architectures, there is still no consensus on what exactly should be considered during this evaluation. From our results, we observe that main challenges in the SoS architecture evaluation are due to the complex interaction among constituent systems and the evolutionary, distributed nature of SoS as well. Therefore, appropriate, scalable evaluation approaches still need to be developed. Moreover, we envisage that these new approaches should be able to successfully capture and evaluate the emergent behavior of SoS.

As future work, we intend to continue our investigation on evaluation of SoS architectures, updating this SLR as well as identifying appropriate architecture evaluation methods that consider quality attributes usually addressed by SoS. Moreover, we will investigate alternatives to combine these methods and techniques in order to reduce the number of difficulties and challenges that are inherent to this new class of complex, large software systems.

References

Ackermann, C., Lindvall, M. and Cleaveland, R. (2009) “Towards behavioral reflexion models.” In *ISSRE’2009*, pages 175–184, Mysuru, India.

Bass, L., Clements, P. and Kazman, R. (2012) “Software Architecture in Practice”. SEI Series in Software Engineering. Pearson Education.

Bosch, J. (2000) “Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach”. ACM Press Series. Addison Wesley.

Chen, H.-M., Kazman, R. B. and Hong-Mei, C.. (2012) “Architecting ultra-large-scale green information systems”. In *GREENS’2012*, pages 69–75, Zurich, Switzerland.

Clements, P., Kazman, R. and Klein, M. (2002) “Evaluating software architectures: methods and case studies”. SEI series in software engineering. Addison-Wesley.

Guariniello, C. and DeLaurentis, D. (2014a) “Communications, information, and cyber security in systems-of-systems: Assessing the impact of attacks through interdependency analysis“. In *CSER’2014*, pages 720–727, Redondo Beach California, USA.

Guariniello, C. and DeLaurentis, D. (2014b) “Integrated analysis of functional and developmental interdependencies to quantify and trade-off ilities for system-of-systems design, architecture, and evolution”. In *CSER’2014*, pages 728–735, Redondo Beach, California, USA.

Kazman, R., Klein, M., Clements, P. (2000) "ATAM: A Method for Architecture Evaluation". CMU/SEI-2000-TR-004. Software Engineering Institute. Carnegie Mellon University.

Martensson, F. and Tekniska Hogskola, B. (2006) “Software Architecture Quality Evaluation: Approaches in an Industrial Context”. Ph. D. thesis, Blekinge Institute of Technology, Karlskrona, Sweden.

Meilich, A. (2006) “System of systems (SoS) engineering & architecture challenges in a net centric environment”. In *SoSE’2006*, pages 1-5, Shanghai, China.

Nakagawa, E. Y., Gonçalves, M., Guessi, M., Oliveira, L. and Oquendo, F. (2013) “The state of the art and future perspectives in systems of systems software architectures”. In *SESoS’2013*, pages 13-20, Montpellier, France.

Stratton, W., Sibol, D., Lindvall, M., Ackermann, C. and Godfrey, S. (2009) “Developing an approach for analyzing and verifying system communication.” In *IEEE Aerospace Conference*, Big Sky, Montana, USA.

Urwin, E., Venters, C., Russell, D., Liu, L., Luo, Z., Webster, D., Henshaw, M. and Xu, J. (2010) “Scenario-based design and evaluation for capability“. In *SoSE’2010*, Loughborough, UK.

Zhu, L., Staples, M. and Jeffery, R. (2008) “Scaling up software architecture evaluation processes”. In *ICSP’2008*, pages 112–122, Leipzig, Germany.