On the Relations between Systems-of-Systems and Software Ecosystems

Rodrigo Santos[1], Marcelo Gonçalves[2], Elisa Yumi Nakagawa[2], Cláudia Werner[1]

[1]PESC/COPPE –Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[2]ICMC/USP – University of São Paulo, São Carlos, Brazil

{rps, werner}@cos.ufrj.br, {marcelob, elisa}@icmc.usp.br

Abstract. *Software development commonly involves many individuals, groups, and organizations related by one or more central platforms. Moreover, organizations have created and maintained products and services with different technologies and for diverse application domains. In this scenario, topics of research have arisen in the Software Engineering area in order to adequately deal with such large, complex software systems. In this perspective, Systems-of-Systems and Software Ecosystems are two topics that have been separately investigated and, at the same time, they are complementary. In this paper, we perform a preliminary discussion on how these topics are related to each other, aiming at supporting collaborative research.*

1. Context and Motivation

According to Boehm (2006), the increasing pace of change in the global industry is driving organizations towards increasing levels of agility in their software development methods, while their products and services are concurrently becoming more and more software-intensive. It other words, software has represented a crucial element for most of existing systems, since it impacts functions, resources, and risks in different sectors of industry (Santos & Werner, 2011). Software-intensive systems have also become increasingly ubiquitous, large, and complex, with considerable dissemination in various application domains. In parallel, the software industry creates products, services, and processes considering different facets and perspectives of value, because it produces value realized by its stakeholders (e.g., companies, developers, clients and users). In this perspective, Software Engineering (SE) community has defined the treatment of tangled technical, economic, and social issues as important research themes (Boehm, 2006).

Diverse research topics have arisen in the SE area, in order to adequately deal with such software systems. Recently, there has been a growing interest in a class of software-intensive systems, known as Systems-of-Systems (SoS) (Maier, 1998), whose constituents are themselves complex, heterogeneous, independent, and large. In another perspective, organizations have opened up their software platforms and assets to others, including a community of partners and 3rd-party developers over the world (Bosch, 2010). The set of actors, artifacts, and relations, either internal or external, over a software platform has been called Software Ecosystems (SECO) (Messerschmitt & Szypersky, 2003). SoS and SECO are two topics that have been still separately investigated, but certainly they should be treated in a complementary way. We perform a preliminary discussion on SoS and SECO as related research topics, promoting opportunities for collaborative research. This paper is organized as follows: Sections 2 and 3 present an overview of SoS and SECO, respectively; Section 4 discusses relations between SoS and SECO; and Section 5 provides some final considerations.

2. Systems-of-Systems Overview

In the last years, there has been a growing interest in research and development of SoS, resulted from the integration of other independent and heterogeneous systems. These systems are useful for several of society's needs, such as healthcare, avionics, logistics, energy, and transportation (De Laurentis & Crossley, 2005). Although SoS have several definitions, there is a set of consensual characteristics (Maier, 1998): (i) the *operational independence* in which all of the constituent systems of an SoS can deliver their functionalities when not working in the SoS; (ii) the *managerial independence* in which each constituent system can be individually managed; (iii) the *evolutionary development* in which SoS may evolve over time to respond to changes in its environment, the constituent systems, or in its requirements; (iv) the *emergent behavior* in which SoS behaviors are the result of constituent systems working together and cannot be provided by any of these systems alone; and (v) the *geographical distribution* in which the constituent systems of an SoS are physically decoupled.

SoS exist to accomplish their major goals. That is, the constituent systems have their own individual mission and contribute to the accomplishment of the global mission. Moreover, both SoS and their constituent systems can have their respective stakeholders, which can also have different perspectives of interest. Furthermore, SoS can assume different categories, according to particular aspects (DoD, 2008): (i) *virtual* in which there is no central control. Therefore, the constituent systems are independently managed in a distributed and uncoordinated environment where the mechanisms to maintain the whole SoS are not evident; (ii) *collaborative* in which the constituent systems voluntarily collaborate more or less in order to address shared or common interests. In this case, there is a central control that offers standards to enable the collaboration of the constituent systems; (iii) *acknowledged* in which the goals, management, resources, and central control of the SoS are recognized, but the constituent systems still retain their independent management; and (iv) *directed* in which there is a central control and specific main purposes. The constituent systems can have their operational and managerial independence, but they are subordinated to the central control.

3. Software Ecosystems Overview

Components, services, and applications developed by the global software industry have a direct relation with collaborators in promoting, distributing or selling, and evolving software systems based on software technologies (platforms), the so-called SECO (Messerschmitt & Szyperski, 2003). External and/or unknown developers are also contributing to maintain and evolve these systems, changing the traditional value chain. In this case, networks of multiple products and services over platforms should be used to support the comprehension of relationships among organizations in SECO, focusing on a *business sense* (Santos & Werner, 2011). On the other hand, social impacts should be taken into account due to the socialization of SE (Mens & Goeminne 2011). The cycle of creating, providing, and operating software systems occurs over a net of tangled stakeholders. These elements contribute to (depend on) the propagation, amplification, and expansion of platforms in the software industry. Thus, a *community sense* emerges because business models are revisited in order to treat transactions/transfers in open value chains (Santos & Werner, 2012). Both senses represent the trend of hybrid models to manage and engineer software systems (Popp, 2012).

Some technical challenges are reinforced in SECO, especially regarding software architecture (Bosch, 2010): (i) *stability*: once an organization is providing a platform for external developers, interfaces should evolve in a predictable fashion and with significant time for adjustments; (ii) *simplicity*: integration at each of the levels of data, workflow, and UI integration should be designed to minimize the complexity of the final solution; (iii) *security and reliability*: architecture should be designed to minimize defective and malicious external code and vulnerabilities; and (iv) *evolution*: the scope of the platform needs to be constantly adjusted upwards in order to incorporate functionalities based on the community's emerging requirements, but also slim down the platform through re-architecting it to replace proprietary code with commercial or *open source* components.

4. SoS & SECO

SoS and SECO are two emerging, relevant research topics in SE area. However, it is observed that typically the solutions for SECO and SoS are individually proposed by isolated teams in order to meet particular domain-oriented problems. First of all, according to Kazman *et al.* (2012), architectural challenges have motivated a number of methods for the design, documentation, and analysis of the traditional single systems and architectures over the past 10 years. Moreover, these methods share many of the same principles, which can be used to systems of different scales. Aiming at helping the comprehension of architecture in SECO, Klein & McGregor (2013) amplified the concept of architecture to the so-called *SoS platform*, or "industry platform", also defined by Cusumano (2010). This kind of platform provides domain-specific and general services to a set of systems that need to interact to form an SoS (Maier, 1998). From the SECO viewpoint, an SoS platform can exist in an environment of different levels of actors, artifacts, and relationships towards the development of globalized, large-scale, and long-term products and services (Santos & Werner, 2012). These products and services can be developed with different technologies, where integration and communication are crucial, since they are software-intensive systems (Brown & McDermid, 2007). Moreover, SoS platforms fulfill the inherent characteristics of SoS.

Additionally, the SoS concept started to gain its popularity mainly in military domain as a strategy for reaching goals, or delivering unique capabilities that are the result of a collaborative work of a dynamic set of existing systems (DoD, 2008). The evolution of computational systems reveals that more software-intensive systems tend to meet the SoS concept. Based on that, it is possible to identify many examples of studies addressing new application domains where SoS is gaining ground, such as Smart Cities, Global Earth Observation, and Critical Embedded Systems. On the other hand, the SECO concept is popular in software business platforms and open source software (OSS) domains (Manikas & Hansen, 2013). SECO can be seen as an application domain for SoS as such (Kazman *et al.*, 2012; Klein & McGregor, 2013; Axelsson *et al.*, 2014).

SoS are typically complex, interdisciplinary systems whose functionalities and purposes can dynamically evolve (Firesmith, 2010), encompassing several new challenges to be developed (DoD, 2008). In turn, the concepts of virtual and collaborative SoS have been explored in the SECO context, allowing collaboration of different constituent systems and organizations in order to produce emergent functionalities (Klein & Vliet, 2013). In both collaborative and virtual SoS, SECO are more valuable because in these categories there is no strict control over the constituent systems. As such, SECO concerns may aid SoS to leverage the network of actors in order to deal with innovation from their constituent systems.

Despite the discussion on the SoS categories for SECO scenarios, we have preliminarily drawn some similarities between SoS characteristics (Maier, 1998) and SECO technical challenges (Bosch, 2010). For example, architectural stability required for SECO platforms regarding to their components, services, and applications can be compared to the operational independence of constituent systems of an SoS. In this case, software systems integration and component-based development can be combined to support strategies to cope with application programming interface issues. On the other hand, platform evolution directly depends on the SECO community's emerging requirements and contributions, as well as the adjustments of underlying hybrid business models. As such, SoS evolutionary development should take into account not only the environment's technical issues but also the business and social ones. In this case, SoS architecture models should be extended to cope with context variables based on value chains and social networks. Finally, emergent behavior produced by constituent systems of an SoS working together can be related to the security and reliability in SECO. System dynamics theory may be a useful instrument to simulate components configurations in order to improve the architectural design.

5. Final Considerations

Both academia and industry have recognized the importance of SoS and SECO to an effective development of software systems. In this perspective, more joint research must be conducted since the notion of SECO may be at least as generalizable as SoS themselves. As future work, we intend to investigate how SECO platforms can benefit from SoS mindset and how SoS can benefit from business and social networks. We also intend to evaluate the interactions between SoS and SECO in order to more clearly state differences and similarities of such systems from an SE perspective through conduction of a systematic mapping study. Finally, a few more concrete implications for Distributed Software Development (DSD) will be investigated and discussed, since one of the SoS/SECO characteristics is the potentially distributed nature.

References

Axelsson, J. *et al.* (2014) "Characteristics of software ecosystems for Federated Embedded Systems: A case study". Information and Software Technology. In press.

Boehm, B. (2006) "A View of 20th and 21st Century Software Engineering". In: 28th International Conference on Software Engineering (ICSE), Shanghai, China, 12-29.

Bosch, J. (2010) "Architecture Challenges for Software Ecosystems". In: Proceedings of the 4th European Conference on Software Architecture (ECSA), 2nd International Workshop on Software Ecosystems (IWSECO), Copenhagen, Denmark, 93-95.

Brown, A., McDermid, J. (2007) "The art and science of software architecture". In: 1st European Conference on Software Architecture (ECSA), Aranjuez, Spain, 237-256.

Cusumano, M. (2010) "The Evolution of Platform Thinking". Communications of the ACM 53(1):32-34.

De Laurentis, D., Crossley, W. (2005) "A taxonomy-based perspective for systems of systems design methods". In: IEEE International Conference on Systems, Man and Cybernetics (SMC), Waikoloa, USA, 86-91.

DoD (2008) "System engineering guide for systems of systems". Technical Report, Version 1.0, August 2008.

Firesmith, D. (2010) "Profiling systems using the defining characteristics of systems of systems". Technical Note CMU/SEI-2010-TN-001, Software Engineering Institute.

Kazman, R. *et al.* (2012) "Scaling up software architecture analysis". Journal of Systems and Software 85(7):1511-1519.

Klein, J., McGregor, J. (2013) "System-of-Systems Platform Scoping". In: 4th International Workshop on Product Line Approaches in Software Engineering (PLEASE), San Francisco, USA, 1-4.

Klein, J., Vliet, H. (2013) "A Systematic Review of System-of-systems Architecture Research". In 9th International Conference on Quality of Software Architectures (QoSA), Vancouver, Canada, 13-22.

Maier, M. (1998) "Architecting principles for systems-of-systems". System Engineering 1(4):267-284.

Manikas, K., Hansen., K. (2013) "Software Ecosystems – A Systematic Literature Review". Journal of Systems and Software 86(5):1294-1306.

Mens, T., Goeminne, M. (2011) "Analysing the Evolution of Social Aspects of Open Source Software Ecosystems". In: 3rd International Workshop on Software Ecosystems (IWSECO), Brussels, Belgium, 77-88.

Messerschmitt, D., Szyperski, C. (2003) "Software Ecosystem: Understanding an Indispensable Technology and Industry". The MIT Press.

Popp, K. (2012) "Leveraging Open Source Licenses and Open Source Communities in Hybrid Commercial Open Source Business Models". In: 4th International Workshop on Software Ecosystems (IWSECO), Boston, USA, 33-40.

Santos, R., Werner, C. (2011) "Treating Business Dimension in Software Ecosystems". In: 3rd ACM/IFIP International Conference on Management of Emergent Digital EcoSystems (MEDES), San Francisco, USA, 197-201.

Santos, R., Werner, C. (2012) "Treating Social Dimension in Software Ecosystems through ReuseECOS Approach". In: 6th IEEE International Conference on Digital Ecosystems and Technologies – Complex Environment Engineering (DEST-CEE), Campione d'Italia, Italy, 1-6.