On Service-Oriented Architectural Concerns and Viewpoints

Qing Gu
Department of Computer Science
VU University Amsterdam
The Netherlands
qinggu@cs.vu.nl

Patricia Lago
Department of Computer Science
VU University Amsterdam
The Netherlands
patricia@cs.vu.nl

Abstract

Despite the many publications around the innovation and challenges introduced by SOSE and SOA, the 'real' differences with traditional software engineering and software architecture are still very fuzzy. To better understand the innovative points (if any), in this work we identified seven fundamental differences by conducting a literature review, linked them to some service relevant aspects and mapped the differences & service aspects on the well-known architecture-related concepts of 'architectural concerns' & 'architectural viewpoints'. As such, we were able to identify an initial set of requirements for service-oriented viewpoints. If specialized in industrial contexts, we would be able to exemplify innovative points by capturing service relevant aspects. In perspective, this work seems to scale down 'innovation' of SOSE and SOA, to 'relevance' of service aspects to engineering and architecture.

1. Introduction

Despite the many publications around the innovation and challenges introduced by Service-Oriented System Engineering (SOSE) [1] and SOA, the 'real' differences with Traditional Software Engineering (TSE) and Software Architecture (SA) are still very fuzzy. There are two reasons that cause the innovative points of SOSE and SOA remain unclear.

First, numerous differences between SOSE (resp. SOA) and TSE (resp. SA) have been discussed in the literature. However, some differences referring to the same or similar meanings are described differently (e.g., both highly dynamic environment and high uncertainty can be seen as open-world assumptions [2]); and some differences can be derived from some other ones (e.g., due to open-world assumptions, security becomes more critical [2]). As a result, it seems that there are many differences. However, the fundamental ones are hidden.

Secondly, innovative points are often considered as issues that are unique to SOSE and as such have never been studied or practiced in the engineering or architecting of traditional software systems. Although numerous SOSE challenges have been proposed both in academia and industry, the lack of common agreements on the novelty of these challenges makes it difficult to determine what are the innovative points [3] [4].

While our ultimate goal is to understand the 'real' differences and their implications on SOSE and SOA, in this particular work we focus on gaining insight into the fundamental ones (without judging whether they are innovative) as the first step towards the goal. With this objective, we conducted a systematic literature review and linked its results to the service aspects [5] [6] (i.e., issues that are specifically relevant to SOSE and SOA) introduced in our previous work.

We further constructed a framework illustrating the relationship between the fundamental differences and the service aspects. Moreover, we mapped the differences & service aspects on the well-known architecture-related concepts [7] of 'architectural concerns' & 'architectural viewpoints'.

By doing so, we were able to identify an initial set of requirements for service-oriented viewpoints. Applying this set of requirements within industrial contexts or SOA project, we would finally be able to highlight service relevant aspects, and therefore exemplify architectural differences (if any), which is one step towards our ultimate goal.

The rest of the paper is organized as follows. In Section 2 we discuss our motivation and the research approach we conducted. Section 3 presents identified differences and their links to service aspects in a framework. In Section 4 we discuss the needs for service-oriented viewpoints and we conclude the paper in Section 5.

2. Motivation and Research Approach

It is often argued, in both academia and industry, whether the existing software engineering and architecting approaches (including techniques, methods and tools) are applicable *as is* in the context of SOA [8] [3] [1]. However, since their 'real' differences with TSE remain fuzzy, it is unclear which existing approaches can be reused or tailored and which ones are completely invalid.

As a result, new approaches and design principles to build service-oriented systems have been continuously emerging in the past 10 years. When analyzing these new approaches

978-1-4244-4985-9/09/ 25.00 ©2009 IEEE

and design principles, we start to wonder whether they are specific to services or yet existing concepts with some new (if any) implications.

During our previous work [5] [6] on the identification of service aspects, we realized that by capturing them in service process models and design decisions, we were able to pinpoint the issues that are specifically relevant to SOSE or SOA. However, the way in which we identified the service aspects were based upon studies of the literature and industrial cases as well as experience of authors. A more systematic approach is lacking.

In this work, with the purpose of establishing a sound foundation for identifying service aspects, we carried out the research in the following two phases:

Phase 1: a systematic literature review. An effective approach to identifying differences between service-oriented approaches and traditional ones claimed in the literature is to conduct a systematic literature review. This method has been used in our previous work to explore SOSE-related challenges [9], where a population of 51 scientific papers (published between January 2000 and July 2008) have been selected as primary studies out of a total of 729 analyzed papers. These primary studies were reused in the review conducted in this work.

Phase 2: a study of the links between the differences and the service aspects. In our previous work we highlighted the way in which SOSE process modeling and SOA architectural decision making are different from the traditional approaches in terms of some *service aspects*. Since we argued that these aspects are peculiar to services and reveal the differences with traditional approaches [5] [6], we are encouraged to verify whether those service aspects can be linked to differences identified in Phase 1.

3. The Framework

A framework illustrating the relation between the service aspects and the identified differences is given in Figure 1. The boxes in the figure represent the seven fundamental differences and the ellipses represent the service aspects. While continuous arrows denote the relationships between the differences, the dashed arrows indicate the links between differences and service aspects. In this section, we explain all the elements in the framework in detail.

The identified differences between service-oriented approaches and traditional ones are explained as follows:

- *Services are the building blocks*: instead of focusing on implementing a software system as a whole, SOSE and SOA focus on composing coarse-grained discoverable services acting as building blocks of service-oriented systems.
- *Services are open*: instead of not allowing any architectural changes after deployment, an architecture of a service-oriented system can be changed or even determined at runtime since services become the building blocks that can be composed at runtime.
- *Additional development roles are involved in development*: developer is not the only development role. This is rather split into three essential roles: service consumer, service provider and service broker since services are building blocks that need to be published and consumed.
- *Services are consumed and executed remotely*: instead of buying and installing software locally, users of services (pay and) consume services that are executed remotely at the service provider's side since services as building blocks are by definition used rather than owned.
- *Services are cross-organizational*: a software is often managed and owned by one organization. Instead, as services are not executed locally at the consumer's side, the control of services is often highly distributed and crosses the trust and organizational boundaries.
- *SOA is designed under open-world assumptions*: instead of assuming stable execution environment, high uncertainty of the external environment has to be kept into consideration in SOSE and SOA.
- *Services are designed with multiple sets of (non)-functional requirements*: a software application is engineered for a single set of (non)- functional requirements. Instead, since the consumers as well as their needs are not completely known at design time (according to open-world assumptions), services are engineered with multiple sets of (non)- functional requirements to fulfill different groups of potential consumers with different quality requirements.

For the sake of space, only the way in which service aspects address the identified differences is explained here. For further discussion on service aspects, the reader is referred to [5] and [6].

- *Increased importance of the identification of stakeholders.* The additional stakeholders involved in development increase the importance of capturing them explicitly in process models.
- *Cross-organizational collaboration.* Capturing the way in which cross-organizational collaboration is carried out is crucial to highlight that the additional development roles are distributed in different organizations and services are owned by different business partners.
- *Increased runtime effort.* The high uncertainty resulted from open-world assumption demands for more runtime effort. Further, open services imply that more decisions have to be postponed at runtime. Making runtime effort explicit in a process model may highlight which uncertainty is dealt with and which decisions are postponed.
- *Different architecture types.* The architectures of ser-

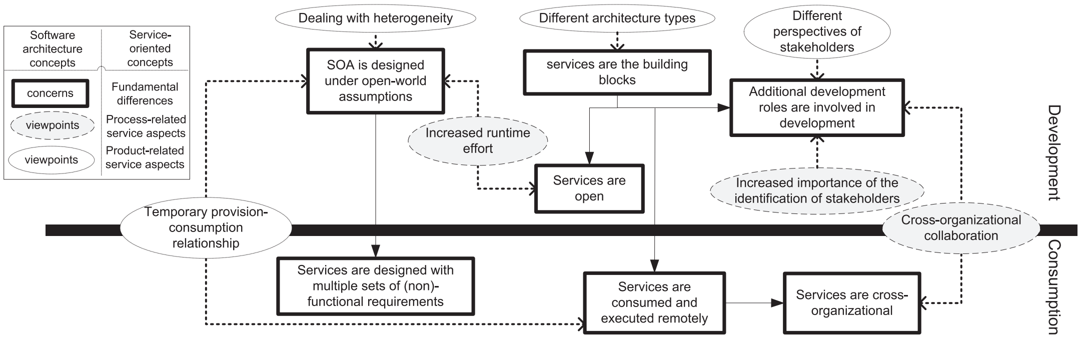
Software architecture concepts
Service-oriented concepts
concerns
Fundamental differences
viewpoints
Process-related service aspects
viewpoints
Product-related service aspects
Dealing with heterogeneity
SOA is designed under open-world assumptions
Different architecture types
services are the building blocks
Different perspectives of stakeholders
Additional development roles are involved in development
Increased runtime effort
Services are open
Increased importance of the identification of stakeholders
Temporary provision-consumption relationship
Cross-organizational collaboration
Development
Consumption
Services are designed with multiple sets of (non)-functional requirements
Services are consumed and executed remotely
Services are cross-organizational

Figure 1. A framework of SOA concerns and service-oriented viewpoints.

vices, composite services and service-oriented systems may indicate whether services are the building blocks of a service-oriented system.

- *Temporary provision-consumption relationship.* One of the open-world assumptions is not knowing the service consumers and their provider at design time. Further, services being consumed and executed remotely at the providers' side means that the provision-consumption relationship ends as soon as the execution of services is finished. Capturing this temporary provision-consumption relationship in making SOA design decisions means making open-world assumptions explicit and highlights the fact that services do not execute locally at service consumers.
- *Different perspectives of stakeholders.* An SOA design decision has different impact on different stakeholders. Explicitly capturing these impact in making SOA design decisions points out the additional stakeholders involved in development.
- *Dealing with heterogeneity.* Open-world assumptions suggest a heterogeneous environment where different data formats, protocols and technologies may coexist. Specifically considering heterogeneity in the SOA design means making open-world assumptions explicit.

In the field of SA, architectural concerns refer to the requirements of different stakeholders; and architectural viewpoints refer to conventions for constructing and using an architectural representation addressing these requirements [7].

Looking at the framework, we noticed that the fundamental differences can be seen as 'architectural concerns' or *SOA concerns*. As such, they resemble the focus or interest of architecting service-oriented systems. For instance, traditional architecting approaches, assuming e.g. the context of an application does not change or change slowly, are not completely valid in SOA design; instead the architecting of SOA focuses on making a set of design decisions so that the architecture in question is able to adapt to highly dynamic environments [10].

Similarly, the service aspects can be seen as 'architectural viewpoints' or *service-oriented viewpoints*. As such, they resemble a set of conventions for capturing these differences from both the engineering and architecture point of view. From the engineering perspective, in [5] we showed that SOSE methodologies provide better guidance on their application when three *process-related service aspects* are emphasized in associated process models where elements (e.g. distributed artifacts, activities requiring cross-organizational collaboration) that are missing or less stressed in traditional approaches are highlighted. From the architecture perspective, in [6] we discussed the relevance of paying particular attention to four *product-related service aspects* that aid the making of SOA design decisions and showed the related views in two exemplified SOA design scenarios.

4. The Need For Service-oriented Viewpoints

The advantage of representing the differences & service aspects in a framework and mapping them to architectural concerns & viewpoints is that the SOA concerns become visible, which points out where service-oriented viewpoints are needed.

Firstly, by looking at the concerns it is visible that they can be separated into two groups in principle: development- and consumption-related concerns. Although all the concerns are relevant to both development and consumption, the top four concerns illustrated in the framework are closer to the development point of view whereas the bottom three are described closer to the consumption point of view.

In the framework, there are more development-related concerns and viewpoints than consumption-related ones. Due to the fact that services are used by consumers but executed remotely, service consumers often "participate" in the development process in the sense that they play a critical role in e.g. service negotiation, service discovery and service composition. However, the existing approaches in SOSE and SOA have a much stronger focus on service provision whereas service consumption is paid less attention to. This observation is in line with the review of existing service life cycle models in [11], which points out that there is a lack of service consumer views in most of the existing models.

Secondly, the viewpoints developed in previous work already address part of the concerns identified here, which not only confirms the relevance of these viewpoints but also, more importantly, points out the need for more viewpoints.

Looking at the framework, we can see that neither the product nor the process viewpoints established so far address the concern of *multiple sets of (non)-functional requirements* for different groups of service consumers. This indicates a gap between the viewpoints established in our previous work and the SOA concerns.

We can also see that only two SOA concerns (*open-world assumption* and *additional development roles as stakeholders*) are addressed by both process- and product-related viewpoints. Being fundamental to SOSE, all of the SOA concerns in principle should be addressed by service-oriented viewpoints from both process and product perspectives. Therefore, we may conclude that future work on service-oriented viewpoints should explore how to address *services are the building blocks* and *services are consumed and executed remotely* from the process perspective and *services are open* from the product perspective.

Furthermore, the service aspects discussed so far are introduced from the perspectives of process and product only. More perspectives (e.g., business) can be studied in the future. As such, the SOA concerns can be addressed by full-scale viewpoints fulfilling needs from multiple perspectives.

To this end, we discussed several needs for SOA and identified an initial set of requirements for service-oriented viewpoints, which consists of highlighting consumption views, addressing the fundamental differences (SOA concerns), and capturing multiple perspectives. If specialized in industrial contexts and with specific SOA projects, we would finally be able to highlight aspects that are specific to services e.g. as discussed in [5]. As such, we could pinpoint as well as rationalize the need for different methods, techniques or tools introduced by services, and therefore exemplify 'real' differences, if any.

5. Conclusion

With the ultimate goal of understanding the innovation of SOSE and SOA, we studied the differences between SOSE/SOA and TSE/SA claimed in the literature and linked them to service aspects discussed in our previous work, as the first step towards this goal.

By constructing a framework illustrating the seven fundamental differences identified from the literature review and the service aspects, we noticed an analogy between them and architectural concerns & viewpoints. Such an analogy further led us to observe that visualizing SOA concerns facilitates the identification of the needs for service-oriented viewpoints. This observation guides us to explore more service aspects to address all the fundamental difference in our future work.

'Innovation' of SOSE and SOA is often claimed but remains abstract and fuzzy. In practice, what indeed should be done differently in engineering and architecting service-oriented systems is left implicit. Instead of focusing on the 'innovation' in terms of new issues that do not exist in TSE, herein we observe another way of interpreting 'innovation', which is the extent to which service aspects are capable of addressing the fundamental differences between SOSE/SOA and TSE/SA. In perspective, this work seems to scale down 'innovation' of SOSE and SOA, to 'relevance' of service-oriented aspects to engineering and architecture, which demands for further discussion in the research community.

References

[1] W. T. Tsai, "Service-oriented system engineering: A new paradigm," in *SOSE '05: Proceedings of the IEEE International Workshop*, Beijing, China, 2005, pp. 3– 6.

[2] L. Baresi, E. D. Nitto, and C. Ghezzi, "Toward open-world software: Issues and challenges," *IEEE Computer Society*, vol. 39, no. 10, pp. 36–43, 2006.

[3] "Software service engineering, Schloss Dagstuhl seminar 09021, www.dagstuhl.de," 2009.

[4] Q. Gu and P. Lago, "A model for exploring the service-oriented software engineering (SOSE) challenges," in *2nd European Young Researchers Workshop on Service Oriented Computing (YRSOC)*, 2007, pp. 44–49.

[5] Q. Gu, P. Lago, and E. D. Nitto, "Guiding the service engineering process: the importance of service aspects," 2009, 14 pages, under submission.

[6] Q. Gu and V. Hans van, "SOA decision making - what do we need to know," in *Proceedings ICSE Workshop on Sharing and Reusing Architectural Knowledge (SHARK 2009)*. IEEE Computer Society, 2009, pp. 25–32.

[7] ISO, "ISO/IEC 42010 - Recommended Practice For Architectural Description of Software-Intensive Systems," 2007.

[8] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: a research roadmap," 2006.

[9] Q. Gu and P. Lago, "Exploring service-oriented system engineering challenges: A systematic literature review," in *Service Oriented Computing and Application*. Springer Verlag, 2009, 28 pages, to appear.

[10] O. Zimmermann, J. Koehler, and L. Frank, "Architectural decision models as micro-methodology for service-oriented analysis and design," in *Proceedings of the Workshop on Software Engineering Methods for Service-oriented Architecture 2007 (SEMSOA 2007)*, Hannover, Germany, 2007.

[11] Q. Gu and P. Lago, "A stakeholder-driven service life cycle model for SOA," in *IW-SOSWE '07: 2nd international workshop on Service oriented software engineering*. Dubrovnik, Croatia: ACM, 2007, pp. 1–7.