

Quality Driven Web Service Composition Modeling Framework

Georgiana Stegaru, Cristian Danila, Ioan Stefan Sacala,
Mihnea Moisesescu, and Aurelian Mihai Stanescu

University Politehnica of Bucharest, Splaiul Independentei 313,
Bucharest, 060042, Romania

{georgiana_stegaru, sacalaioan, amstanescu}@yahoo.com,
{danila.cristian.84, mamihnea}@gmail.com

Abstract. Software as a Service (SaaS) has become the new paradigm in software development. Ecosystems of Web Services can be flexibly and dynamically composed to suit user specific needs. Quality is essential to the success of service based applications. The process of service composition has a major impact on the quality of the final product. We propose a dynamic and flexible Quality of Service Composition (QoS) model for the composed service and a conceptual framework for the service composition process.

Keywords: Service Composition, Quality of Service, Internet of Services.

1 Introduction

Rapidly changing Information technology (IT) is bringing radical changes and new opportunities in different aspects of our lives: economic, social and political. The Internet has evolved from a source of information to a critical infrastructure representing the ideal medium for the development of new services.

One area of research for Future Internet is Internet of Services where everything (e.g. information, software, platform and infrastructure) is available as a service. Software as a Service (SaaS) has become increasingly significant in the past few years due to the popularity growth of Service Oriented Architecture (SOA). SOA provides key concepts for Future Internet Enterprise Systems (FINES) such as service composability and reusability. Using today's broadband Internet connection Web-based Information Systems providers can develop complex, cross-organizational business solutions through Web Service compositions. In order to increase market share, organizations will need to develop competitive services that would differentiate themselves through high availability, flexibility, performance, more exactly through high quality. Quality assurance becomes a key factor to provide the desired end-to-end quality of distributed services. This requires not just an agreement on quality attributes, but also monitoring and control during runtime.

While much focus has been recently spent on services and service composition specifications, little research has been done in the area of service composition

validation and verification. The ability to correctly assemble new systems based on existing services strongly relies on the quality of each service and on the quality of the assembly. Research in this area has been mostly focused on Quality of Service (QoS) attributes that describe non-functional requirements, while functional specifications are supposed to be tested in-house by service providers.

A quality driven service composition should address quality from both the final product and process points of views. In this paper we propose a QoS (Quality of Service Composition) model for the quality of the composed service and a framework for the service composition process. Section 2 presents an overview of related work concerning the evaluation of the quality of service composition. In section 3 we present the QoS model and section 4 describes conceptually the framework for a quality driven service composition. We conclude and present our future developments in section 5.

2 Related Work

Innovative services are created by composing existing Web Services (WS). Web Service composition implies two important aspects: the specification by means of a composition language and the execution by means of an appropriate runtime. In the area of specification there are several existing process-oriented composition languages widely accepted by the industry and the researchers. [1]

Research in the service composition area recently shifted its focus from service composition tools and techniques to validation and verification of composed services. Verification of composed services borrowed testing principles from software testing and adapted them to SOA [2]. Strategies for testing services in isolation or in composition have been developed similar to the ones used in component-based systems. One approach for service composition unit and integration testing argues that choreography-based or orchestration-based testing must be done, according to the service view [3]. Dedicated tools are used for the validation of functional requirements for Web Services. One example is TGSE (Test Generation, Simulation and Emulation), a testing tool using a black-box approach for testing WS composition described in BPEL [4].

Most QoS driven service composition approaches focus on selection methods. The authors of [5] present such an approach, where the main features consist in a WS quality model and a quality driven service selection method. The WS quality model takes into consideration WS non-functional properties: execution duration, execution price, availability and reliability. Service selection is formulated as an optimization problem where linear methods are used to compute optimal service execution plans. Global optimization techniques and local selection methods can be used to find the best combination of services [6].

Other researchers consider a different approach that is more user-oriented [7]. They propose a framework for Quality of Experience aware service composition for Future Internet that allows users to compose service templates and coordinate network and computing resources in order to meet user requirements.

Another field of research is concerning the composition of services using Service Level Agreements (SLAs). A SLA driven approach proposes a model-based framework that autonomously builds services with a guaranteed level of quality [8]. There are also a number of works focused on QoS management of Web Services which can be classified by the methods used for monitoring and controlling the QoS properties of services: policies [9], [10], contracts [11], agents [12], [13].

State of the art approaches address quality only from the point of view of the final product i.e. the composed service. The focus is set, in most cases, on the selection of different services based on their non-functional properties. But there are process related quality indicators that have an impact on the final product, such as interoperability or adaptability. Also most approaches are context-dependent and cannot be applied to different business scenarios. They consider a certain domain where the services are running, or a certain specific language of the service composition (e.g. BPEL). We argue that there is a need for a Quality of Service Composition model that addresses both product and process quality and we present it in the following section.

3 Quality of Service Compositions Model

The Quality of a Service Composition (QoSC) is a key factor to the success of a business process. State of the art approaches focus mostly on non-functional requirements of Web Services. A reliable service composition also requires service interoperability and adaptability as services are heterogeneous in nature. Therefore, there is a need for quality assessment models that not only evaluate the performance of the composite service but also cover important aspects such as: interoperability, adaptability and security. We propose a hierarchical model for quality driven service composition that takes into consideration all these factors.

The model has three quality dimensions which assess the composed service from a certain point of view: business, operational or systemic. Each quality dimension is composed of one or more quality aspects. For each aspect we consider one or more quality items, which are tightly coupled with service characteristics and can be measured based on a quality criteria. The measurement can be either qualitative or quantitative. The three quality dimensions are: Business, Operational and System as represented in figure 1. The business quality dimension describes quality aspects that belong to the business value of service compositions. The operational quality dimension describes quality aspects related to interactions between services while the system dimension includes quality aspects that apply to the entire service composition.

Business quality dimension contains the following quality items: reputation (service provider reputation and service reputation), service affordability, service discoverability, service penalty and incentive, service usability and governance. These items can be qualitatively evaluated based on a history of the service usage experience and user feedback.

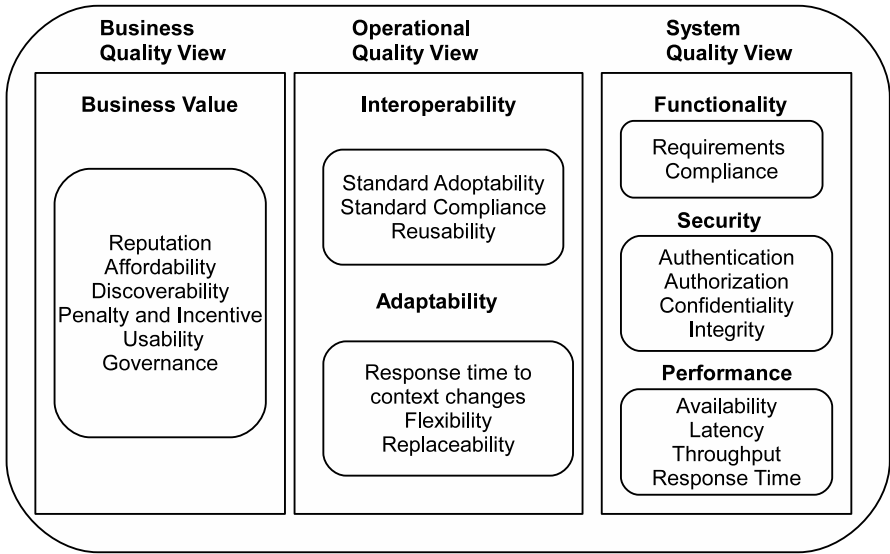


Fig. 1. Quality of Service Composition Model

Operational quality model includes service interoperability and service adaptability quality aspects. Interoperability quality aspect is concerned with the evaluation of the compatibility level between Web Services. Although WS technologies have been standardized, Web Services are still developed on different platforms and according to different specifications. Standards adoptability and standard compliance are quality items that evaluate the message exchangeability and conformity to a certain standard in order to make the composition inter-operable. Reusability defines the extent to which a Web Service can be used in another composition. Adaptability quality aspect evaluates the ability of a service to respond to external stimuli. Adaptability implies three abilities: the ability to recognize an environmental change, the ability to determine the change to be applied to the service and the ability to effect the change. These abilities reflect a Web Service’s flexibility and replaceability. Based on metrics defined in [14], we can evaluate the adaptability of a Web Service.

System quality model evaluates the composite service compliance with user defined requirements both functional and non-functional. It contains the following quality aspects: service functionality, service security and performance. The quality of the security incorporated into the service has the following quality items: authentication, authorization, encryption, non-repudiation, audit and integrity. The performance quality aspect of the composite service can be composed of quality items like: response time, throughput, availability, accessibility, latency, accuracy. Metrics for security and performance evaluation have been discussed in [15], [16].

The list of quality items is not exhaustive, but only describes most common quality features desired in a Web Service. The QoSC model includes not only product related QoS characteristics but also process specific QoS attributes. The quality model can be easily extended with more quality items according to the context of the service

composition. Domain-dependent quality items can be added for complex domain specific service based applications.

In the following section we present a conceptual framework to achieve quality driven service composition based on the QoSC model proposed earlier.

4 Framework for Quality Driven Service Composition

Technological, social and behavioral factors are driving businesses on the online market. Web-based Information Systems (WIS) are at the core of the networked devices, organizations and people. Adaptable WIS must support high service customization in order to be accepted by any user. Consumers must choose from an increasing number of services providing similar functionalities. Quality of Service Composition has a great impact on the composed service; therefore we need to address not only the quality of the final service but also the quality of the service composition process.

Based on the analysis on different service composition methods presented in [17], we identified the main phases of a service composition process. In figure 2 we present the model of a generic service composition process, developed using the business process modeling standard, Business Process Model and Notation, BPMN.

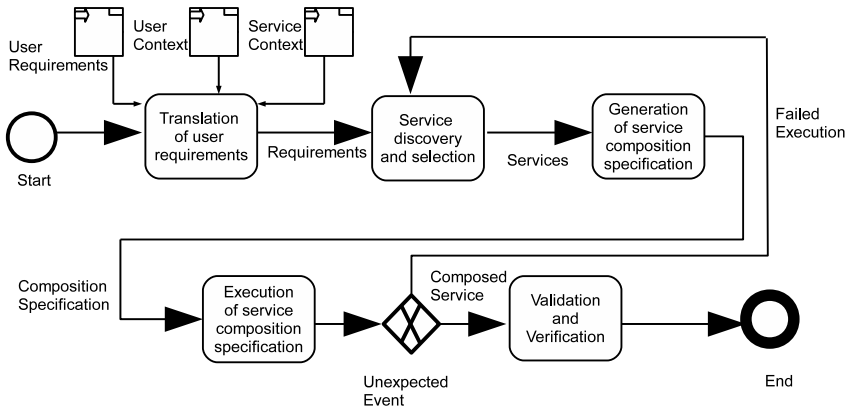


Fig. 2. Quality driven Service Composition Process in BPMN

4.1 Translation of User Requirements

First requirements are collected from the service request initiated by a user. A configurator based on dynamic questions should be used to capture user preferences. Then the user's requirements and preferences should be mapped to QoS properties, both functional and non-functional. At this phase we must also consider user context (e.g. location, request scope and purpose of usage, software and hardware capabilities) and service context (e.g. provider, availability, location, description,

SLA). By analyzing these inputs we identify the quality aspects and quality items at the business, operational and system levels based on the QoSC model in section 3. The result is the service selection specification which includes QoS properties, both functional and non-functional. The QoS properties can be explicit, stated by the user, and implicit, derived from the context.

4.2 Service Discovery and Selection

Service discovery and selection processes are based on the QoS criteria identified at the previous phase. Service registries are queried to identify services with required functionalities. To select the best matching service accordingly to QoS specification, each candidate service should be verified to ensure that it matches both functional and non-functional requirements (e.g. performance, security). The candidate services should be kept into a sorted list according to the probability for their selection.

In order to prevent possible issues at the execution phase, interoperability and adaptability of services involved in the composition must be assessed. The proposed QoSC model captures these requirements in the operational quality dimension. An approach to measure service interoperability is to verify the degree of adoptability and compliance to industry interoperability standards and guidelines. Service adaptability is necessary in pervasive environment where devices come and go, but also when a significant mismatch occurs between the supply and demand of a resource: such as network problems, service disconnection, change of service, new service available, or change in user requirements. Adaptability for a service can be calculated as the division between the number of dimensions where the service is capable to detect and react to a change and the total number of dimensions taken into account.

4.3 Generation of Service Composition Specification

After having identified the services that match the user requirements, both explicit and implicit, as captured in the first phase, the specification of the service composition must be developed.

The service composition specification depends on the composition method. The most common are service orchestration and service choreography. Service orchestration describes interactions at the message level in a service composition using a central coordinator. In contrast, service choreography describes from a global perspective the interactions between services participating in a business process. Service composition requires collaboration between services involved in the composition, thus interoperability and adaptability at the service level is needed. To prevent mismatching service interfaces or lack of compatibility between messages exchanged in the execution phase, we address these requirements at the service selection phase.

4.4 Execution of the Service Composition Specification

Finally the builder executes the service composition specification and produces an implementation corresponding to the required composite service. Depending on the service composition language, the WS-CDL specification or WS-BPEL is executed.

The SLA of the composite service is also produced during this step. The SLA of the composite service will specify the agreed levels of quality for the composite service. In case of a possible infrastructure/service problem the system returns to step 2: Service discovery and selection, and replaces the affected service with the next available service from the list. Steps 3 and 4 are executed again.

4.5 Verification and Validation of the Composed Service

Verification and validation should be realized during the entire process, from user requirements gathering and service selection to execution of the service composition. Checkpoints must be implemented at each phase to verify if it is successfully completed, before going to the next one. Fail-safe mechanisms need to be established in case of failure during the execution phase.

The new value-added service must be verified to ensure its correct functioning according to its specification. In the literature there have been several approaches to verification of service composition, either service orchestration [18] or service choreography [19].

Monitoring capabilities should be employed during the entire service composition process, from the requirements gathering phase to the execution of the service composition specification. However the service should be monitored also during its functioning to ensure that it runs in the desired parameters specified in the SLA.

5 Conclusions

Software as a Service has become the new paradigm in software development allowing organizations to achieve their business goals in a flexible manner that only requires a connection to the Internet. This new paradigm brings benefits but also new challenges regarding the Quality of Services and Service Compositions. Different quality aspects were covered in the literature, but there isn't a standard quality model for service compositions.

The proposed QoSC model addresses quality from a business, operational and systemic view, including both functional and non-functional requirements. However, our approach considers the quality of the final product (the composed service) and also the quality of the service composition process. We provide a framework to achieve quality driven service composition, independent of the composition language or platform. The framework addresses issues that might have an impact on the quality of the composed service, from user requirements gathering to execution of the service composition specification. Using state of the art technologies and techniques, the proposed QoSC model and framework can be adapted to different service composition types in different domains, such as Semantic Web or Artificial Intelligence.

Future work consists in the implementation of the framework on a real business use case in order to validate the proposed QoSC model.

Acknowledgments. The work has been funded through the European Social Fund, Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection, Financial Agreement POSDRU/107/1.5/S/76813.

References

1. Baryannis, G., Danylevych, O., Karastoyanova, D., Kritikos, K., Leitner, P., Rosenberg, F., Wetzstein, B.: Service Composition. In: Papazoglou, M., et al. (eds.) *Service Research Challenges and Solutions for the Future Internet*. LNCS, vol. 6500, pp. 55–84. Springer, Heidelberg (2010)
2. Harris, T.: SOA Test Methodology (2007), <http://www.thbs.com/soa4>
3. Bucchiarone, A., Megratti, H., Severoni, F.: Testing Service Composition. In: *Proceedings of ASSE 2007, Mar del Plata, Argentina (2007)*
4. Cao, T.D., Patrick, F., Castanet, R., Berrada, I.: Testing Service Composition Using TGSE tool. In: *IEEE 3rd International Workshop on Web Services Testing*. IEEE Press, Los Angeles (2009)
5. Zeng, L., Benatallah, B., Dumas, M.: Quality Driven Web Services Composition. In: *Proceedings of the 12th International Conference on World Wide Web, WWW 2003*. ACM Press, New York (2003)
6. Alrifai, M., Risse, T.: Efficient QoS-aware Service Composition. In: *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, New York, USA (2009)*
7. Park, J., Lee, H.Y., Yi, D., Kim, J.: QoE Dynamic Service Composition for Immersive Media-oriented Services. In: *Proceedings of 3rd International Conference on Future Internet Technologies, New York, USA (2008)*
8. Di Marco, A., Sabetta, A.: Model-based dynamic QoS-driven service composition. In: *Proceedings of QUASOSS@MODELS (2010)*
9. Badidi, E.: A Scalable framework for Policy-based QoS management in SOA Environments. *Journal of Software* 6(4) (2011)
10. Farroha, B.S., Farroha, D.L.: Policy-based QoS Requirements in a SOA Enterprise Framework – An Investigative Analysis. In: *Proceedings of 3rd Annual IEEE Systems Conference*. IEEE Press, Vancouver (2009)
11. Gwyduk, Y., Tsai, W.-T., Bai, X., Min, D.: Design of a Contract-based Web Services QoS Management System. In: *Proceedings of 29th IEEE ICDCS Workshop*. IEEE Press, Washington (2009)
12. Nyguen, X.T., Kowalczyk, R.: Agent-Based QoS Conflict Mediation for Web Service Compositions. In: *Proceedings of the IEEE/WIC/ACM, IAT 2006, Washington, USA (2006)*
13. Malak, J.S., Mohsenzadeh, M., Seyyedi, M.A.: Web Service QoS Prediction Based on Multi Agents. In: *Proceedings of ICCTD 2009, Kota Kinabalu, Malaysia (2009)*
14. Chung, L., Subramanian, N.: Process-Oriented Metrics for Software Architecture Adaptability. In: *Proceedings of 5th International Symposium on Requirements Engineering*, pp. 310–320. IEEE Press, Toronto (2001)
15. Ran, S.: A model for Web Service Discovery with QoS. In: *ACM SIGecom Exchange*, vol. 4(1), ACM Press, New York (2003)
16. Gollmann, D., Massacci, F., Yautsiukhin, A.: Quality of Protection: Security measurement And Metrics. In: Gollmann, D., Massacci, F., Yautsiukhin, A. (eds.) *AIS*, vol. 23. Springer, Heidelberg (2006)

17. Kapitsaki, G., Kateros, D.A., Foukarakis, I.E., Prezerakos, G.N., Karlamani, D.I., Venieris, I.S.: *Service Composition: State of the art and future challenges* (2007), <http://www.cin.ufpe.br/~redis/intranet/bibliography/middleware/kapitsaki-composition07.pdf>
18. Wang, Y., Wang, L., Dai, G.: A Web Service Orchestration Model Based on Concurrent Transaction Logic. In: *Proceedings of 7th International Conference on Grid and Cooperative Computing, GCC 2008*, pp. 475–482. IEEE Press (2008)
19. Zhou, L., Xiao, H., Ping, J., Pu, G., Zhang, H.: Simulation and validation of Web services choreography. In: *Proceedings of IEEE International Conference on Service-Oriented Computing and Applications, SOCA*. IEEE Press (2009)