Towards a Method for Service Design

Olga Levina, Trung Nguyen Thanh, Oliver Holschke, and Jannis Rake-Revelant

Berlin Institute of Technology Franklinstr. 28/29, 10587 Berlin, Germany
{olga.levina; oliver.holschke, jannis.rake@sysedv.tu-berlin.de}

Abstract: Services are the vital part of a service-oriented architecture. Their development and design are essential parts of the development and implementation of a service-oriented architecture. Thus, numerous approaches in research and practice exist that refer to different aspects of service design. These are focused on specific needs or aspects in service design. According to the literature review provided in this paper, no single service design approach covers all the aspects that are needed for the implementation and deployment of a service-oriented architecture. Beside the literature review this paper provides a service design approach that combines the existing methods and approaches. The goal is its further development towards a service design method for service design in research and industry.

Keywords: method design, service design, literature review, service-orientation

1 Introduction

Research activities and practice implementation attempts in the area of Service-Oriented Architecture (SOA) have gained a lot of momentum in the resent years. Methods for SOA development and evaluation started to emerge. Although, despite the evident progress there is still a lack of widely spread methods that can be used in every stage of SOA development and implementation.

SOA is defined here according to OASIS as “a software architecture of services, policies, practices and frameworks in which components can be reused and repurposed rapidly in order to achieve shared and new functionality”[1]. This definition includes one of the main characteristics of SOA: the reuse of the software components, i.e. services. A service can be further described as an element that encapsulates a business function and cannot be further decomposed without harming its functionality. Services can be defined as autonomous, platform-independent entities that can be described, published, discovered and assembled; they are technologically neutral, loosely coupled and support local transparency encapsulating business functionality[2]. Services can be differentiated among others by their goal, functionality, granularity, scope, and interaction.

Service design specifies how the service can be described and therefore found, which business operations underlie the service function, which SOA design principles are supported and which technologies are needed to implement the service. Service

design principles are embedded within the general principles of service-orientation that include: Service reuse[3,6,7], formal contract[3,8], lose coupling[2,3], abstraction[3,7,9], composability[3,7], autonomy[3,7], statelessness[3,6], discoverability[10].

This paper presents a generalized approach to service design that is based on existing rules for service design and SOA design patterns. The suggested framework aims to support service design by providing a certain procedure and detailed activities in order to enhance the development procedure and to increase the probability for service reuse. The paper is structured as follows: First, the existing service development approaches and methodologies both in research and practice are reviewed. The results of the analysis serve as basis for the development of a canonical framework for service design. Discussion of the results and outlook finish the paper.

2 Existing Approaches to Service Design

Approaches or service design methods reviewed here were chosen using literature research on SOA or service design topic. Their evaluation was not completely based on the method components suggested by [11], because not only service design methods but also approaches explicitly situated in the SOA context were reviewed. Further criteria applied were completeness [11], i.e. the necessary description of the activities, elements and tools as well as realization of the components described in [11], identification process [4], goal reference as well as context of the method (business or software development) [11] and consistency referring to the temporal and logical dependencies between the suggested process steps.

There is little consensus on general service design principles in research and practice. Though, the basic principles for service design can be considered as being: interface orientation, interoperability, modularization and process orientation, e.g. [3, 7-9]. Thus, the service design method or approach needs to include specification on interface design of a service. Besides the technical specification, meta-data on service content and use need to be specified. Communication aspects such as message formats, protocols and addresses need to be included as well as the effect of the service on data, process composition, security and further run-time aspects. Deployment of the interface design needs a service level agreement (SLA) as a realization of the abstraction principle of SOA, as well as its contents need to be included into service design approach. A further advantage lies in the changeability of the software elements.

Interoperability can be defined as the ability of software elements to exchange and interpret information with each other [13, 14]. Modularization implies composition of application systems or business processes into domains or services. Services are supposed to provide flexible support for business process automation. A service design approach or method need to provide guidelines for service granularity of data, business logic as well as functionality.

In the following existing service design approaches or methods are evaluated referring to the above mentioned criteria. The SOA Approach (SOAA)[3] describes top-down and bottom-up service design using examples. It considers aspects of

business engineering as well as technical aspects. Design and development of data models and interfaces are described as well as service design pattern are suggested. The service design takes all the basic SOA design principles such as autonomy, lose coupling, statelessness, etc. into account. SOAA also uses industry standards such as XMPL, SOAP, BPEL and UDDI [15, 16].

The Service-Oriented Design and Development Methodology (SODDM)[5] is based on the RUP-Method and analyzes the business requirements for service design. SODDM is based of Web Service[17] technology and does recommend specific vendor tools for service realization. Though the approach is described as a methodology it does not offer a role specification for the design process. The aspect of service orchestration is not further elaborated and service design is concentrated on a fine granular level. The SOA Method (SOAM)[4] puts the focus on the business requirements and business-oriented service realization. Bottom- up as well as top-down service design patterns are conceptualized independently from their domain. SOAM considers the main SOA construction principles, provides a domain model and service classification. This aspect supports a better service reuse and flexible service granularity. Service and interface implementation are mentioned referring to standards such as ebXML, UN/CEFACT[18], etc. for reducing the number of data attributes.

Method For Component-Based And Service-Oriented Systems Engineering (CBSOSE)[20] puts the focus on service identification according to business requirements, business engineering as well as software engineering. In software engineering the focus lies on component-based software development. CBSOSE offers an exhaustive documentation containing action description supported by examples and expected results. Services are modeled using Unified Modeling Language (UML), service granularity is based on business activities. Service reuse is not explicitly considered. The Process Model for Business Service Development (PMBSD) [21] identifies business services according to business requirements. PMBSD provides a role model and generic description of the process activities and results. The SOA design principles are not explicitly mentioned but SLA, reuse and discoverability of services are considered.

Web Service Implementation Methodology (WSIM)[22] focuses on SOA development using Web Service technology. Business engineering is not considered in the methodology though a role model is given. There are any tools specified that can support the methodology in service design. The design principles are not mentioned explicitly but reuse, interoperability and service orchestration are considered. Every phase of the method is supported by best practices. Executive's Guide to Service-Oriented Architecture (EGSOA)[7] approach includes bottom- up and top-down strategy for service identification using business services as well as legacy systems. The focus is on the business aspects of service development. There is no role model present in the method and no examples are provided. The analysis and design phase support the design principles such as SLA, interoperability, reuse, lose coupling and discoverability of the services.

3 Framework for Service Design – Towards a Service Design Method

The suggested framework for service design is based on the SOAM and SOAA approaches described above. SOAM provides a clear process for service identification and system analysis on the business side, while SOAA provides comprehensive description of technical service design.

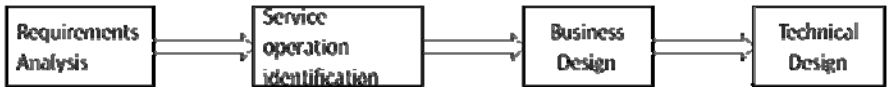

Fig. 1. Basic steps of the service design process

The abstract design process can be divided into four main phases: requirements analysis, identification of service operations, business design, and technical design (see figure 1). The approach includes three actors: Process Owner, Business Process Analyst and IT Developer. The process owner is involved into the process operation, business analyst is a member of a (internal) consulting team or a team that is concerned with business process management tools that are used e.g. for process modeling, and workflow systems. IT developer transforms business requirements into software components. The first two phases are conducted by the process owner and business process analyst. The last two phases are completed by business process analysis and IT developer.

The purpose of the approach is to support business process automation, and integration of technical innovations. Thus, the main sources for the requirements analysis are the business process models and descriptions. The identified operations need to be mapped to the service model. This model is transformed into a service description including interface definition in a WSDL-file.

In the requirements analysis phase business requirements that need to be realized using services are elevated. Here the purpose of the service is defined: encapsulation of a legacy system or business process support. Whether it is possible to support a business process in a SOA is evaluated according to the criteria proposed in [9].

Additionally to the evaluation the business processes need to be captured. As documentation method Business Process Modeling Notation (BPMN) is suggested here as being both easily adoptable to business requirements and providing necessary information for technical implementation. Documentation level is defined by the activities that are refined to the level where further division does not provide any sensitive process information. Besides the activities necessary data object types need to be identified and enriched with accordant attributes. For data object type documentation the UML class diagram is suggested. Resulting process models can be reviewed and examined using the ebXML or RosettaNet[23] standards. OASIS and UN/CEFACT [18] provide context categories for relevant attributes. This information can be used to reduce the information amount.

During the legacy systems analysis application systems supporting the business process are identified. Operations, data objects and interfaces are identified and

documented. Legacy systems can be encapsulated using the following possibilities [4]: APIs, Web Services, Data bases, Object oriented software classes, etc.

For business design of the services classes of service candidates need to be identified. This differentiation allows the application of service design principles to each service. Here basic, process-oriented, public enterprise and semi-services are taken into account [8]. During the business-oriented service design identified operations need to be analyzed regarding their granularity and grouped to service candidates. Operations can be grouped according to the user rights or to the data object type. Third possibility includes operation grouping according to their task or operational domain. Documentation of the identified operations and the grouped services can be achieved using a simple table tool.

During the technical service design a technology for service realization needs to be chosen. Here the Web Service technology is suggested due to its popularity and a quasi-standard status in research and practice. Basic Profile including WSDL 1.0 [24], UDDI 2,0, SOAP 1.0, XML 1.0 [25], XML Schema 1.0 can be used for service specification. It can be supported by the use of the WS- extension frameworks. Development patterns that can enable statelessness on the technical level are e.g.: Atomic Service Transaction, State Messaging, etc.

4 Discussion and Outlook

The presented approach provides a standardized procedure to service design deriving its concepts from existing methodologies and technological approaches. The method needs to be enriched with approximate time-levels for each of the phases as well as a detailed result description for the results of each phase. The roles concept cannot be précised any closer as it often depends on the given circumstances in the enterprise. The suggested role concept assumes C-level management support for the SOA implementation as well as know-how in business (process) analysis and documentation methods. Communication and information structures need to be specified in detail to allow management and control of the implementation initiative. Another assumption here is the cooperation of the business and IT departments that is facilitated by a mediation team of interdisciplinary analysts.

The presented approach can be evaluated using the criteria for method description [11] with the result, that the approach cannot yet be referred to as a method, as the constructs used and principles of form and function are not elaborated extensively. Testable propositions also need to be defined and applied in a subsequent case study.

Hence, to be able to be referred to as a method the approach has to provide some considerable aspects such as construct definition and description, the review of principles of form and function as well as testable propositions. These aspects will be deepened in the future work. The founding will be applied on several case studies so that the method can be further refined and made more feasible for practitioners and researchers.

References

1. OASIS. *Reference Model for Service Oriented Architecture 1.0* 2006 [cited; Available from: http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf.
2. Papazoglou, M.P. *Service -Oriented Computing: Concepts, Characteristics and Directions*. in *Proceedings of the Fourth International Conference on Web Information Systems Engineering*. 2003.
3. Erl, T., *Service-Oriented Architecture: Concepts Technology, and Design*. 5 ed. 2005, New Jersey: Prentice Hall PTR.
4. Offermann, P., Bub, U. *A Method for Information Systems Development According to SOA*. in *AMCIS 2009*. 2009.
5. Papazoglu, M., Heuvel, W.-J. van d., *Service-Oriented Design and Development Methodology*. International Journal of Web Engineering and Technology 2006: p. 412-442.
6. Josuttis, M.N., *SOA in Practice*. 1 ed. 2007, Sebastopol: O'Reilly.
7. Marks, E.A., Bell, M., *Service-Oriented Architecture: A Planning and Implementation Guide for Business and Technology*. 1 ed. 2006, Hoboken: John Wiley and Sons Inc.
8. Krafzig, D., Banke, K., Slama, D., *Enterprise SOA: Roads and Best Practices for Service-Oriented Architectures (in German)*. 1 ed. 2007, Heidelberg: REDLINE GmbH.
9. Heutschi, R., *Serviceorientierte Architektur* 2007, Berlin: Springer.
10. Fringer, P., Zeppenfeld, K., *SOA and Web Services (in German)*. 1 ed. 2009, Heidelberg: Springer.
11. Offermann, P., Blom, S., Levina, O., Bub, U., *Proposal for Components of Method Design Theories*. Wirtschaftsinformatik/Business & Information Systems Engineering, 2010. **52/2**(5/5): p. 287-297/295-304.
13. Beverungen, D.K., R.; Müller, O., *Development of Service-Oriented Architectures for Manufacturing and Service Integration (In German)*. 2008.
14. Thomas, O., Leyking, K.; Scheid, M.: and S.-. 2009, *Prozess Models for Service-Oriented Software Development (in German)*. Wirtschaftsinformatik 2009: p. 181-190.
15. OASIS. *OASIS Standard 20041: Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)* 2004
16. OASIS. *Web Services Business Process Execution Language Version 2.0 OASIS Standard 11*. 2007 [cited.
17. W3C. *Web Services Architecture, 2004*. 2004
18. UN/CEFACT, *UN/CEFACT's Modeling Methodology (UMM) UMM Meta Model – Base Module Candidate for 2.0*
Public Draft, C. Huemer, Editor. 2008.
19. Mayerl, C., Link, S, Racke, M., Popescu, S., Vogel,T., Mehl, O., Abeck, S., *Method for Design of SLA-enbaled IT-Services (in German)* in *Communication in Distributed Systems*, P.G. MÜLLER, R.; SCHMITT, J. B., Editor. 2005, Springer.
20. Stojanovic, Z., *A Method for Component-Based and Service-Oriented Systems Engineering*. 2005.
21. Stein, S.I., K. . *Process Model for Business Service Development*. 2006
22. Lee, E.W., Haines, M., Chan, L. P., Ang, C. H., Tan, S. P., Lee, H. B.,Cheng, Y., Xu, X., Yin, Z. . *Web Services Implementation Methodology*. 2005 [cited; Available from: http://www.oasis-open.org/committees/download.php/21354/fwsi-im-1.0-guidelines-doc-wd-PublicReviewDraft1.0.pdf.
23. RosettaNet. 2010 [cited; Available from: http://www.rosettanet.org/.
24. W3C. *Web Services Description Language (WSDL) 1.1*. 2001
25. W3C. *Extensible Markup Language (XML)*. 2003