

Value-Aware Service Model Driven Architecture and Methodology

Xiaofei Xu¹ and Zhongjie Wang²

¹School of Computer Science and Technology, Harbin Institute of Technology, China. xiaofei@hit.edu.cn

²School of Computer Science and Technology, Harbin Institute of Technology, China. rainy@hit.edu.cn

Abstract: Fully realizing proposed “value” for both customers and providers is considered as an ultimate goal of developing a good service system. In this paper, based on the mechanism of “value co-production” in services we propose a value-aware service methodology based on Service Model Driven Architecture and Service Quality Function Deployment. Such methodology integrates top-down service model transformation and bottom-up service component reuse together, to help service designers be fully aware of service values, i.e., how “value” is defined, decomposed, transferred and transformed in the lifecycle of a service system. Aiming at some key decision-making points that have great influences on the delivery of service values, we also present some general ideas.

Keywords: service, value-aware, methodology, transformation

1. Introduction

Innovations on service business patterns have been a key force to accelerate GDP growth. By importing new information technologies, new management techniques, new resource configuration patterns and new specialized social division of labors, the invented service patterns are expected to produce new added-values. Specific service pattern describes how customers and providers co-produce value and share risk [1], whose innovations are that it could provide some new values that other service patterns cannot provide.

However, service patterns just make value proposition in high level. These innovative values should be elaborately defined and refined in detailed service models (e.g., service process, configuration of resource and people, etc) during service de-sign, then step-by-step transformed into the infrastructural IT-based service systems. During the execution of service systems, these values are delivered to customers and providers, respectively.

Please use the following format when citing this chapter:

Xu, X. and Wang, Z., 2008, in IFIP International Federation for Information Processing, Volume 280; *E-Government, ICT Professionalism and Competences; Service Science*; Antonino Mazzeo, Roberto Bellini, Gianmario Motta; (Boston: Springer), pp. 277–286.

It is easily to see that, whether and to which degree the proposed values are to be effectively realized and delivered to customers and providers as expected, depend on the design quality of service models and IT-based service systems to a great extent. In practical services, because there might be some thinking gaps between service innovators, service designers and service system developers, the proposed values might not be fully top-down transformed and implemented [2], i.e., “good ideas cannot become good reality”, just as shown in Figure 1.

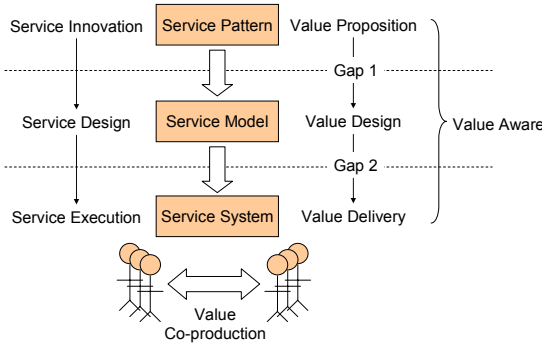


Fig. 1. Gaps between three phases of service lifecycle

In order to reduce or eliminate Gap 1, there should be a systematic service design method to transform service patterns into service models which have the ability of describing innovative service values [3]. By evaluating and optimizing such service models, we reach the objective of “be aware of values during service design, and make service models fully express service values”.

To reduce or eliminate Gap 2, there should be some service system design method to transform service models into IT-supporting service systems, i.e., selecting proper service elements (or called “service component”, “service asset”, including service behaviors, resources, people, technology, etc [4]) and compose them together according to some specific architectural styles. By evaluating and optimizing such service systems, we reach the objective of “be aware of values during service system development, and make service systems support value delivery as far as possible”.

In conclusion, initial proposed service values should be materialized in each phase of service lifecycles, e.g., modeling, design and implementation. In traditional service methodologies, they focus most of attentions on service functionalities while ignore “value”. For this issue, we do some research on value-aware methodologies for service innovation, modeling, implementation and optimization, i.e., “be aware of service values every when and everywhere”.

This paper is organized as follows. Section 2 explains some primary concepts relating to services. In section 3 we briefly introduce the framework of value-aware methodology containing five horizontal layers and three vertical threads. Section 4 and 5 introduces top-down and bottom-up approaches of the methodology, respectively. Finally is the conclusion.

2. Basic Concepts on Services

In recent years, “service” becomes more and more a hot topic in both industry and academia. In broad sense, a service is defined as the application of competences for the benefit of another, meaning that service is a kind of action, performance, or promise that’s exchanged for value between provider and customers [1].

Different with manufacturing in which values are transferred thoroughly from providers to customers, values in services are co-produced and shared between service provider and customers.

Each party of a service has his own expectations on values [5], which are further classified into more detailed types, e.g., (1) economic values; (2) improvement on knowledge or skills; (3) improvement on experiences; (4) improvement on market competitiveness; (5) physical values; etc. Some of them are tangible values while others are intangible ones. Based on a well-designed service system, these values would be delivered to related parties by co-production, and the degree of value delivery lies on the design and development quality of service models and service systems to a great extent.

To support the implementation of services, there must be corresponding service systems to compose various service elements together. A service system is considered as a complex socio-technological systems and is defined as “a value co-production configuration of people, technology, other internal and external service systems connected by value propositions, and shared information (such as language, processes, metrics, prices, policies, and laws)” [1]. Generally speaking, a typical service system is composed of the following service elements:

- People: service customers, service providers, including their organizations, roles, human resources, professional skills and capabilities.
- Resource: including technological resources and physical material resources, e.g., software, equipment, physical environment and IT infrastructure, etc.
- Shared information: files, data and documents that are created, transacted by and exchanged between service customers and providers.
- Behavior: physical, mental or social behaviors of people doing work in service systems. Such behaviors are usually stochastic and can’t be easily modeled and simulated.

Service methodology is defined as the method for systematical planning, developing and managing full lifecycle of service systems, to support service providers build well-designed services and service systems [6].

Strictly speaking, service methodology is such a technological way to describe, define, design, build, implement, provision, and dynamic evolution services and service systems, by importing and applying related theories and methods of services science, management and engineering, and providing enabling tools and platforms, thereby validating whether customer requirements would be satisfied on demand, and whether value would be properly co-produced.

3 A framework for value-aware service methodology

In this section we will give a brief introduction to the value-aware service methodology, which forms a two-dimensional framework shown in Figure 2.

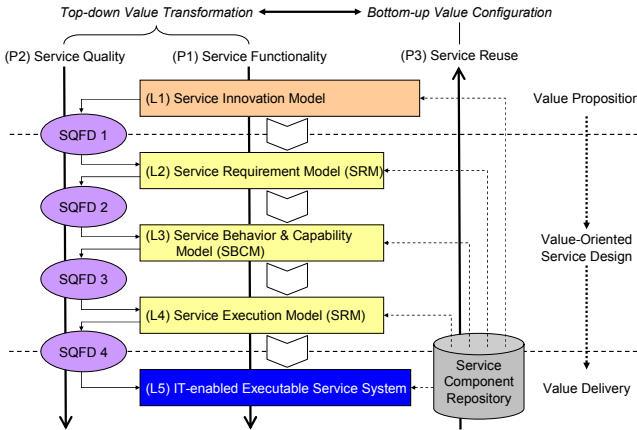


Fig. 2. Framework for value-aware service methodology

In horizontal dimension, multi-layered service models are developed for describing customer requirements of services, detailed behaviors of services, etc. There are five layers, i.e.,

- (L1) Service Innovation Model (SIM);
- (L2) Service Requirement Model (SRM);
- (L3) Service Behavior and Capability Model (SBCM);
- (L4) Service Execution Model (SEM);
- (L5) IT-enabled Executable Service Systems (SES);

L1 declares the objectives of services, especially what kinds of innovative value-added would be provided to customers and providers. Purposes of L2, L3 and L4 are design of the services, showing details of service behaviors and processes, service parties and people, service resources and their capability, and shared information. How proposed values are realized step by step in services is emphasized, too. L5 is actually not models but real executable systems in reality under the support of information technologies. It delivers values to each related party. Detailed specifications of these models could be found in our previous works, e.g., [7][8].

Above models describe contents of service in different phases of its lifecycle, and there are close associations between neighboring layers. Therefore in vertical dimension we design three inter-related paths to connect these models, i.e.,

- (P1) Top-down transformation of service functionality;
- (P2) Top-down transformation of service quality;
- (P3) Bottom-up service component based reuse.

P1 is the most fundamental transformation in our methodology. After SIM is gradually made clear, service designers begin to collect customer requirements and model SRM, which are then gradually transformed to SBCM and SEM, then to SES. In such procedure, each proposed value is allocated to one or more specific service functionalities and refined along with the top-down transformation.

However, P1 only concerns about how values are embedded in and supported by various layers of service models, it does not mention to which degree each value is to be implemented. In fact, every layer of service models is attached with a set of quality parameters (e.g., time, cost, price, etc) to quantitatively measure such degree, and such quality parameters should also be fully considered during functionality transformation. We use P2 for this purpose.

In P2, Quality Function Deployment (QFD) technique is imported to transform quality parameters of upper layer to the neighboring lower layer. P1 and P2 are interrelated closely.

From another point of view, most of services need not be completely developed from scratch, and there might be some existing service elements (e.g., people, service behaviors, resources, etc) that could be reused. In our methodology we import the third path P3, which is a bottom-up transformation. After a specific layer is modeled, service designers may pick out service components that most match with service models in both functional and quality aspects, then select a proper service architecture style to compose them together to implement service system. Objective of P3 is to find the best reuse solution to maximize the degree of implementing proposed values.

In conclusion, the ultimate goal of value-aware service methodology is to “be aware of values in each phase of service lifecycles and maximize them”. By means of such methodology, a service system can be developed on demand of customers, be implemented efficiently and run with good service performance. We will discuss detailed P1, P2 and P3 in Section 4 and 5.

4. Top-down: SQFD-based model transformation

Concerning P1 and P2, for arbitrary neighboring models there are three types of top-down transformation policies, i.e.,

- Direct mapping, i.e., a service element in upper-layer model is directly mapped and referenced in lower-layer model;
 - Refinement, i.e., after one service elements in upper-layer model is mapped to lower-layer model, it should be manually refined to describe more details of services;
 - Instantiation, i.e., an abstract service element in upper-layer model is instantiated to one or multiple concrete service elements.
- No matter what types of transformation, the following two aspects should be elaborately considered, i.e.,

- Service semantics driven transformation. Semantic information (e.g., re-resource availability, customer preferences, skills of providers, etc) attached to upper-level models should be fully utilized and transformed to lower-level models to make the models more feasible and close to reality. Ontology-based rules for semantics mapping are adopted for this purpose;
- Service quality based transformation. Quality requirements consisting of a set of quality parameters (e.g., SLA) are elaborately transformed between layers. QFD-based method is imported here [9].

The main reason we consider quality transformation along with functionality transformation is to lessen the gap between customer perceived service from the service system and their expectations. As core of QFD, the House of Quality is adopted as the tool for quality transformation. There are three-level HoQs to transform VOC to SRM, SBCM and SEM, respectively, as shown in Figure 3. Figure 4 shows an example of the first HoQ in Figure 3.

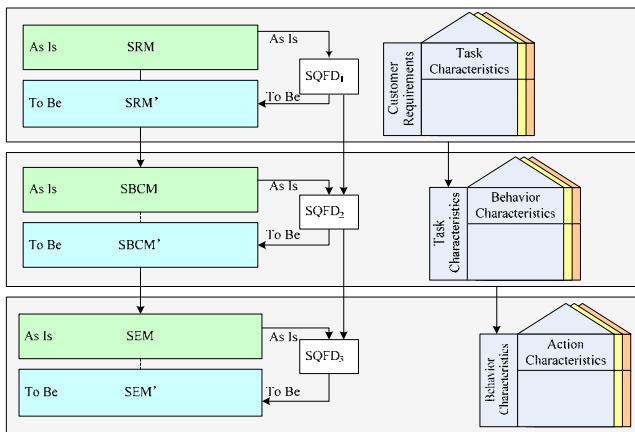


Fig. 3. Three-level HoQs in P2

However, even the designed service models and systems fully reflects customers' quality requirements, during the execution of service systems, there might still exist some gaps. Therefore run-time quality performance evaluation is necessary for future optimization. In run-time quality/performance evaluation phase, monitoring tools will be imported to automatically send queries and receive reports from different service component environments periodically. The system run-time Key Performance Indicators (KPIs) will be evaluated based on collected data. The results from run-time quality/performance evaluation are analyzed for further ser-vice system optimization.

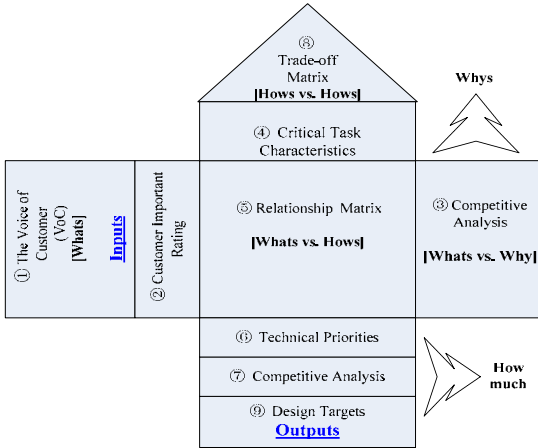


Fig. 4. Service House of Quality

Service quality/performance optimization is the third phase of P2. The gaps between customer expected service quality from SRM and customer perceived service quality from service system are identified, analyzed, and traced backward to early phase for reasoning the causes. In order to improve the quality of service system efforts are made to optimization by considering re-design, re-configuration, and or re-negotiation. This closes the loop of quality assurance for lifecycle.

5. Bottom-up: reuse-based service system development

Besides top-down transformation (P1 and P2), a bottom-up service reuse (P3) are important likewise. Basic idea of P3 is to fully make use of existing service elements to develop service systems instead of doing from scratch, so as to improve development efficiency, quality and to decrease cost.

Since service system is a complex socio-technological system which is composed of various elements, we classify service elements as the following five types, i.e.:

- People-ware SC: a service person with specific professional skills to provide specific behaviors during services.
- Software SC: a software entity with specific transaction or computation capabilities to provide specific behaviors in services, e.g., a web services with WSDL-based interfaces, an encapsulated legacy system, etc.
- Hardware SC: a hardware or equipment with specific capabilities, e.g., a computer server for residing software, an instrument for measuring and checking, a GPS for indicating directions, etc.

- Environment SC: a location with specific facilities as a container where ser-vice behaviors will take place, e.g., a meeting room with tables, chairs, projectors and whiteboards for training activity, etc.
- Behavior SC: processes, activities or actions that a person could behave to accomplish a service task, e.g., consulting, training, manipulating a machine, using a software system, reporting problems, etc.
- Information SC: a physical or electronic entity that contains data and is exchanged and shared among software systems, people, hardware, etc, e.g., a sales order, a log of call center, a service manual for guidance, etc.

If there has been a large service component repository, service modelers could leave the top-down process of P1 and P2 at any time and shift to the bottom-up P3, i.e., selecting proper SCs that most match with the service models and compose them together to form the service system. In this process, there are two decision-making issues that should be addressed, i.e.,

(DP1) Selection of service architecture styles (SAS)

As there have reached a consensus that there are some similarities between service system and software system, it is possible to use some software design principles and methods for reference in service system design, e.g., firstly dealing with architecture design by considering what (types of) service elements are required and how they are connected together, then carrying out detailed behavioral design for each service element. Here, architecture design is the most important step, where a high-level design solution is achieved by continuous decision-makings on some functional and non-functional dimensions aiming at specific features of ser-vice business.

Until now we have summarized tens of service architecture styles (SAS) already by investigating different service domains, including creational, structural and behavioral styles, e.g., Call-Return, Service Desk, Service Grid, Service Outsourcing, etc, and a comparison has been made to find that different SAS have different non-functional performance to support value delivery (detailed information could be found at [8]). Aiming at the features of a specific service system, an evaluation method is developed to prioritize various SAS to find the best one.

(DP2 & DP3) Selection of service components.

There might exist multiple service components which could provide the same service functionality that contains in specific layer of service models, however there would be difference on their non-functional performance (or, degree of value delivery). During SC selection, the first step is to identify each independent service elements from service models, then select a set of candidate SCs which would provide optimal values for each element (DP2). Because there would be some dependencies between different service components, it is necessary to look for a global optimized solution for the composition of above candidate SCs for

optimization of service value delivery (DP3). General process is shown in Figure 5.

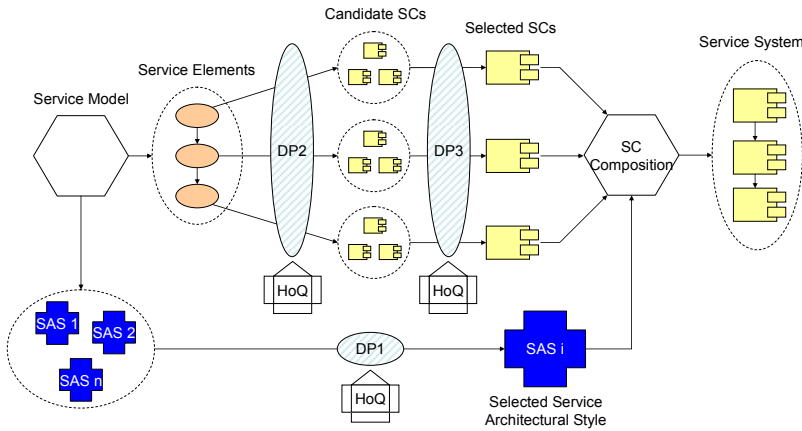


Fig. 5. Bottom-up: reuse-based service system development

During above process, the following matching and selection policies are considered to evaluate and prioritize SCs, i.e.,

- Whether and to which degree could the functionality of a SC satisfy requirements described in service models;
- Whether and to which degree could the non-functional performance (SLA) of a SC satisfy requirements described in service models;
- Whether is a SC usable, e.g., whether it satisfies time constraints, physical environment constraints, or connectivity constraints, etc;
- Whether and to which degree could a SC deliver the values that have been declared in service models;
- Whether and to which degree could multiple candidate SCs match with each other on functionalities, SLA, various constraints and value delivery.

We develop a QFD based decision-making algorithm to support above process. In this algorithm, input of House of Quality (HoQ) is a set of functional, non-functional, value and constraint parameters picked up from service models, and a set of candidate SCs. By specific decision making process, optimal SCs are selected as output of the HoQ.

6. Conclusions

Target users of such value-aware methodologies are large numbers of service providers or service designers. After they have formed some innovative ideas on

services, such methodologies would provide them a systematic approach to define and refine such services, until to build an executable service system.

Compared with existing methodologies on service system development, ours help service designers focus on “how to be aware of values (both customers and providers) as far as possible” and “how to improve service system’s capability to deliver values to each party right, rapidly and on demand”.

Acknowledgments

Research works in this paper is supported by the National Natural Science Foundation (Grant No. 60573086) and the National High-Tech Development Plan Foundation (Grant No. 2006AA01Z167).

References

1. J. Spohrer, P. Maglio, J. Bailey J and D. Gruhl. Steps towards a Science of Service Systems, *IEEE Computer*, vol. 40(1), pp. 71-77, 2007.
2. H. Cai. A Two Steps Method for Analyzing Dependency of Business Services on IT Services within a Service Life Cycle, *Proceedings of International Conference on Web Service*, pp. 877-884, 2006.
3. [M. Henkel, E. Perjons, J. Zdravkovic. A Value-based Foundation for Service Modelling, *Proceedings of the European Conference on Web Services*, pp. 129-137, 2006.
4. H. Akkermans, Z. Baida, J. Gordijn, et al. Value Webs: Using Ontologies to Bundle Real-World Services, *IEEE Intelligent Systems*, vol. 19(4), pp. 57-66, 2004.
5. C.B. Stabell and Ø.D. Fjellstad. Configuring value for competitive advantage: On chains, shops, and networks, *Strategic Management Journal*, vol. 19(5), pp. 413-437, 1998.
6. X.F. Xu, Z.J. Wang and T. Mo. An Introduction to Methodology for Service Engineering, *Computer Integrated Manufacturing Systems*, vol. 13(8), pp. 1457- 1464, 2007.
7. X.F. Xu, T. Mo and Z.J. Wang. SMDA: A Service Model Driven Architecture, *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications*, pp. 291-302, 2007.
8. Z.J. Wang, X.F. Xu and T. Mo. Service Architecture: High Level Descriptions of Service System, *Journal of Harbin Institute of Technology (New Series)*, vol. 15(Sup.1), pp. 7-12, 2008.
9. S. Liu, X.F. Xu and Z.J. Wang. SQFD: QFD based Service Quality Assurance for the Lifecycle of Services, *Proceedings of the Fourth International Conference on Interoperability for Enterprise Software and Applications*, 2008.