# Service Delivery Process Based on Service Composition Mechanisms

Thècle Alix and Bruno Vallespir

Univ. Bordeaux, IMS, UMR 5218, F-33400 Talence, France
CNRS, IMS, UMR 5218, F-33400 Talence, France
firstname.name@ims-bordeaux.fr

**Abstract.** Service engineering has merged to boost the development of services required by users to satisfy Quality of Service requirements. As there is a shift towards a service based economy planned to increase profit, more and more complex services are proposed by companies. At the same time customer needs due to numerous offers are ill-defined, change rapidly over the time and even does not exit on the market. The combination of existing services to build a new one is a possibility that is time consuming for the service consumer and necessitates expertise to make the combination successful. In IT, service brokers are used to provide services satisfying QoS requirements. In that paper we propose to use concepts from service engineering to define a service composition mechanism enabling the development of complex services regardless its type as well as a service delivery process modeling to support service production.

**Keywords:** service modeling, service composition, service engineering, service delivery.

## 1 Introduction

As the service sector expands into the global economy, service engineering develops as a field enabling service efficient innovation by applying scientific understanding and management practices to designing, improving, and scaling service systems. Service Engineering is becoming a hot topic even in manufacturing companies as the shift towards service-based economies is bringing new business concerns to focus.

Service Engineering is a comprehensive solution that delivers everything as a service. Based upon service components (human, machine, etc.) configured to work together in the service framework and deliver a service through a single system, service engineering is supposed to increase company profit, satisfy beneficiary requirements in terms of quality of service with less energy and resource consumption, unlike manufacturing sector. To achieve this goal, some considerations on beneficiary requirements, service added value, service availability, service customization, etc. might be taken into account. These considerations discussed in the next part show the importance of the relationship between the service provider and the service beneficiary to elicit the requirements and the difficulty for the former to

V. Prabhu, M. Taisch, and D. Kiritsis (Eds.): APMS 2013, Part II, IFIP AICT 415, pp. 226–233, 2013.

satisfy the latter with a level of quality that is requirement compliant as the service required is sometimes not available or even does not exist.

Service broker is a solution to decompose a service into sub services in the IT domain, allocate the existing sub-services in the service market and integrate them together to generate a new one, which can help greatly improving beneficiary experience and satisfaction. Here, we propose to use the concepts of service engineering, service composition to define a dynamic delivery process of complex services that can be IT independent.

The relationship between the provider and the beneficiary of the service is depicted in section three, through a service delivery process modeling, in its static and dynamic perspective. Section four details a service composition mechanism based on service engineering aiming to favor complex service delivery. The last section draws conclusions and highlights future works.

## 2 A Few Considerations to Make Service Engineering Efficient

### 2.1 Definitions and Service Engineering Insights

Service Engineering approach is global and encompasses the entire service trading process. It provides services to the mass, ranging from the end-users to enterprises. Service Level Agreements (SLAs) [1] including Quality of Service (QoS) requirements are set up between service consumers and service providers. An SLA is a commitment entered by a provider with regard to a customer that specifies the quality of service to be provided for a hardware or software, over a certain period of time, in terms of metrics agreed upon by all parties, and penalties for violating the expectations. SLAs act as a warranty for users.

A requirement is a statement that identifies a capability or function needed by a system in order to satisfy a consumer need [2]. The stakeholders include all the people, organizations, and institutions that are a part of the system environment because the system provides some benefit to them and they have an interest in the system. This includes end users, operators, bill payers, owners, regulatory agencies, victims, sponsors, maintainers, architects, managers, customers, surrogate customers, testers, quality assurance, risk management, purchasing, and the environment. As the relation is bijective, protagonists are subsequently shortly named provider and consumer. To make complex service delivery efficient, attention must be paid to the following assertions:

- Consumers always don’t know exactly what they want and their requirements can change rapidly over time.
- Different consumers have their own business logic that cannot be satisfied by the existing services in the market or the existing ones are too expensive to afford.
- Consumers (like small and medium companies, individuals) cannot afford customized services whose price is often expensive and time consuming.
- Providers could not provide specified services to every consumer, as it is not realistic. They focus on facing a large group of population and try to satisfy requests from majority of them.

## 2.2 Requirements Categorization

Requirements are categorized in several ways [3]. The following are common categorizations of requirements that relate to technical management: Customer Requirements, architectural requirements, structural requirements, behavioral requirements, functional requirements, non-functional requirements, performance requirement, etc. Among them, service requirements are more related to functional and non-functional types.

Functional requirements specify particular results of a system (specific behavior or functions). This should be contrasted with non-functional requirements that specify overall characteristics such as cost and reliability. Often called quality of service requirements, non-functional requirements can be divided into two categories: execution qualities and evolution qualities. Broadly, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. The knowledge of these two requirements is mandatory to reach the service engineering abovementioned objectives.

## 2.3 Requirements Elicitation as a Pre Requisite

Service requirements truly represent the business logic and consumer perspective. When a service is desired, a good understanding of the goals, motivations and needs consumers have must be established. The latter are difficult to predict and mostly consumers are not consciously aware of those needs [4].

Services so far are provided by organizations and are thought through and planned (designed) mostly from this provider perspective. The point of difference that the Service Engineering approach offers regarding more traditional IT service design approach is the development of service systems focusing on the consumers as well as on the organization. Consumers are involved in designing the service delivery process; without their participation there can be no service. Over time, they encounter different touch-points. A touch-point is a contact point with one of the elements of the service offering. All touch-points can be considered experience puzzle pieces of a service and can be built out of product and service components. The overall experience a consumer has, is driven by the Service Interface. It is a mental concept in the consumer mind that has consequences on the perceived QoS. Designing this interface means to align all touch-points against the service concept. Furthermore as the consumer is a mandatory stakeholder of the service delivery process, he is an integral part of the service performance. QoS will be dependent of the quality of consumer's participation in the process (requirements elicitation ability) and of the quality of stakeholders' experience/expertise (interaction ability).

It's difficult to build a solution if the requirements are not known. The "elicitation" phase is the phase of the service delivery process during which the requirements are gathered from the client. Many techniques are available in the IT domain for gathering requirements. Each has value in certain circumstances, and in many cases, we need multiple techniques to gain a complete picture from a diverse set of clients and stakeholders like one-on-one interviews, group interviews, joint application development (JAD), questionnaires, prototyping, Use cases, brainstorming, etc.

The concepts borrowed from the computer science and IT domain presented here assumes that everything is a service (XaaS). Previous works dealing with service in manufacturing already mentioned this fact: in 1972, Lewitt claims that “everybody is in service” [5] and later Gronross attempted to quash the division between products (tangibles) and services (intangibles), saying that: “it does not make sense to determine whether customers buy products or services [6]. What they actually buy are the benefits that products and services provide them with. From this perspective, all companies can be said to offer services, even traditional manufacturing firms. Our works pointing in the same direction, the previous concepts are adapted for use in service (more or less tangible) delivery system (i.e. provision of a product or a service in a dual representation).

# 3 A Conceptual Model of Service and Service Delivery

## 3.1 Basic Modeling Concepts

Based on section 2, we consider that (Figure 1):

- A service as an action performed by a provider to the benefit of a consumer [7], interaction between both [8]. The provider uses means and has a capability enabling one or more actions. These actions lead to a result or an effect aiming to satisfy consumer requirements against its own actions performed using its own means and capability.

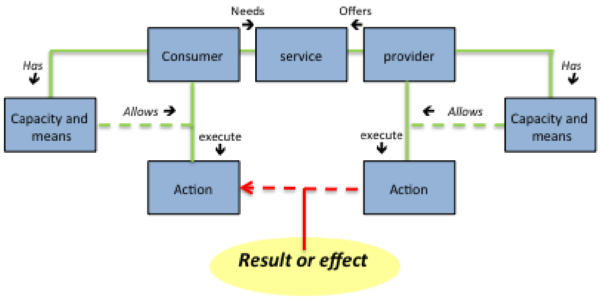


**Fig. 1.** Service interaction modeling

- A provider is identified with regard to his function and is noted F. A consumer is characterized by his need and is noted B (figure 2, left part). As a provider can be consumer of one or more services and conversely, we assume that it can become a hybrid object (figure 2, right part) [9].


**Fig. 2.** Basic principle and hybrid object generic representation

- Service delivery relations become then more complex i.e. each relation is defined for a given service. A hybrid object is then part of a service delivery series, linked to another upstream object as consumer and a downstream object as provider.

## 3.2 Service Delivery Process

A service delivery process is a process enabling the delivery of a service. It requires a coupling between a provider and consumer and sometimes necessitates means. The service is returned as long as the coupling exists. When it stops, the service ends and each stakeholder find back its freedom. When the coupling starts, requirements have to be elicited by the consumer.

The abovementioned description supposes that the service delivery process can only be led during the coupling. Obviously, the interaction between the service provider/consumer is the main part of the service delivery process. However, in more complex cases both actors can require to be prepared in an upstream phase (some sort of pre-process) and to get free in a downstream phase (some sort of post-process) (figure 3). The corresponding phases are the following ones:

**Initialization:** this phase does not require the coupling to be established but requires to know that the service must be returned. Information on the service needs is necessary to activate the phase. Information refers for the consumer to the identification by his requirements and for the provider to the identification of the requests he can satisfy.

**Customization and contextualization:** in case the service is not standard, a phase of customization based on information coming from consumers is to be envisaged. The contextualization focuses on the adaptation to the context (consumer, surrounding conditions, etc.) of the service to be returned and of the service delivery process.

**Closing and de-contextualization** phases exist when both actors require a process to close the activity. This process is similar to the one of the initialization phase but occurs after the service delivery. It can lead for the provider in a change regarding its ability to deliver the same service [10].

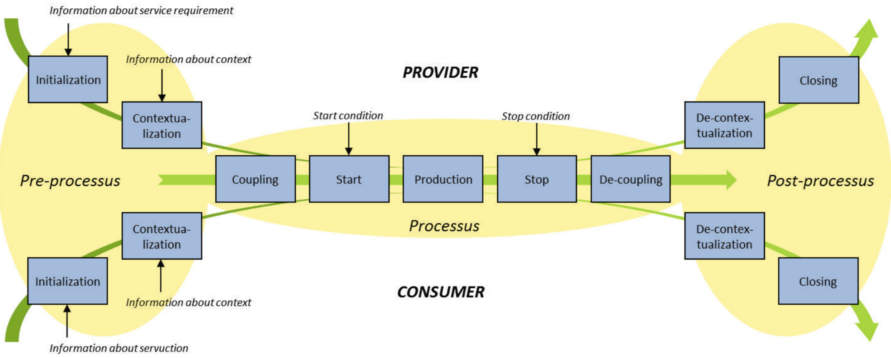


**Fig. 3.** Cartography of the whole service delivery process

### 3.3 Global Dynamics of Simple Services

The above-presented model focuses on an elementary service (sub-service) to be delivered. Three items of importance concerning the dynamic of the model and its operationalization need to be addressed: the matching procedure between a function and a need, the definition by the consumer of the function that can fulfill his need (service requirements) and at least, the identification of the provider that can propose the function.

Accordingly, the process is as follows: any provider has to declare publicly the precise nature of the functions he can fulfill in a "service directory". The statement is build upon a reference frame in order to facilitate the function description, its comparison and ranking regarding functions that can be competitive. The nature of the function is the static part of the statement while the capacity that can be used at a moment corresponds to the dynamic part of the statement (figure 4, left part).

When a consumer has a request, he has to elicit his requirements (needs) using the same frame to make a comparison feasible (figure 4, central part), i.e. at first in term of nature and then in term of load. Expression of QoS is sought during this step.

Once this is done, the matching between providers and consumer can be started. The function offered is compared to expressed needs. In case there is a matching i.e. a provider can provide the consumer with the required function to satisfy the requirement (nature, load and QoS) and a service is delivered. If no, the request cannot be satisfied and then can be updated, or confirmed and put on hold until a provider declare a relevant function. A final opportunity, discussed in the next part, concerning the composition/decomposition of sub-services exists.

The whole process is described on the right part of figure 4: (1) requirement elicitation by the consumer, (2) service discovery according to the frame (request), (3) choice among providers able to fulfill the need and service allocation, (4) service integration and delivery by provider to the consumer.

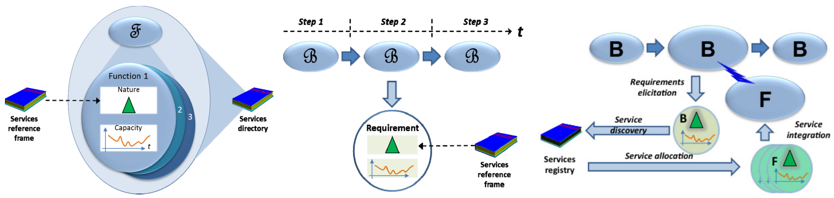


**Fig. 4.** Service statement, expression of needs and matching sequence

# 4 Service Composition

Obviously, service requirements are most of the time specific and no existing service in the market is able to satisfy them. A solution is to combine existing services together in order to fulfill the request. This trend known as service composition receives attention from both academia and industry. The selected and finally integrated services should optimize the overall QoS of the composed service while satisfying all the constraints specified by the consumers on individual QoS parameters. Algorithms exist to select services to combine based on global QoS optimization [11], [12]. The implementation of these algorithms can sometimes lead to unfeasible solutions because of the lack of interoperability among providers during integration or between consumers and providers.

The mechanism we propose is first to build the service registry (pre-processus, see figure 3) and then use it (service delivery process). The process is described hereafter:

**Step 1-service decomposition:** any existing service will be decomposed to collect the service components from user requirements and identifying the important details of each service to reduce the efforts in service integration and allocation.

**Step 2- Service registry:** any existing service, sub-service and component details will be registered using a registration procedure. When new services enter into service market, its whole description is registered.

**Step 3- Consumer requirement discovery:** customer requirements are described as explicitly as possible. Consumers are expected to fill out a form that is composed of a host of items on a user interface.

**Step 4- Service decomposition:** Once customer requirements are elicited, the requested service is to be decomposed to suitable extent, neither too detail nor too specific. The decomposition process is based on the functional requirements. The interrelationship and interoperability conditions of the sub-functions that guide the decomposition are described in a wise way so that decomposed sub services don't have too many interactions between them to reduce errors and interaction.

**Step 5- Service selection and integration:** Based on QoS requirements and service ranking coming from previous user experiences and former service performances, service integration gives as an output a sorted list of services that can be used to satisfy the need. To rank services based on multiple KPIs, we propose a ranking mechanism based on Analytic Hierarchy Process. There are three phases in the process: problem decomposition, judgment of priorities, and aggregation of these priorities. In the first phase, the ranking of a complex problem is modeled in a hierarchy structure that specifies the interrelation among three kinds of elements, including the overall goal, QoS attributes and their sub-attributes, and alternative services. The second phase consists of two parts: a pairwise comparison of QoS attributes is done to specify their relative priorities; and a pairwise comparison of services based on their QoS attributes to compute their local ranks. In the final phase, for each alternative service, the relative local ranks of all criteria are aggregated to generate the global ranking values for all the services. **Service integration strategy:** in many cases to have a standalone service is not enough. When there is no single service having the ability to satisfy a consumer requirement, service composition is needed to select several correlative services together for the purpose of fulfilling the need. Therefore, the problem is how to effectively and efficiently integrates the services provided by different providers to new services with higher value. After service allocation, one decomposed sub service could have more than one kind of candidates with the ranking of service allocation strategies. However, it doesn't means that integrating the highest ranked sub services together could generate an optimal result. Accordingly, the sets of the services selected by the service allocation mechanism is the preliminary one which provide a plenty sets of services to do a further selection.

## 5 Conclusion

The existing service market proposes services that are not enough to satisfy all consumer requirements as each one has his proper own logic. Customizing a new one with a specified provider is not a wise choice for service consumers as it costs a lot and is also time consuming. As a consequence, service decomposition and integration mechanisms based on Service Engineering are proposed to solve the above problem focusing on user requirements discovery, service decomposition and integration strategy. There are still some works need to be done in the future. Among them is the need to make more efforts to seek for the non-existing but largely demanded service components published by consumers, in the service market within which to guide the providers' future business work to produce such greatly demanded services. Finally, integrated service or newly generated services should be registered as a new service that can gradually enrich the quantity of services provided in market and will continuously benefic service component providers and future consumers.

## References

1. Varia, J.: Best practices in architecting cloud applications in the AWS cloud. In: Buyya, R., Broberg, J., Goscinski, A. (eds.) Cloud Computing: Principles and Paradigms, ch. 18, pp. 459–490. Wiley Press, New Jersey (2011)
2. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Journal of Future Generation Computer Systems 25(6), 599–616 (2009)
3. Zhang, L., Li, B., Chao, T., et al.: Requirements Driven Dynamic Business Process Composition for Web Services Solutions. Journal of Grid Computing 2(2), 121–140 (2004)
4. Yu, T., Lin, K.J.: Service selection algorithms for Web services with end-to-end QoS constraints. In: Proc. of the IEEE International Conference on e-Commerce Technology (CEC), pp. 129–136 (2004)
5. Levitt, T.: Production‐line approach to service. Harvard Business Review 50(5), 41 (1972)
6. Grönroos, C.: Service Management and Marketing: A Customer Relationship Management Approach. John Wiley & Sons (2000)
7. Grönroos, C.: Service Management and Marketing – Managing the Moments of Truth in Service Competition. Lexington Books, Toronto (1990)
8. Tocquer, G., Langlois, M.: Marketing des services, Le défi relationnel. Gaëtan Morin éditeur (1992)
9. Touzi, W.: Conceptualisation et modélisation de la production de service: application aux domaines de la santé et de l'enseignement. Thèse de doctorat, Université Bordeaux 1 (2001)
10. Touzi, W., Alix, T., Vallespir, B.: Contribution to the development of a conceptual model of service and service delivery. In: Emmanouilidis, C., Taisch, M., Kiritsis, D. (eds.) APMS 2012, Part II. IFIP AICT, vol. 398, pp. 369–376. Springer, Heidelberg (2013)
11. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: An Approach for QoS-aware Service Composition based on Genetic Algorithms. In: Proc. of the GECCO 2005 Conference (2005)
12. Ye, X., Mounla, R.: A Hybrid Approach to QoS-aware Service Composition. In: Proc. of the ICWS 2008 Conference (2008)