

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Progressive Service Value Network Design for Bi-lateral Service Applications

<sup>1</sup>Dian-Hui Chu, <sup>1</sup>Zhong-Jie Wang, <sup>1</sup>Xiao-Fei Xu and <sup>2</sup>Zhe-Ming Lu

<sup>1</sup>School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China

<sup>2</sup>School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China

---

**Abstract:** In service industries, a dominant morphology of service applications is a bi-lateral Resource Integration Service Pattern (called BIRIS), in which the broker is responsible for aggregating distributed service resources into a whole and providing it to customers. The innovation of such a service was based on the innovation of its Service Value Network (SVN) which determined the effectiveness and efficiency of value co-production and delivery to service participants to a large extent. This study presented a progressive service value network design scheme for bi-lateral service scenarios. It consisted of four steps. The first step was the initial SVN identification. The second step was the Virtualized Service Resource (VSR) identification and its lifecycle design. The third step was the importation of enablers and the final step was the selection of brokers. Detailed process for each step was elaborately described and analyzed.

**Key words:** Bi-lateral services, service value network, service enabler, service broker, virtualized service resource, bi-lateral resource integration service

---

### INTRODUCTION

A third-party broker based service pattern, called Bi-Lateral Resource Integration Service (BIRIS) (Wang and Xu, 2009), is a prevalent and dominant morphology in current service applications. In BIRIS, a large number of customers and providers are aggregated by a service broker, providers register their available services to the broker and customers submit their requests to the broker, then the broker acts as an intermediary between both sides by selecting proper services from one or multiple providers to fulfil the customer's demands. The advantage of BIRIS is that the cost of discovering services (Zhang *et al.*, 2009) and establishing relations between customers and providers can be reduced, the complexity of service collaborations is lowered and the difficulty of monitoring and assuring service quality and credit (Wong, 2009) is released.

The essence of a service is the "value" and the essence of a service system is the ability to assist all participants in co-producing values (Spohrer *et al.*, 2007; Lee and Lee, 2011). The service value is defined as the benefit or improvement that a participant gains from the service. There are tangible and intangible values (Allee, 2002). The former are defined as "all exchanges of goods, services or revenue", while the latter are mainly defined as "knowledge and benefit". In spite of the types of values, they are achieved by service behaviors and their interactions, the utilization of service resources, the exchange of information and their compositions

(Corchado *et al.*, 2012; Wang *et al.*, 2011). In order to help customers and providers better achieve their expected values, another type of participant, called service enabler, is imported into complex services (Basole and Rouse, 2008). A customer receives the expected values from the service by paying money or another economic cost, a provider receives economic benefits from the service by providing its resources and behaviors to help customers achieve their goals and an enabler helps customers or providers perform a better service.

According to the behavior that an enabler exhibits, there are three types of enablers (Wang *et al.*, 2010): (1) Customer-oriented Enabler (CE) that helps customers to receive the services more easily and conveniently. (2) Provider-oriented Front-stage Enabler (PFE) that helps service providers interact with customers more easily during the front stage of service encounters. (3) Provider-oriented Back-stage Enabler (PBE) that provides support for service providers in the back stage and does not interact directly with customers.

Along with the increasing number of participants (customers, providers and enablers) in a specific service, the interaction and value exchanges between them become increasingly complicated. Service Value Network (SVN) (Allee, 2000) is widely used as a tool to describe the value co-production relationships. An SVN defines a directed network topology in which the nodes are service participants, the directed edges indicate the value production and value recipients and values are attached to corresponding edges. The corresponding service

functional models like business processes are utilized to describe the functions that support the implementation of SVN (Saleem *et al.*, 2012).

In one service, there is at least one customer and one provider and zero or multiple enablers. Aiming at a specific BIRIS-oriented service, the most important enabler is the “broker”; besides, there might be a variety of other enablers. Adopting different strategies to import enablers and select a broker, thereby leading to different forms of SVN, has a significant impact on the effectiveness and efficiency of value production and delivery to the customers and providers (Wang *et al.*, 2010, Wang *et al.*, 2012).

This study presented a progressive service value network design process consisting of four steps. The first step was to identify the end participants (customers and providers) and clarify their value expectations, then establish the initial Service Value Network SVN<sup>(1)</sup>. The second step was to identify the Virtualized Service Resources (VSR) and analyze their lifecycle. The third step was to import a set of necessary enablers and form SVN<sup>(2)</sup>. The final step was to identify the broker and form SVN<sup>(3)</sup>.

**INITIAL SVN IDENTIFICATION**

This step is to identify the end customers and providers along with their value expectations.

The initial SVN is defined as  $SVN^{(1)} = \langle C \cup P, G, V \rangle$ , where C and P are the sets of customers and providers, respectively, G is the set of value exchange relationships and V is the set of exchanged values.  $\forall g \in G, g = \langle v, a_i, a_j \rangle$ , where  $v \in V$  is the exchanged value between  $a_i$  and  $a_j$ , with  $a_i$  the value producer and  $a_j$  the value receiver and  $a_i, a_j \in C \cup P$ .  $\forall a_k \in C \cup P, PV_k = \{v | \forall g \in G, g = \langle v, a_i, a_k \rangle\}$  is the set of values that  $a_k$  expects to obtain from the service.

A single expected value  $v_i \in PV_k$  of the customer C is exhibited exteriorly as a Virtualized Service Resource, denoted as  $VSR_i = \langle \{r_{ij}, A_{ij}\}, D_i \rangle$ , where there are multiple concrete resources  $r_{ij}$  attached with a set of attributes  $A_{ij}$  (e.g., quantity and priority) and the  $D_i$  are the dependencies between different  $r_{ij}$ . Here,  $r_{ij}$  may be specific information, behaviour, physical resources, etc. Only when  $VSR_i$  has been successfully produced and transferred to its receiver, has  $v_i$  been successfully co-produced and transferred.

In  $SVN^{(1)}$ , however, there are no enablers or brokers. It does not conform to the BIRIS pattern but it is merely the initial state of a service, i.e., end customers and end providers interact with each other directly without any help from others.

**VSR IDENTIFICATION AND ITS LIFECYCLE DESIGN**

In this step, the designer analyzes the VSRs identified in the previous step, examining each phase in their lifecycle to ascertain whether there are any deficiencies. The results will facilitate the improvement and optimization of the initial SVN<sup>(1)</sup> in the next two steps. Two types of lifecycle are used: (1) SR-LC: the lifecycle of a concrete service resource; (2) VSR-LC: the lifecycle of a VSR.

First, the SR-LC is considered. Suppose a resource ‘r’ is provided by ‘p’ and a customer ‘c’ requests it, then ‘r’s’ typical lifecycle is shown in Fig. 1.

The service provider first manufactures r himself or procures r from another resource manufacturer and then publicizes r’s information to potential customers. The customer submits a query and obtains r’s information and then raises a request to use it. The provider accepts the reservation and both sides reach a consensus on the SLA of the usage of r by negotiation. Further, the provider configures the reserved r based on the customer’s personalized demand. In the service process, both sides co-use the resource and co-govern its usage. Thereafter, the customer pays for the resource and the provider receipts the money. No matter whether r is in use or not, the provider should self-govern and maintain r to keep it in a sound condition.

The VSR-LC gets more complicated. When a customer requests the VSR, there are two possible approaches: (1) Following the process of the SR-LC, ‘c’ requests each ‘r<sub>i</sub>’ contained in the VSR separately, then aggregates them as a VSR. This indicates that (a) c submits a request and negotiates with multiple providers multiple times and (b) c is himself responsible for the governance and payment of multiple resources during the service process. This approach is complex and is actually

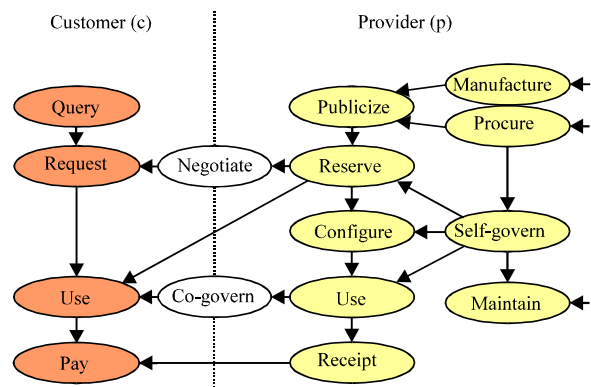


Fig. 1: Lifecycle of a concrete service resource (SR-LC)

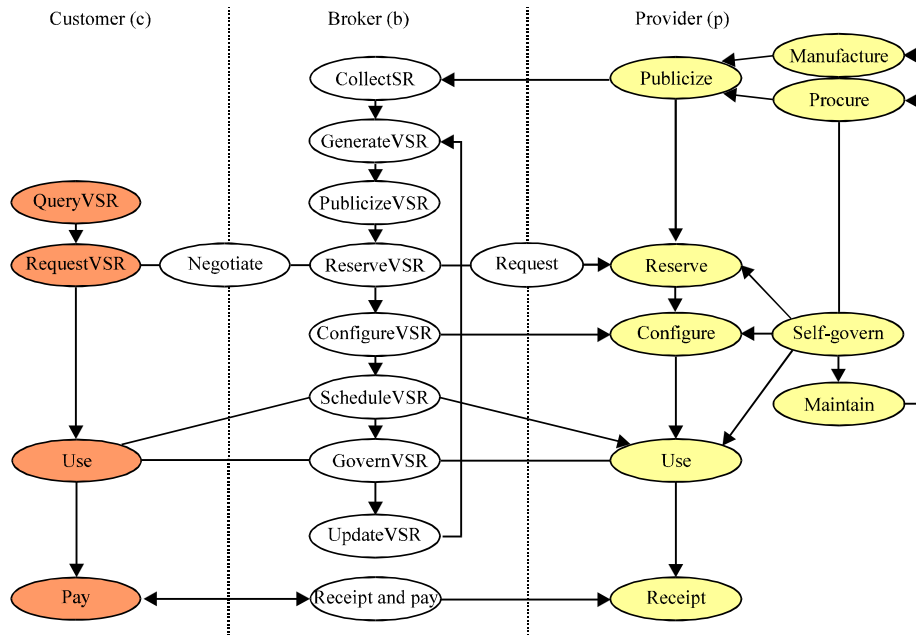


Fig. 2: Lifecycle of VSR (VSR-LC)

not the focus of BIRIS. (2) There is a broker ‘b’ that is responsible for collecting all the resources and aggregating them as an integrated VSR, which is then provided for c. This lifecycle, as shown in Fig. 2, is the actual approach adopted in BIRIS.

In the VSR-LC, for each service provider  $p_i$ , some of the phases in the SR-LC are followed, i.e., first, manufacture or procure the resource  $r_i$ , then publicize it. But now it is not the customer but the broker b, that collects the information of the distributed  $\{r_i\}$  (i.e., CollectSR) and aggregates some of them to generate a VSR (i.e., GenerateVSR), the information of which is then publicized to potential customers (i.e., PublicizeVSR). A customer may make a query on the VSR owned by the broker (i.e., QueryVSR), make a corresponding reservation (i.e., RequestVSR and ReserveVSR) and reach a consensus on the SLA by negotiation with the broker (i.e., negotiate). Then, the broker, playing the role of the customer in the SR-LC, sends requests to the providers whose resources have been included in the reserved VSR and makes a reservation for the individual resources  $\{r_i\}$ . Next, the configuration of the VSR and  $\{r_i\}$  is considered. The broker configures the VSR according to the customer’s personalized demand (i.e., ConfigureVSR). The configuration of the VSR is decomposed into each  $r_i$ ’s configuration, which is sent to the corresponding service providers who then configure their own resources themselves (i.e., configure). Subsequently, the broker schedules the VSR and starts the service process, in

which the customer and all the providers use the VSR for value co-production, while the broker governs its usage (i.e., GovernVSR). Based on the statistical data from the governance records, the broker may update the setup and configuration of the VSR for better usage in the future (i.e., UpdateVSR).

Finally, the three parties together make the settlement; specifically, the customer pays the broker, the broker receives the money and then pays each individual provider (i.e., Receipt&Pay) according to the previously agreed price in the SLA and finally, the providers receipt their own income.

Similarly, each provider self-governs and maintains  $\{r_i\}$ , keeping it in a sound condition during the whole lifecycle. A set of domain-independent QoS metrics is attached to each phase of the SR-LC and VSR-LC, including time, cost, availability, reliability, etc. As there are no any enablers in the SVN<sup>(1)</sup>, almost all the phases have obvious defects in terms of these QoS. This is the reason why the next two steps are necessary.

### IMPORTING ENABLERS INTO SVN<sup>(1)</sup>

The objective of this step is to import one or more enablers into the SVN<sup>(1)</sup> to form SVN<sup>(2)</sup> = {C, P, E, G, V}, where E is the set of new imported enablers.

Starting with the SVN<sup>(1)</sup>, each of its value exchange relations is analyzed. If there are potential deficiencies, it is modified by iteratively applying one of the three basic

operations (i.e., outsourcing, mashup and third-party payment) to import new enablers. This process is repeated until there are no further possible improvements.

The starting point of this phase is  $SVN^{(1)}$  and the end point is the  $SVN^{(2)}$  and between these phases there may be a set of intermediary SVN's (denoted as  $SVN^{(M1)}$ ,  $SVN^{(M2)}$ , ...) during the continual improvement. For the sake of brevity, here two neighboring SVN's ( $SVN^{(i)}$  and  $SVN^{(i+1)}$ ) are aimed to introduce the method for transforming  $SVN^{(i)}$  into  $SVN^{(i+1)}$ .

In  $SVN^{(i)}$ , each exchanged value may be exteriorly exhibited in the form of a VSR. Together with the iterative exploitation of the four operations, a VSR is gradually divided into multiple finer-grained VSRs or concrete resources. The decomposition is carried out with gradually imported new enablers that help the original participants provide or use the service resources better. The lifecycle of these VSRs/SRs are also possibly divided and allocated to new enablers, i.e., every new enabler is responsible for one or several phases of the VSR/SR's lifecycle. This step is a process whereby the VSRs are broken into parts in a top-down manner. Considering the objective of importing new enablers in this step, the deficiencies in the QoS in each phase of the VSR are eliminated or decreased, thereby greatly enhancing the value to both the customer and provider.

**Service Outsourcing (OTS):** In the proposed context, outsourcing (OTS) is defined as the operation whereby one or many phases of a VSR/SR's lifecycle are separated out from the responsibility of an original participant p in the  $SVN^{(i)}$  to a new enabler e rendering assistance to p. This operation is beneficial to both p and e as it enables p to reduce costs and increase quality in non core areas of the business and to utilize his expertise and competencies to the maximum in his core business, thereby enhancing his competitiveness (Kedia and Lahiri, 2007).

Suppose in  $SVN^{(i)}$  there is a value exchange relation  $g = \langle v, a_i, a_j \rangle$ , where the VSR that satisfies v is  $VSR = \langle \{r_k, w_k\}, D_k \rangle$ ,  $a_i$  is the provider of  $\{r_k\}$  and the producer of v and  $a_j$  is the resource user of  $\{r_k\}$  and the receiver of v. From the viewpoint of a resource  $r_k \in VSR$ , there are four types of OTS operations:

- **OTS<sup>(1)</sup>:** This operation takes place from  $a_i$ 's perspective. After OTS<sup>(1)</sup> is applied,  $a_i$  still owns  $r_k$ , but one or more management or usage-oriented phases of  $r_k$  (publicize, reserve, negotiate, configure, self-govern, co-govern, maintain, receipt) are

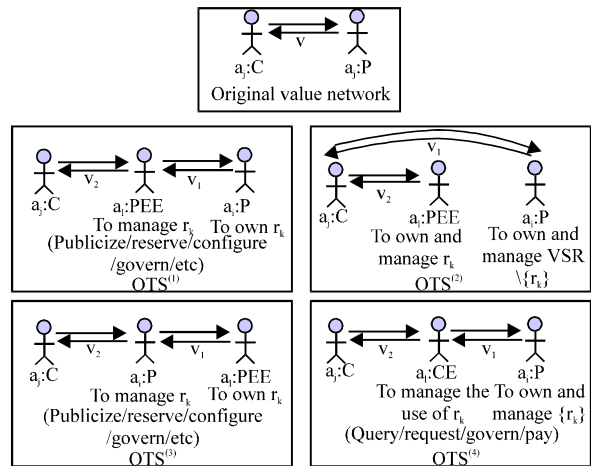


Fig. 3: Four types of Service Outsourcing (OTS) Operations

transferred from  $a_i$  to a new PFE  $a_i$  that exercises the management right of  $r_k$  on behalf of  $a_i$ . g is subsequently divided into two value exchange relations  $g_{i1} = \langle v_1, a_i, a_j \rangle$  and  $g_{i2} = \langle v_2, a_i, a_j \rangle$ .

- **OTS<sup>(2)</sup>:** This operation takes place from  $a_i$ 's perspective. After OTS<sup>(2)</sup> is applied,  $a_i$  no longer owns  $r_k$ ; instead  $r_k$  is owned by and provided to  $a_j$  by a new PFE  $a_i$ . This operation splits the original VSR (provided by  $a_i$ ) into two parts ( $\{r_k\}$  and  $VSR \setminus \{r_k\}$ ) provided by  $a_i$  and  $a_j$ , respectively. g is transformed into  $g_{i1} = \langle v_2, a_i, a_j \rangle$  and  $g_{ij} = \langle v_2, a_i, a_j \rangle$ .
- **OTS<sup>(3)</sup>:** This operation also takes place from  $a_i$ 's perspective. After OTS<sup>(3)</sup> is applied,  $a_i$  no longer owns  $r_k$ , instead  $r_k$  is owned by a new enabler  $a_i$  and  $a_i$  rents  $r_k$  from  $a_i$  and again provides the usage to  $a_j$  and the management and provision right belongs to  $a_i$ . Now  $a_i$  is regarded as a PBE and g is divided into two value exchange relations  $g_{i1} = \langle v_1, a_i, a_j \rangle$  and  $g_{ij} = \langle v_2, a_i, a_j \rangle$ .
- **OTS<sup>(4)</sup>:** This operation takes place from  $a_i$ 's perspective. After OTS<sup>(4)</sup> is applied, one or many usage-oriented phases (query, request, negotiate, co-govern, pay) of  $r_k$  are not dedicated to  $a_i$  himself, but outsourced to a new Customer-oriented Enabler (CE)  $a_i$  and g is divided into two value exchange relations  $g_{i1} = \langle v_1, a_i, a_j \rangle$ ,  $g_{ij} = \langle v_2, a_i, a_j \rangle$ .

The above definitions are based on a single resource  $r_k$  in a VSR. In fact, the OTS operations may be applied to multiple resources in a VSR individually or simultaneously. Aspects of these OTS operations are depicted in Fig. 3.

Figure 4 gives an example in which four OTS operations are applied jointly. The original VSR includes

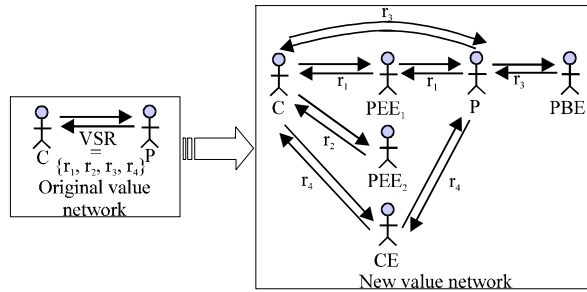


Fig. 4: Complex service outsourcing example

four resources  $\{r_1, r_2, r_3, r_4\}$  and these are outsourced to new enablers PBE, PFE<sub>1</sub>, PFE<sub>2</sub> and CE, respectively as follows: (1) Applying OTS<sup>(1)</sup> to  $r_1$ : P owns  $r_1$  but PFE<sub>1</sub> manages  $r_1$  and provides it to C. (2) Applying OTS<sup>(2)</sup> to  $r_2$ : PFE<sub>2</sub> owns, manages and provides  $r_2$  to C. (3) Applying OTS<sup>(3)</sup> to  $r_3$ : PBE owns  $r_3$  and provides the usage right to P and P manages  $r_3$  and provides it to C. (4) Applying OTS<sup>(4)</sup> to  $r_4$ : P owns and manages  $r_4$  but CE receives  $r_4$  from P and provides it to C.

**Service mashup (MSP):** Mashup (MSP) is defined as the operation in which new enablers are imported into the original SVN to aggregate resources from multiple providers as an integrated VSR to be provided to customers, or to solicit numerous customers for providers (Zou and Pavlovski, 2007). This operation aims at improving the cost of the query/reservation/usage of service resources to both sides.

There are two types of MSP operations, i.e., MSP<sup>(1)</sup> and MSP<sup>(2)</sup>. The MSP<sup>(1)</sup> operation takes place from the customer’s perspective. It aggregates resources of either the same or different types, i.e., (1) If a VSR that a customer expects to receive consists of multiple resources  $\{r_1, \dots, r_n\}$  that are provided by different providers, a new enabler  $a_i$  is imported to aggregate all or part of  $\{r_1, \dots, r_n\}$  into a VSR and provide it to the customer; (2) If there are numerous service providers that could provide the same resource  $r_k$  expected by a customer, a new enabler  $a_i$  is imported to collect and aggregate the information on the multiple available  $r_k$ , provide it to the customer and assist him in selecting an appropriate one. Suppose there are  $n$  value exchange relations  $g_1, g_2, \dots, g_n$  between a customer  $a_i$  and  $n$  providers  $a_{i1}, a_{i2}, \dots, a_{in}$ . Then, after the enabler is imported, they are transformed into  $\langle v_1, a_i, a_i \rangle$  and a set of  $\langle v_k, a_{ik}, a_i \rangle$  as shown in Fig. 5a.  $a_i$  plays the roles of both CE and PFE. On the other hand, the MSP<sup>(2)</sup> operation takes place from the provider’s perspective. When a resource  $r_k$  is publicized to attract widespread requests from customers, an enabler  $a_i$  intended to appeal to customers,

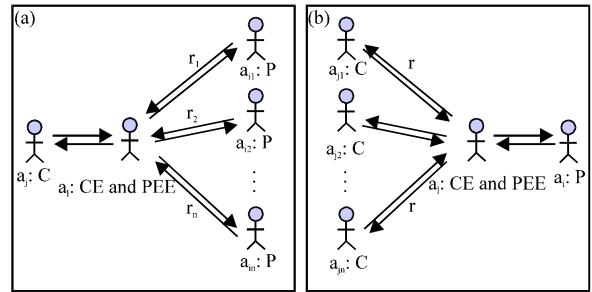


Fig. 5(a-b): Service Mashup (MSP) operation

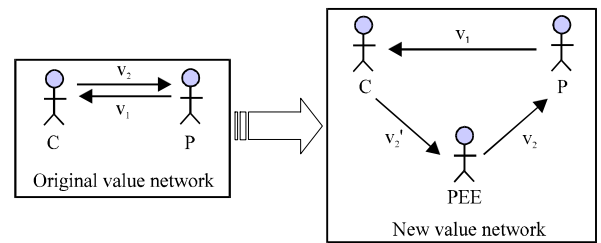


Fig. 6: Third-Party Payment (TPP) operation

is imported to assist the provider increase the utilization of  $r_k$  and the scale of the market. This is shown in Fig. 5b.

**Third-Party Payment (TPP):** Suppose there exist  $g_1 = \langle v_1, a_i, a_i \rangle$  and  $g_2 = \langle v_2, a_i, a_i \rangle$  in which  $v_2$  is a kind of economic value, referred to as the money that  $a_i$  pays to  $a_i$ . Third-Party Payment (TPP) (Li and Zhang, 2008) is defined as the operation whereby a new enabler  $a_i$  is imported and forms new value exchange relations  $g_1 = \langle v_1, a_i, a_i \rangle$ ,  $g_2 = \langle v_2, a_i, a_i \rangle$  and  $g_3 = \langle v_2', a_j, a_i \rangle$ , as shown in Fig. 6.

In reality, such an operation is usually employed in scenarios where a third-party participant attempts to gain large-scale “attention” (also called propagation effect, advertisement effect, or market effect). In Fig. 6, P owns and provides a concrete resource to C and the service relations between P and C are considered to be a propagation route for the attention to PFE and numerous customers (C) use the resources from P freely and as an auxiliary effect their “attention” to PFE accumulates, forming the advertisement effect that PFE wishes to achieve. This is the reason why PFE is motivated to pay the resource usage fee, also called the “advertisement fee” to P on behalf of C.

**Summary:** Here the proposed views are reasserted that the above three basic operations aim to improve the QoS of the SR/VSR lifecycle:

Table 1: Improvement in four types of operations on SR-LC phases

Participant	Phase	OTS <sup>(1)</sup>	OTS <sup>(2)</sup>	OTS <sup>(3)</sup>	OTS <sup>(4)</sup>	MSP <sup>(1)</sup>	MSP <sup>(2)</sup>	TPP
Customer	Query							
	Request							
	Negotiate							
	Use							
	Co-govern							
Provider	Pay							
	Procure							
	/Manufacture							
	Publicize							
	Negotiate							
	Reserve							
	Configure							
	Use							
	Co-govern							
	Receipt							
	Self-govern							
	Maintain							

OTS<sup>(1)</sup> is the first type of outsourcing operation defined in the sub-section of Service Outsourcing (OTS), OTS<sup>(2)</sup> is the second type of outsourcing operation defined in the sub-section of Service Outsourcing (OTS), OTS<sup>(3)</sup> is the third type of outsourcing operation defined in the sub-section of Service Outsourcing (OTS), OTS<sup>(4)</sup> is the fourth type of outsourcing operation in the sub-section of Service Outsourcing (OTS), MSP<sup>(1)</sup> is the first type of mashup operation defined in the sub-section of Service Mashup (MSP), MSP<sup>(2)</sup> is the second type of outsourcing operation defined in the sub-section of Service Mashup (MSP), TPP is the third-party payment operation defined in the sub-section of Third-Party Payment (TPP). The up arrow ↑ means that the performance of the corresponding phase on the left of the row could be improved by the operation on the top of the column

- **OTS:** By high-degree socialization (i.e., breaking up the entire VSR into parts), service providers can enhance their specialization of resource affordance and efficiency/quality of resource management, while customers can use the resources with little cost
- **MSP:** Through high-degree aggregation (i.e., merging parts into a whole), customers can easily obtain a coarse-grained VSR at once, while providers can also obtain the attention of more customers
- **TPP:** Through the importation of third-party stakeholders, the customers' cost to use the resources is reduced and the market scale is increased

In Table 1 the symbol “↑” is used to indicate which phases of the SR/VSR-LC are improved by each of the above operations. In most cases, the selection and application of these operations on a specific SVN<sup>(1)</sup> depends on the experience and intuitive insights of the service designers. Specific decision-making methods are required to make the improvement process more objective and quantitative. This is one of the future work.

**SELECTION OF BROKERS FROM ENABLERS**

The objective of this step is to select one or several enablers from CUPUE as brokers (B) of BIRIS, thereby producing a new value network SVN<sup>(3)</sup> = {C, P, E, B, G, V}.

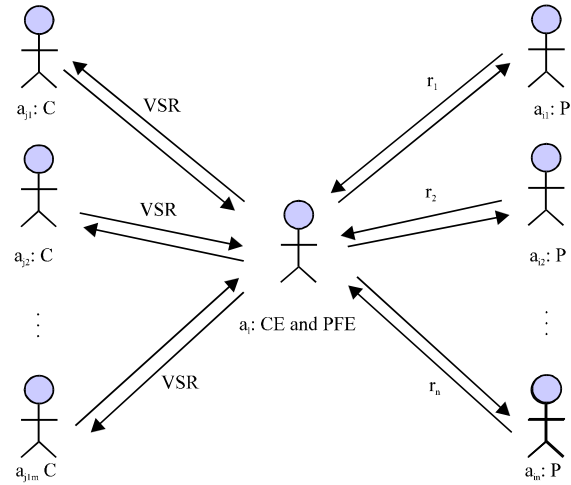


Fig. 7: Position that a broker holds in SVN<sup>(3)</sup>

The position that a broker holds in SVN<sup>(3)</sup> is shown in Fig. 7. It must be a bi-lateral (connected to both the customer and provider sides) oriented enabler that looks like being imported by the combination of MSP<sup>(1)</sup> and MSP<sup>(2)</sup>.

The sole decision-making problem during the transformation from SVN<sup>(2)</sup> to SVN<sup>(3)</sup> is: which participant(s) should be selected as the broker(s) so that the total performance is optimum?

First, which types of service participants may play the broker's role are considered. From Fig. 7, it is known that a broker must face up to and connect customers and providers directly or indirectly. Accordingly, of the five types of service participants (C, P, PFE, PBE and CE), only CE and PFE are qualified as brokers (e.g., CE brokers C and PFE/P, PFE brokers C/CE and P), P is qualified as a broker only when it has the corresponding PBE (i.e., it brokers C/CE/PFE and PBE), while C and PBE are unqualified because they are only connected to one side of the service (e.g., C only connects to CE and PBE only connects to P). The analysis results are shown in Table 2.

Secondly, if there are multiple CE/PFE/P in the SVN<sup>(2)</sup>, which would be the best broker? Aiming at one participant p in SVN<sup>(2)</sup>, the following three empirical principles are used to speculate whether it is suitable to act as a broker:

- Location of p in SVN<sup>(2)</sup>. The broker should be located in a relatively central position, serving as a connecting hub between the participants on the customer- and provider-oriented sides. The number of participants on both sides should be relatively equal and balanced

Table 2: Qualifications require to be a broker

Type of participant	Customer side it connects to	Provider side it connects to	Qualification
Customer-oriented side	C	—	CE (if there are CE) PFE (if there are no CE) P (if there are no CE and PFE)
	CE	C	PFE (if there are PFE) P (if there are no PFE)
Provider-oriented side	PFE	C (if there are no CE) CE (if there are CE)	P
	P	C (if there are no PFE and CE) CE (if there are no PFE) PFE (if there are PFE)	PBE (if there are PBE) — (if there are no PBE)
	PBE	P	—

C: Customer, CE: Customer-oriented enabler, PFE: Provider-oriented Front-stage enabler, PBE: Provider-oriented Back-stage enabler, P: Provider

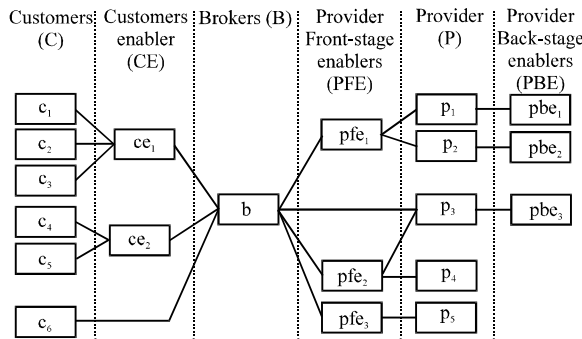


Fig. 8: Representative BIRIS-oriented SVN<sup>(3)</sup>

- p’s market impact. The broker should have the largest impact on the other participants in the service; in other words, he is powerful enough to aggregate a large number of customers to use the service and persuade a large number of providers to publicize and share their resources on the platform
- The criticality of resources or their lifecycle phases controlled by p. The broker should own or handle key phases of the SR/VSR, impacting most of the key QoS of the SR/VSR-LC at large

Once a common CE/PFE/P is promoted to be a broker, besides the original responsibilities it has in SVN<sup>(2)</sup>, a set of extra fundamental functions concerning the VSR’s lifecycle are added, including distributed resource collection, VSR construction, VSR selection/scheduling/usage/feedback/governance/payment, etc.

There may be multiple brokers in a SVN<sup>(3)</sup> and these take the form of a hierarchically nested structure. In this case, there must be a dominant broker that divides the entire SVN into two halves, while in each half, there exist subordinate brokers and so on. After this step, a representative BIRIS-oriented SVN<sup>(3)</sup> has the form as illustrated in Fig. 8.

## CONCLUSION

Concerning the prevailing bi-lateral resource integration service pattern (BIRIS) in present real-world service applications, the design of its value network is of great significance to the quality of value production and delivery between customers and providers. How to import new enablers and select a proper enabler to play the role of broker, is the core issue addressed in this study. A four steps progressive design process is presented, based on the optimization of Virtualized Service Resource (VSR) lifecycle and the iterative application of three pattern-oriented operations. This work will facilitate the effective innovation of fast-growing service-oriented enterprise information systems. Future work will concentrate on applying our framework to developing a real system and test the effectiveness of our framework in practical applications.

## ACKNOWLEDGMENTS

This work is partially supported by the National Science Foundation of China (grant no. 61272187, 6103300, the National Science and Technology Support Program of China (grant no. 2012BAH10F00), Key Technologies R & D Program of Shandong Province (Nos. 2010GZX20126, 2010GGX10116 and 2011GGA10001), Science and technology projects in Jiangsu Province (Nos. BC2010107, CSRC1010 and ZXG201006). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

Allee, V., 2000. Reconfiguring the value network. *J. Bus. Strategy*, 21: 36-39.  
 Allee, V., 2002. A value network approach for modeling and measuring intangibles. Presented at Transparent Enterprise, Madrid, Spain. <http://www.vernaallee.com/images/VAA-A-ValueNetworkApproach.pdf>



- Basole, R.C. and W.B. Rouse, 2008. Complexity of service value networks: Conceptualization and empirical investigation. *IBM Syst. J.*, 47: 53-70.
- Corchado, J.M., D.I. Tapia and J. Bajo, 2012. A Multi-agent architecture for distributed services and applications. *Int. J. Innovative Comput. Inform. Control*, 8: 2453-2476.
- Kedia, B.L. and S. Lahiri, 2007. International outsourcing of services: A partnership model. *J. Int. Manag.*, 13: 22-37.
- Lee, W.I. and C.L. Lee, 2011. An innovative information and relationship between service quality, customer value, customer satisfaction and purchase intention. *Int. J. Innovative Comput. Inform. Control*, 7: 3571-3581.
- Li, Y. and X. Zhang, 2008. The revenue model of online third-party payment based on game theory. *Proceedings of the International Conference on Management of e-Commerce and e-Government*, October 17-October 19, 2008, Jiangxi, China, pp: 349-353.
- Saleem, M.Q., J. Jaafar and M.F. Hassan, 2012. Secure business process modeling of SOA applications using UML-SOA-SEC. *Int. J. Innovative Comput. Inform. Control*, 8: 2729-2746.
- Spohrer, J., P.P. Maglio, J. Bailey and D. Gruhl, 2007. Steps towards a science of service systems. *IEEE Comput.*, 40: 71-77.
- Wang, J.B., Z.X. Cheng, L. Jing, K. Ota and M. Kansan, 2011. A Smart-gate based composition method to provide services by solving conflict using dynamic user priority and compromise policy. *Int. J. Innovative Comput. Inform. Control*, 7: 2987-3005.
- Wang, Z., Q. Wu and X. Xu, 2012. Value-added analysis of bi-lateral e-business services. *Proceedings of the International Conference on e-Business Engineering*, September 9-11, 2012, Hangzhou, China, pp: 185-192.
- Wang, Z.J. and X.F. Xu, 2009. Bilateral resource integration service mode for value innovation. *Comput. Integr. Manuf. Syst.*, 15: 2216-2225.
- Wang, Z.J., D.H. Chu and X.F. Xu, 2010. Value network based service choreography design and evolution. *Proceedings of the IEEE 7th International Conference on e-Business Engineering*, November 10-12, 2010, Shanghai, China, pp: 495-500.
- Wong, D., 2009. From network management to service management-A challenge to telecom service providers. *Int. J. Innovative Comput. Inform. Control*, 5: 669-675.
- Zhang, Y., H. Huang, D. Yang and H. Zhang, 2009. A Hierarchical and Chord-based semantic service discovery system in the universal network. *Int. J. Innovative Comput. Inform. Control*, 5: 3745-3753.
- Zou, J. and C.J. Pavlovski, 2007. Towards accountable enterprise mashup services. *Proceedings of the 2007 International Conference on e-Business Engineering*, October 24-October 26, 2007, Hong Kong, China, pp: 205-212.