International Journal of u- and e- Service, Science and Technology
Vol. 7, No. 6 (2014), pp. 253-264
http://dx.doi.org/10.14257/ijunesst.2014.7.6.22

Value-driven Service Matching and Bundling via a Service Value Broker

Yucong Duan, Nanjangud C Narendra, Honghao Gao and Xiaobing Sun

Hainan University, Hukou, China; duanyucong@hotmail.com
Cognizant Technology Solutions, India; ncnaren@gmail.com
Computing Center, Shanghai University, Shanghai, China; gaohonghao@shu.edu.cn
Yangzhou University, Yangzhou, China;1825740912@163.com

Abstract

Most research on service matching and composition (or bundling) has been technical in nature, viz., at the IT service level. Very little has been done on value-based service matching and bundling. By "value-based", we mean the value added by a service over and above the basic functional requirement that the service is supposed to satisfy. Examples of value added could be lowered cost, higher performance, improved security, etc. To that end, in this paper, we present our idea of a Service Value Broker (SVB),that performs value-driven service matching and bundling. Via an approach specifying business and technical interfaces for business services and IT services respectively, we show how value can be calculated and integrated at the business layer. Further, we also show how our approach makes it possible to determine the appropriate IT services to be selected so as to provide the required value add at the business layer. We demonstrate our ideas via a realistic case study in the E-Commerce domain.

Keywords: *Value Modeling, Service Matching, Service Bundling*

1. Introduction

The idea of broker can be traced back to the beginning of human economic activity. Typically, a broker is an individual or firm that charges a fee or commission for executing services among the service providers and customers initiated from either side. It acts as an agent for a customer or provider and charges the customer a commission for its services. Its duties typically include: determining service values, advertising properties for service, showing properties to prospective service customers, and advising customers with regard to offers and related other services.

Similar challenges exist in E-commerce; a natural extension is to create an E-commerce platform to address coherence and harmony between business objectives and technology enablement. It has been shown that E-brokers provide support to enhance interaction among services and stakeholders. In E-Commerce, the communications among different parties are completed through Web service calls. In the previous research of E-commerce, Bichler *et al.* [2] used e-brokers to enhance the interoperability of E-commerce at the application level. Loreto *et al.* [12] used brokers to integrate telephone business and IT world in the manner of an intermediate layer. Rosenberg and Dustdar [15] used e-brokers to bridge the difference of heterogeneous business rules. Brokers have been used to fill the gap of QoS expectations. Yu and Lin [20] utilized service brokers to meet SLAs of services and construct trust network for bridging reputation information [11].

ISSN: 2005-4246 IJUNESST

Brokers can be used for enhancing resource usage. Srikumar *et al.* [17] used a broker to enable grid resource searching and distribution where a broker functions mostly as an autonomous agent [13]. D'Mello [7] used a broker to select qualified services in terms of QoS. Budgen *et al.* [3] introduced an information broker to integrate health knowledge and data with enhanced privacy protection.

Existing broker researches have used brokers to discover, match, negotiate, select and compose services from either a technological or a business perspective. But there are very few which cover both perspectives at the same time.

Enlightened by the ideology of value driven engineering [5], we have proposed the concept of Service Value Broker (SVB) [8, 9][1] to fulfill the missing link which we have depicted as various mismatching situations. SVB as modeling and implementation element helps to integrate the analysis of economic and technical aspects for a business implementation on top of Web services. SVB related services are not limited to technological level alone such as most SLA-based approaches [20] but also move up to the business level [2]. This integration also helps to fill the gap of global [4] scheduling and local planning in an explicit and direct manner. Although contract negotiation [21] provides the flexibility of service transactions, it does not provide assistance on guiding the global schedule which SVB supports since the context of the service composition is usually not explicitly expressed inside a predefined service contract.

Hence we feel that service matching and composition, which has so far been limited to the technical level (a notable exceptions being *e*[3]*value*[2]), needs to consider value-based matching and bundling at the business level also. To that end, in this paper, we present our work on Service Value Broker that helps match and bundle business services, and also helps compose the constituent IT services thereof, in a holistic manner. We show that such an approach helps bridge the well-known business-technical divide in SOA, and helps discover and match services that can maximize value generation for customers. We also demonstrate our ideas on a realistic case study in the E-commerce domain, and also present a proof of concept prototype.

Our paper is organized as follows. The next section provides background material that will be used throughout the rest of the paper. Section 3 introduces our Service Value Broker idea, and describes how it works. Section 4 describes our two-level method for calculating value at business and technical layers, and integrating value parameters to provide a single unified value metric for the customer's evaluation. Our implementation and experiments on a realistic case study in the E-commerce domain are presented in Section 5, while comparison against related work is presented in Section 6. Finally, the paper concludes in Section 7 with suggestions for future work.

2. Background

We define a *business service* as any business functionality provided by a service provider that solves a business need. The business service is itself decomposable into a set of IT services; each IT service is a software component that accepts a set of inputs, provides a set of outputs, and is accessible only via its interface.

We say that each service (business or IT) helps to provide a value to the service customer, using which the customer can measure the extent to which the service in question satisfies the customer need. Hence from the interface perspective, we therefore identify two layers of

---

[1] This paper was supported in part by NSFC grant 61363007 and by HNU grant KYQD1242 and HDSF201310.
[2] http://www.e3value.com/

service interfaces: *technical interface* (*TIF*), at which we have the IT services whose interfaces are represented in computer-readable form (*e.g.*, REST, SOAP, WSDL), and *business interface* (*BIF*), where IT services are aggregated into business services.

Value is in turn measured as a weighted sum of several orthogonal parameters, each of which is normalized so that an increase in their numerical value signifies improvement in value added to customer. Hence a value parameter such as cost can be normalized by taking its reciprocal, so that an increase in its value implies an improvement in value add to customer. Hence we assume the value parameters $\{P_j\}$ for services $\{S_i\}$. This results in a service value matrix (V_{ij}), where V_{ij} stands for the value of the parameter P_j for the service S_i. Such a matrix lends itself to two ways in which value calculation can be composed.

In row-based composition, the values of the parameters for a single service can be combined into an overall value. Depending on the relative importance of each value parameter, we assign each as weight w_j, so that the overall value is calculated as $C_i = \sum w_j V_{ij}$ for the service S_i, where $\sum w_j = 1$.

In column-based composition, the overall value for a parameter of the service bundle can also be similarly calculated. In order to decompose this down to the TIF layer, the row-based composition becomes important. The business service in question is to be further decomposed into its constituent IT services. Let the business service S_i be decomposed into a set of IT services T_k, which are put together in a particular workflow sequence to meet the input and output requirements of S_i. Then for any value parameter P_j for S_i, its value V_{ij} would be derived from the (incremental) value provided by the individual services T_k. Hence, depending on the overall value to be provided by S_i, the appropriate IT services need to be chosen, along with the appropriate execution sequence, in order to ensure that the execution does deliver the value which S_i needs to provide.

Figure 1 depicts our conceptual model of business and IT (technical) interfaces of services. A business service would therefore be realizable by a composition of one or more IT services. The business interface of the service would participate, as part of an E-contract, to provide value to customers of that service. At this level, several aspects of the business service are crucial as depicted in Figure 1, viz., administrative, legislative, value provided (or demanded) and the overall Service Level Agreement (SLA) under which the service needs to deliver its value. (The focus in this paper would be on the value aspect, leaving other aspects for future work.)

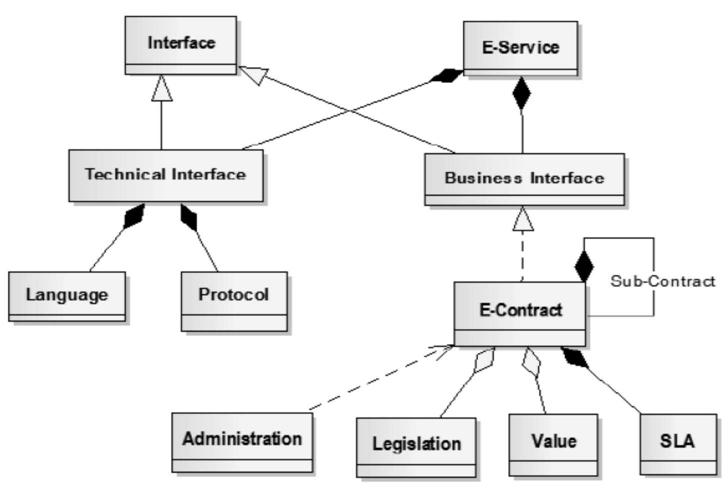


Figure 1. Metamodel of Business and Technical Interfaces of Web Services

3. Service Value Broker

In order to provide value-based matching and bundling of services to meet customer needs, we present in this section our Service Value Broker (SVB). Contrary to the usual notion of a broker that merely matches services against stated customer requirements, our SVB attempts to provide value-based matching and bundling with a view towards maximizing customer value. The overall meta-model of our SVB is depicted in Figure 2.

Thus the brokering function can be not only involved in merely fulfilling requirements as specified by the customer, but can also go above and beyond this to optimize service bundling so as to maximize overall value to be delivered to the customer. This can go along several directions:

1. *Requirements Engineering*: From the (traditional) requirements engineering perspective, brokers help fulfill the expected requirements via service bundling
2. *Value Driven*: From the value driven perspective, any processing will be justified as necessary only if it enhances user satisfaction. Brokers implement this via appropriate service matching and bundling to maximize value addition.
3. *Meta View*: Overall, brokering can one or more of the following components (see Figure 2):
    a. *Function*: a broker implements a specified function including certain processing, etc .
    b. *Quality*: a broker improves the required quality including the availability, throughput, timeliness, etc.
    c. *Resource*: a broker supplies the expected resources including data, information, storage, etc.
    d. *Knowledge*: a broker fills the gap on how to enable a mismatching situation without explicit indications on functionality, quality or resource, e.g., location mismatching, language mismatching, format mismatching, security, privacy, etc.
    e. *Control*: a broker can supply complex tradeoff or composition of basic forms of brokerings including function, quality, resource and knowledge to maximize customer satisfaction.

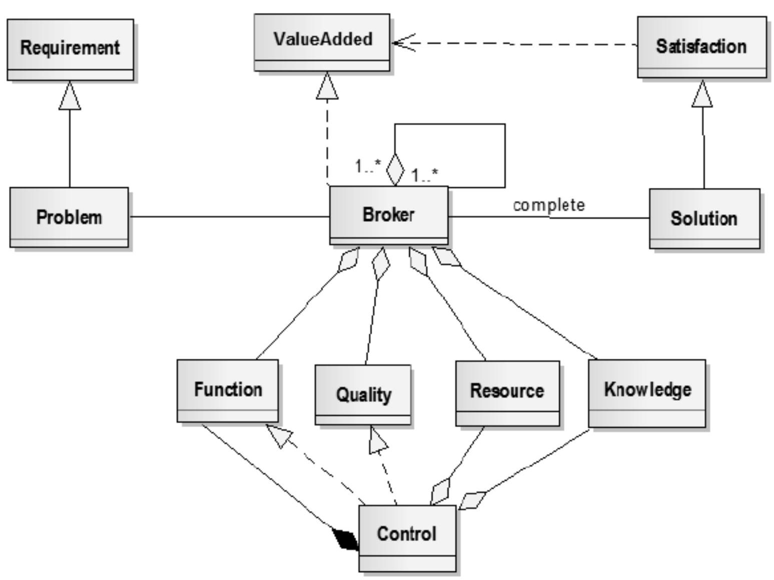


Figure 2. Metamodel of Service Value Broker

4. Service Value Broker

In this section we present the core contribution of our paper, viz., value-driven service matching and bundling. We first present our service matching approach with emphasis on the BIF layer, after which we show how this approach can then be extended to service bundling at BIF and TIF layers.

4.1. Service Matching

The first step in our approach is service matching. We assume that the customer has a set of business needs, expressed via a service contract. For each business need, the SVB searches for the appropriate service or collection of services that would fulfill the need. Among those services that do fulfill the business need, the SVB would then select those that would help maximize customer value.

To that end, we define a customer need via the following model that comprises the following:

- Inputs needed $\{d_{in}\}$
- Outputs to be generated $\{d_{out}\}$
- Description of the need in any format readable by the SVB
- A set of value parameters $\{V_j\}$, with each value parameter containing the following:
    - *ID* which represents the value name
    - Type I which represents the unit in which the value parameter is measured
    - Order *Ord* which could be one of the following: increasing or decreasing
    - Value *Val* which is the minimum (or maximum, in case *Ord* is decreasing) value expected

For example, a business need could be "Provide a mobile phone", with inputs being the mobile phone's technical specifications; and outputs being the mobile phone type, model number, and other details. A value parameter could be cost, which is expressed in US$, whose *Ord* is decreasing, and whose *Val* limit could be an upper limit of US$1,500. While there could be many business services that can provide a mobile phone for the customer within his/her price limit, the business service with best value would be that which provides the cheapest mobile phone satisfying the technical specifications.

Hence any business service can be modelled as per the following capability model:

- Inputs to the service $\{d_{in}\}$
- Outputs of the service $\{d_{out}\}$
- Service description in a format readable by the SVB
    - *ID* which represents the value name
    - Type *T* which represents the unit in which the value parameter is measured
    - Order *Ord* which could be one of the following: increasing or decreasing
    - Value *Val* which is the value promised by the service

4.2. Value-based Service Bundling

More often, however, a collection (*i.e.*, bundle) of services would need to be assembled in order to meet the customer need. Alternately, the customer may also specify a set of needs, which would have to be met by a service bundle. Bundling services based on customer needs alone can be implemented via the approach detailed in [14]; however, our emphasis here is more on maximizing the value of the service bundle presented to the customer.

Our method for calculating overall service value is a 2-step method. First, we determine the overall value of a particular value parameter across all services in the service bundle, via column-based value composition as introduced in Section 2. Second, we then determine the weighted average of these combined value parameters to determine an overall single value. This single value can be used to rank the service bundle against others in terms of their suitability to maximize customer value.

For the first step, therefore, the overall value of the j^{th} parameter would be $V_j' = \sum_{i=1}^{j} V_{ij}$, for the *m* services in the bundle. The second step would calculate the overall value as $\sum_{j=1}^{n} w_j V'_j$, where the w_j are the weights associated with each value.

Table 1.

Service	Price	Privacy	Timeliness	Reliability
Distribution	100	40	50	20
Currency exchange	20	20	30	10
Language	20	10	40	32
Time	10	5	20	40

For example, in our experiment of the E-commerce system (see Table 1 for illustration), we have four business services, viz., distribution, currency exchange, language and time. Their value parameters are price, privacy, timeliness and reliability. We define Reliability as the probability of returning responses after web services are successfully processed. We have,

$$\text{Reliability} = \frac{\text{Number of Response Messages}}{\text{Number of Request Messages}}$$

For the first step, we get

$$V_j' = \begin{bmatrix} 100 & 40 & 50 & 20 \\ 20 & 20 & 30 & 10 \\ 20 & 10 & 40 & 32 \\ 10 & 5 & 20 & 40 \end{bmatrix}$$

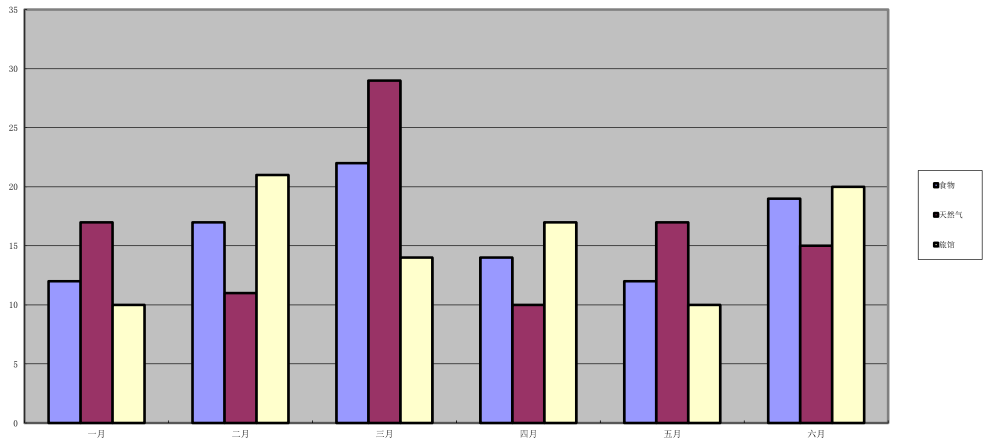
Let us assume the following weightages for the value parameters:

w_j = [0.4 0.4 0.1 0.1]

We calculate the overall value as $V = \sum_{j=1}^{n} w_j V'_j = 111.5$.

4.3. Value-based Service Generation

The main task, therefore, is to generate a set of service bundles that meet the customer need (*i.e.*, a set of feasible service bundles), and determine that bundle that helps maximize customer value. To that end, we present two algorithms.



Algorithm 1 Select Feasible Service Bundle(S)

1: $S_{SEL} = Ø$
2: Repeat
3: **if** (I (Si) $\subseteq \{d_{in_i}\}$) **or** (O $(Si) \subseteq \{d_{out_i}\}$) **then**
4: $S_{SEL} = S_{SEL} \cup Si$
5: $\{d_{ini}\} = \{d_{ini}\} - I(Si)$
6: $\{d_{outi}\} = \{d_{outi}\} - O(Si)$
7: end if
8: Until (All S_i visited) OR (($\{d_{ini}\} = Ø$) AND ($\{d_{outi}\} = Ø$))

Algorithm 2 Generate All Service Bundle Combinations

1: i = 0;
2: **for all** S_{SUB} **do**
3: $Inputs = \cup I(Si)$, where $Si \in S_{SUB}$
4: $Outputs = \cup O(Si)$, where $Si \in S_{SUB}$
5: S'= Select Feasible Service Bundle(S - S_{SUB});
6: i = i + 1
7: $S^i_{SEL} = S_{SEL} - S_{SUB} + S'$
8: end for

Algorithm 1 generates a feasible service bundle, while Algorithm 2 tweaks the feasible service bundle to determine all other service bundles that also meet the customer need. The value parameters for each service bundle can then be calculated as per Section 4.2, and the service bundle with the best value can be presented to the customer. Alternately, all feasible service bundles can be presented to the customer; he/she can then tweak the weightages among the value parameters, determine the overall value, and select the service bundle of his/her choice.

Algorithm 1 works as follows. Each service registered in the SVB is evaluated to determine if it meets any subset of the inputs and outputs of a customer need. If it does, it is added to the feasible list. The algorithm iterates until either all customer needs are met, or the list of services registered in the SVB is exhausted without all customer needs having been met.

In order to determine all possible service bundles, Algorithm 2 does three things. First, it picks any combination of services in the feasible service bundle generated via Algorithm 1, and removes them from the feasible service bundle. Second, it extracts the customer needs satisfied by this combination; then it determines a set of services (not in the feasible service bundle) that meet the extracted customer needs. These services are then integrated into the feasible service bundle to create a new service bundle. Since Algorithm 2 is iterated for all possible combinations of services from the feasible service bundle, it will generate all possible service bundles, for each of which the overall value can be calculated as per Section 4.2.

4.3. Discussion – Derived IT Service Composition

At the TIF layer, any business service is to be represented by a composition of IT services. The inputs and outputs of the business service can be the requirements to determine this composition. Several algorithms have been developed in the literature for IT service composition, viz., [1]. However, these approaches should be enhanced to ensure that the IT service composition does satisfy the value added to be provided by the business service in question. For example, if the business service would be shipping a product to a customer, this would be decomposed into IT services such as Select Shipper, Payment Gateway, and Inventory Management. If the value parameter in question is cost, then the overall cost of product shipment would be the sum of the costs involved in using these three IT services. Hence if the aim were to minimize cost in order to maximize customer value add, the appropriate combination of IT services can be chosen. Determining this optimal combination, therefore, would be a procedure similar to Algorithm 2 in order to generate all possible combinations so as to pick the optimal one. For now, we assume that each business service comes with the appropriate composition of IT services that are able to provide the necessary value add. For future work, we will be investigating how to compose IT services in order to generate a business service that meets prespecified value requirements.

5. Implementation and Experiments

We demonstrate an E-commerce system built using value driven service matching and bundling. This system helps deliver a product for a Chinese customer from an American provider. This involves services for language translation, logistics, validation, ordering, inventory management and currency exchange. Our SVB, therefore, should locate the appropriate business services that also maximize customer value. Our value parameters are, therefore, quality of service (QoS), price, privacy, timeliness and reliability. Table 2 lists the various value parameters of the services, normalized to a 0 - 100 range, and adjusted so that the *Ord* for each parameter is increasing. That is, a service with higher price parameter can be assumed to be relatively cheaper.

For each service, there are alternate services that do the same job but provide different value, such as different price, less or more reliability and less or more privacy, etc. We have shown alternate services in different composition with different value parameters.

From Section 4.3, using the Algorithm 1, each service registered in the SVB is evaluated to determine if it meets any subset of the inputs and outputs of a customer need. If it does, it will be added to the feasible list. For instance, we set the input to the language translation required service to be English, and output as Chinese. However, the input to $translation_b$ is French and the output of $translation_b$ is English. Clearly, $translation_b$ is not in the feasible list.

Table 3. Service Bundles

$bundle_1$	$bundle_2$
$translation_a$	$translation_c$
$currency_c$	$currency_a$
$shipping_b$	$shipping_b$
$ordering_a$	$ordering_c$
$validation_b$	$validation_a$
$inventory_b$	$inventory_b$
$security_b$	$security_b$

For simplification, we have assumed there were two service bundles in Table 3 that were finally chosen.

For $bundle_1$ and $bundle_2$, we get the matrices B1 and B2 from Tables 2, and the composition information of Table 3. That is,

$$B1 = \begin{bmatrix} 70 & 53 & 53 & 53 \\ 44 & 14 & 23 & 45 \\ 60 & 40 & 69 & 44 \\ 42 & 65 & 55 & 23 \\ 30 & 50 & 62 & 54 \\ 53 & 70 & 62 & 83 \\ 65 & 50 & 64 & 82 \end{bmatrix} \qquad B2 = \begin{bmatrix} 80 & 67 & 68 & 59 \\ 60 & 40 & 50 & 70 \\ 60 & 40 & 69 & 44 \\ 62 & 65 & 85 & 83 \\ 20 & 40 & 32 & 44 \\ 53 & 70 & 62 & 83 \\ 65 & 50 & 64 & 82 \end{bmatrix}$$

Let us assume the weightages for the value parameters are as follows.

w_j = [0.2 0.3 0.2 0.3]

We have calculated the final value of B1, B2, that is B2 = 1687 > B1 = 1474. In other words, bundle B2 provides greater value add.

6. Related Work

There are many factors, including business processes, which influence service matching. Gordijn *et al.* [10] have proposed a Value-Driven Service Matching method which aids general Business-IT Alignment through matching customer's requirements against available products/services from providers. However, Comerio *et al.* [6] demonstrate that, in general, service contract networks are influenced by the control flow and data flow of service compositions. Weigand *et al.* [18] have proposed a design method for value-based service modeling and design. Based on these findings, our work proposes an integrated method for service matching and combines customer needs and value adds to customers.

Traditional IT service design follows the process of software design where structural analysis and structural design focus on static and behavioral aspects respectively. Wittern and Zirpins [19] propose to model services' commonalities via feature modeling. However as is well known, it is Web services that integrate static objects in data flow with behavioral functionality in control flow. We will be leveraging this approach for our future work, where we will be extending our approach for deriving IT service compositions from business services.

Future software systems will be more adaptive and distributed in nature, and their value related requirements are not expected to remain static. To that end, Telang and Singh [16] model service transactions via contractual relationships expressed as commitments among business partners. We consider that work significant, since software systems of the future will be increasingly differentiated in terms of the overll value they provide, rather than on single factors such as performance, throughput, price, etc. To this end, we will be considering integrating the work in [16] as part of our future work.

7. Conclusions

In this paper, we addressed the crucial research issue of value-based service matching and bundling. Our work is based on the basic premise that service providers need to provide services that not only meet customer needs, but also maximize value added to customers. To that end, we proposed the idea of a Service Value Broker that can match customer needs with business services so as to maximize value. We proposed a definition of value which can be delineated via several value parameters, and integrated via a weighted approach. We showed that such an approach also provides the customer sufficient flexibility to evaluate multiple service bundle options by adjusting their weights. We demonstrated our approach via a realistic case study in the E-Commerce domain.

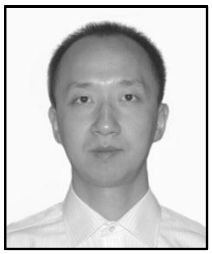
Apart from the future work proposed in Section 6, we propose to extend our idea by incorporating derived IT service composition so as to make our approach more dynamic. In addition, we will also be evaluating our approach on larger case studies.

References

[1] U. Bellur and R. Kulkarni, "Improved matchmaking algorithm for semantic web services based on bipartite graph matching", In ICWS, (**2007**), pp. 86-93.
[2] M. Bichler, A. Segev, and C. Beam, "An electronic broker for business-to-business electronic commerce on the internet", Int. J. Cooperative Inf. Syst., vol. 7, no. 4, (**1998**), pp. 315-330.
[3] D. Budgen, M. Rigby, P. Brereton and M. Turner, "A data integration broker for healthcare systems", IEEE Computer, vol. 40, no. 4, (**2007**), pp. 34-41.
[4] V. Cardellini, E. Casalicchio, V. Grassi, and R. Mirandola, "A framework for optimal service selection in broker-based architectures with multiple qos classes", In SCW, (**2006**), pp. 105-112.
[5] P. Collopy and P. Hollingsworth, "Value-driven design", Journal of Aircraft, vol. 48, no. 3, (**2011**), pp. 749-759.
[6] M. Comerio, H. L. Truong, F. De Paoli and S. Dustdar, "Evaluating contract compatibility for service composition in the Seco2 framework", In ICSOC/ServiceWave, (**2009**), pp. 221-236.
[7] D. A. D'Mello, V. S. Ananthanarayana and S. Thilagam, "A qos broker based architecture for dynamic web service selection", In Proceedings of AMS2008, pp. 101-106.
[8] Y. Duan, A. Kattepur, and W. Du, "Service value broker patterns: Integrating business modeling and economic analysis with knowledge management", In IEEE ICWS, (**2013**) June, pp. 615-616.
[9] Y. Duan, N. Narendra, W. Du, Y. Wang, and N. Zhou, "Exploring cloud service brokering from an interface perspective", In IEEE ICWS (**2014**).
[10] J. Gordijn, S. de Kinderen, and R. Wieringa, "Value-driven service matching", In RE, (**2008**), pp. 67-70.
[11] K.-J. Lin, H. Lu, T. Yu and C.-e. Tai, "A reputation and trust management broker framework for web applications", In Proceedings of the IEEE EEE, (**2005**), pp. 262-269.
[12] S. Loreto, T. Mecklin, M. Opsenica, and H.-M. Rissanen, "Service broker architecture: location business case and mashups", Comm. Mag., vol. 47, no. 4, (**2009**) April, pp. 97-103.
[13] Z. Qian, S. Lu, and L. Xie, "Mobile-agent-based web service composition", In 4th Intl. conf. on Grid and Cooperative Computing, pp. 35-46.
[14] I. S. Razo-Zapata, J. Gordijn, P. De Leenheer, and H. Akkermans, "Dynamic cluster-based service bundling: A value-oriented framework", In CEC, (**2011**), pp. 96-103.
[15] F. Rosenberg and S. Dustdar, "Design and implementation of a service-oriented business rules broker", In CECW, (**2005**), pp. 55-63.
[16] P. R. Telang and M. P. Singh, "Specifying and verifying cross-organizational business models: An agent-oriented approach", IEEE T. Services Computing, vol. 5, no. 3, (**2012**), pp. 305-318.
[17] S. Venugopal, R. Buyya, and L. Winton, "A grid service broker for scheduling distributed data-oriented applications on global grids", In MGC, (**2004**), pp. 75-80.

[18] H. Weigand, P. Johannesson, B. Andersson, and M. Bergholtz, “Value-based service modeling and design: Toward a unified view of services”, In CAiSE, (**2009**), pp. 410-424.
[19] E. Wittern and C. Zirpins, ‘On the use of feature models for service design: The case of value representation”, In ServiceWave Workshops, (**2010**), pp. 110-118.
[20] T. Yu and K.-J. Lin, “A broker-based framework for qos-aware web service composition”, In EEE, (**2005**), pp. 22-29.
[21] X. Zheng, P. Martin, and K. Brohman, “Cloud service negotiation: A research roadmap”, In IEEE SCC, (**2013**), pp. 627-634.

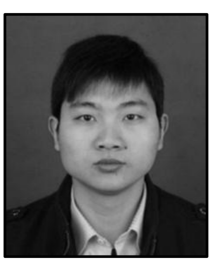
Authors



Yucong Duan, He received a PhD in Software Engineering from Institute of Software, Chinese Academy of Sciences, China in 2006. He is currently a full professor and vice director of Computer Science Department in Hainan University, P.R.China. His research interests include software engineering, service computing, cloud computing, and big data. He is a member of IEEE, ACM and CCF (China Computer Federation).

Nanjangud C Narendra, He received a PhD from Rensselaer Polytechnic Institute in 1991. He is currently a Chief Architect at Cognizant Technology Solutions, Bangalore, India. His research interests, include software engineering, service oriented computing, cloud computing, big data analytics, and Internet of Things. He is a Senior Member of IEEE and ACM.

Honghao Gao, He received the Ph.D. degree in computer application technology from the School of Computer Engineering and Science of Shanghai University, Shanghai, China, in 2012. His research interests include Web service and model checking.



Xiaobing Sun, He received his Ph.D. from Southeast University in 2012. He is currently is an assistant professor in School of Information Engineering at Yangzhou University. His research interests include software analysis, maintenance, and evolution. He is a CCF and ACM member.

