Using Pattern Perspective to Model Key Business Process Performance

Ruliang Xiao

Dep. of Information & Management, Hunan University of Finance & Economics, Changsha 410205, China Email: xiaoruliang@163.com

Abstract—Web service composition is a more and more promising solution for building distributed applications on the e-business processes. It has been recognized as a flexible way for resource sharing and application integration. As the number of functional similar Web services increases, after Web services be selected that can best meet the requirements of the consumers, how to construct a kind of key business process performance (KBPP) model has been an ongoing research direction in business process performance management for Web services. In this paper, the key business process performance of business with specific features (non-functional characteristics or QoS) based on pattern perspective. This new approach helps developers to direct related engineering implementation.

Index Terms—Quality of Service, BPM, Pattern Perspective, KBPP

I. INTRODUCTION

Recently Web service has been recognized as the next generation framework for building agile distributed applications over the Internet. A composite service is usually modeled as a business process involving multiple component services. Web services are loosely-coupled, platform-independent, self-describing software components that can be published, located and invoked via the web infrastructure using a stack of standards such as SOAP, WSDL and UDDI. It offers a promising solution for building distributed applications on the ebusiness processes. People are able to rapidly design, implement, deploy and deliver various application functionalities using a standardized web services model [1]. Service oriented computing advocates discovering, selecting and binding to services dynamically according to users' requirements in functional aspects as well as in non-functional aspects, especially the quality of service (QoS).

To be more responsive and cost-effective in today's economy, many enterprises provide different web services. Examples of this include Google SOAP Search API for information queries [2], Amazon web services for various e-commerce solutions [3]. But individual web service can not able to meet complex requirements of business processes. They can be combined to create new value added composite services for those requirements. With more and more web services providing similar functionalities and possessing different non-functional property values, how to construct a kind of KBPP model

has been an ongoing research direction in business process performance management for Web services.

Non-functional properties are characterized as quality of service (QoS). It is a broad concept that consists of a number of nonfunctional properties such as response time, price, availability, reliability, and reputation. They can apply both to individual web service and to composite web services. However, there is currently no standardized description framework to include all aspects of service's nonfunctional characteristics. Especially, QoS concerns all of our business process. It leads to the issue of selecting of the most appropriate web services composed into the optimization scheme for the required business process of a service request among a list of candidate web services. Although previous studies have proposed many solutions to these concerns [4,5,6], service requesters still lack an efficient way to compare web service QoS based on global optimization.

Considering that work flow technology continues to be subjected to ongoing development in its traditional application areas of business process modeling and business process coordination, and now in emergent areas of component frameworks and inter-workflow, businessto-business interaction, this paper proposes a QoS based web services selection model based on the Key Business Process Performance (KBPP). This paradigm should typically deals with following issues. (1) Creation of a QoS selection model: the purpose of this model is to describe the key performance of business with specific features; (2) Regulation design pattern of key business process performances; (3) Pattern based on optimal weight value of varied functional and non-functional constraints.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 suggests a framework of web services optimal selection based on key business process, builds a key business process model of web services, succeeding presents a kind of business process with five basic formal business process from the viewpoint of web service pattern perspective in the section 4. Section 5 integrates QoS information into the KBPP, indicates principle of application of business processes, and specifies web service combining patterns with QoS-based service selection scheme. Finally, section 6 makes a conclusion.

II. RELATED WORK

As the number of functional similar web service increases in the Internet, QoS based web service selection and combination has become a hot research. OoS information describes non-functional aspects of web services and QoS capability is becoming a decisive factor to distinguishing services for selecting most appropriate services. In the literature [4], Yu and Lin design the service selection algorithms to meet end-to-end QoS constraints. It models the problem as the Multiple Choice Knapsack Problem and provides efficient solutions. Li Y et al. [5] suggest a service selection approach considering the trustworthiness of QoS data, which classifies and computes the QoS attributes according to the source of QoS data. Huang, Lan and Yang in the Literature [6], based on QoS measurement metrics, propose multiple criteria decision making and integer programming approaches to select the optimal service.

Zeng et al. [7] discuss a global planning approach for selecting composite web services, and propose a simple QoS model containing five QoS criteria (price, duration, availability, reliability, and reputation). They did not derive aggregate QoS performance from workflow patterns, but split a service composition into execution paths represented by directed acyclic graphs. They also applied integer programming to solve the objective function of each execution path and merge the solutions of all execution paths to help service requesters select an optimal service. Inevitably, the computational cost of this approach increases exponentially with the number of exclusive choices in the composition, and multiple execution paths may produce conflicting service selections. However, these approaches are often too complex for run-time decisions. Ref.[13] propose a model-driven approach, which automatically transforms a design model of service composition into an analysis model, which then feeds a probabilistic model checker for quality prediction. They developed a prototype tool called ATOP, and demonstrate its use on a simple case study. Through which QoS properties maybe specified and formally analyzed for service compositions. In particular, QoS reasoning is based on probabilistic modeling, which is crucial for performance and reliability prediction.

Process models based on web service selection and combination with QoS performance can be configured to support specific business processes. Now, several languages have been proposed to support processorientation in the context of web services (cf. BPEL4WS, BPML, WSCI, etc.)([9]). The support of IBM, Microsoft, HP and SAP for a language like BPEL4WS (Business Process Execution Language for Web Services, [14]) reinforces the fact that process-awareness has become one of the cornerstones of information systems development. Existing languages and tools focus on control-flow and combine this focus with mature support for data in the form of XML and database technology. As a result, control-flow and data-flow are well addressed by existing languages and systems. But all these works do not focus on the specification of web service combining patterns with QoS-based service selection scheme,

III. A FRAMEWORK OF WEB SERVICES OPTIMAL SELECTION BASED ON KEY BUSINESS PROCESS

Previous researches mainly place emphasis on the individual QoS of nonfunctional aspects round service oneself not the whole. Now, our work focuses on the whole optimization over aggregating QoS performance from the key business process. General speaking, different business processes are often autonomy and heterogeneity, and come from various virtual organizations. In the meantime of composing web services, all collaborating partners need to have governance of the overall process status at all time [8]. When a service request selects web services that belong different organizations, all QoS element of to participating business processes are considered. Hence, optimal web service selection decision must be derived from the aggregating KBPP model explicitly. In this context, the key point is to build optimal schedule work on service selection through the business process workflow.

As the Figure1 shows, it presents the framework of a web services optimal selection based on key business process. The framework demonstrates the functional structure of the OoS information aggregating management and the web services selection according to the requirement of user application. The left interpreter translates the requirement of user application into the key business processes model, and then, the middle selector queries the broker about some aggregating QoS information through the repository where the web service providers register with UDDI, the OoS aggregating broker return optimal scheme to support for the services selector component while the former selector depends on the aggregating QoS attributes.



Figure 1. A Framework of Web Services Optimal Selection Based on Key Business Process

Our modeling techniques tend to focus on a particular aspect of the problem domain, and does not place stresses

on the methods of interpreter, but also devotes our mind to the model of the key business process that the results of translating the user application requirement. Figure 2 presents a kind of business process around application requirement as following.



Figure 2. A Kind of Business Process Workflow of Web Services

There exist a lot of composite service description approaches such as Web Services Business Process Execution Language (WSBPEL)[9] and Color Petri Netbased model[10]. Here are developed from traditional business process workflow modeling works. The following task aims at building the core structure of business workflow process. By partitioning this business process workflow of web services, we can acquire five basic kinds of service components, and further build a new model of KBPP, see Fig.3.



Figure 3. Building Key Business Process Structure Model of Web Services

The sequence structure (see Fig.3(a)) serves as the fundamental building block for workflow processes. It is used to construct a series of consecutive activities which execute in turn one after the other. The second sub-graph Fig.3 (b) illustrates the implementation of the Switch branches, where control flow goes through one of the branches from the first operator XOR and out of the second XOR operator. The third presents a web services parallel structure, where control flow of execution parallels by splitting from the first operator AND, and

joining after going out of the second AND operator, in the meantime, a new synchronizing system implicitly exists these two process (see Fig.3 (c)). But similar control flows are shown in the Fig.3 (d) where the AND operator implements a synchronizing process, the XOR operator practices in an asynchronous way. This system is called Pick structure. The final is called as web services loop structure, while a condition XOR is satisfied, the control will flows along.



Figure 4. Five Key Business Process Pattern of Web Services Represented by Colored Petri Net

As already mentioned above, there are a series of distinct modeling structure that are captured. From the viewpoint of patterns, based on the previous model, we will further refine the work flow patterns out of the KBPP model and present in the section after assimilating literature [10]. By partitioning this business process workflow of web services, we can acquire five basic kinds of service patterns of KBPP, see Fig.4. There are some assumptions that apply to next models on the Colored Petri Net. For each of them, we adopt a notation in which input places are labeled i1...in, output places are labeled o1...on, internal places are labeled p1...pn and transitions are labeled A...Z. In the case where either places or transitions serve a more significant role in the context of the pattern, they are given more meaningful names. In general, transitions are intended to represent tasks or activities in processes and places are the preceding and subsequent states which describe when the activity can be enabled and what the consequences of its completion are. Structured key business process pattern are presented, and have their respective duties as following description, see Fig.4.

(1)Web Service Sequence Pattern (see Fig.4.(a)):

Description:

An activity in a workflow process is enabled after the completion of another activity in the same process. The Sequence pattern acts as the fundamental building block for workflow processes. It is used to construct a series of consecutive activities which execute in turn one after the other. Two activities form part of a Sequence if there is a control flow edge from one of them to the next which has no guards or conditions associated with it.

Implementation:

The sequence pattern is used to model consecutive steps in a web service workflow process and is directly supported by each of the workflow management systems available. The typical implementation involves linking two activities with an unconditional control flow arrow.

(2)Web Service AND-Split Pattern (see Fig.4.(b)):

Description:

A point in the web service workflow process, where based on a decision or workflow control data, a number of branches are all chosen. In general, the divergence of a branch into two or more parallel branches each of which execute concurrently. This point in the web service work flow process, where a single thread of control splits into multiple threads of control, which can be executed in parallel, thus allowing activities to be executed simultaneously or in any order.

Implementation:

The AND-Split pattern is implemented by two ways: either (1) the edge representing control-flow can split into two (or more) distinct branches or (2) the activity after which the AND-Split occurs has multiple outgoing edges which do not have any conditions associated with them.

Full support for this pattern is demonstrated by any offering that provides a construct (either implicit or explicit) that allows the thread of control at a given point in a process model to be split into two or more concurrent branches.

(3)Web Service AND-Join Pattern (see Fig.4.(c)): Description:

The convergence of two or more branches comes into a single subsequent branch, resulting that the thread of control is passed to the subsequent branch when all input branches have been enabled, where multiple parallel sub-processes / activities converge into one single thread of control. It is an assumption of this pattern that each incoming branch of a synchronizer is executed only once. Implementation:

This pattern similarly to Pattern 2, one can identify two basic approaches: explicit AND-joins and implicit joins in an activity with more than one incoming transition. Only when each of these arcs has received the thread of control can the activity be enabled.

(4)Web Service XOR-Split Pattern (see Fig.4.(d)):

Description:

A point in the web service workflow process, where based on a decision or workflow control data, one of several branches is chosen. The divergence of a branch splits into two or more branches, when the incoming branch is enabled; the thread of control is immediately passed to precisely one of the outgoing branches based on the outcome of a logical expression associated with the branch.

Implementation:

It provides three distinct means of implementing this pattern: (1) based on the evaluation of bool expression one of two possible branches chosen; (2) one of multiple possible branches is chosen based on the value of a specific data element, and (3) based on the outcome of a preceding activity, a specific branch is chosen. The work flow designer has to emulate the exclusiveness of choice by specifying exclusive transition conditions.

(5)Web Service XOR-Join Pattern (see Fig.4.(e)): Description:

The convergence of two or more alternative branches becomes into a single subsequent branch such that each enablement of an incoming branch results in the thread of control being passed to the subsequent branch.

Implementation:

Given that we are assuming that parallel execution of alternative threads does not occur, this is a straightforward situation and it requires the merge construct to always be preceded by a corresponding exclusive choice construct. If more than one path is taken, synchronization of the active threads needs to take place. If only one path is taken, the alternative branches should re-converge without synchronization. It is an assumption of this pattern that a branch that has already been activated, cannot be activated again while the merge is still waiting for other branches to complete. The implementation of the synchronizing merge typically is not straightforward.

Once QoS estimates for above key business process model are determined from these web service work flow patterns, we can integrate some QoS information into KBPP model. We will describe an aggregating mathematical modeling technique in the next section.

V. INTEGRATING QOS INFORMATION INTO THE KBPP

We have already established that aggregating key business process model represents basic component configuration underlying the business process workflow of web services to be executed during the information system is operating. Web service description language (WSDL) [6,11] has provided a standard model to specify service functionality by separating the abstract representations of service input and output messages from the concrete descriptions of each end point's bindings. However, there is currently no standardized description to all aspects of service's non-functional include characteristics. This section synthesizes related work [6, 10, 11, 12] to present a QoS model of web service which aggregates the Key Business Process Performance (KBPP), and can be used to discuss the proposed service selection scheme hereafter.



Figure 5. KBPP: An Aggregating QoS Information Model

According to literature [4, 6], response time T(s) (for service s) is a common and universal measure of performance for web service. Reliability of web service R(s) (for service s) refers to the service provider's ability to successfully deliver requested service functionality. Availability of web service A(s) (for service s) is the degree to which a service is operational and accessible when it is required for use. Price P(s) (for service s) is the cost of service for a request. It is always associated with the value of the service's functionality. According to the previous works on aggregating model, we can further obtain an aggregating QoS model based the KBPP as follows Fig.5.

VI. CONCLUSION

References

- F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana, Unraveling the web services web, an introduction to SOAP, WSDL and UDDI, IEEE Internet Computing 6(2) 86-93(2002).
- [2] S. Koshman, Visualization-based information retrieval on the web, Library and Information Science Research 28 (2) 192-207 (2006).
- [3] L.S. Chen, F.H. Hsu, M.C. Chen, Y.C. Hsu, Developing recommender systems with the consideration of product profitability for sellers, Information Sciences 178 (4) 1032-1048 (2008).
- [4] T. Yu, K.J. Lin, Service selection algorithms for web services with end-to-end QoS constraints, in: Proceedings of IEEE International Conference on Ecommerce Technology (CEC), pp. 129-136 (2004).
- [5] Li Y, Zhou MH, Li RC, Cao DG, Mei H. Service selection approach considering the trust worthiness of QoS data. Journal of Software, 19(10):2620-2627 (2008).
- [6] Angus F.M. Huang a, Ci-Wei Lan b, Stephen J.H. Yang, An optimal QoS-based Web service selection scheme, Information Sciences 179 3309-3322 (2009)
- [7] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, H. Chang, QoS-aware middleware for web services composition, IEEE Transaction on Software Engineering 30 (5) 311-327(2004).
- [8] Soffer, P., & Ghattas, J. Business process flexibility in virtual organizations. Workshop on Business Process Modeling, Design and Support (BPMDS'06), Proceedings of CAiSE'06 Workshops, 188-197 (2006).
- [9] Web Services Business Process Execution Language (WSBPEL), Version 2.0. wsbpel-specification-draft-01. OASIS(2006)
- [10] N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar. Workflow Control-Flow Patterns: A

Web service composition is a promising solution for building distributed applications on the e-business processes. In this work, we construct a QoS aggregated web services selection model based on the KBPP, give five basic formal business processes, further present relative web service pattern, such as web service sequence pattern, web service And-Split pattern, web service And-Join pattern, web service XOR-Split pattern, web service XOR-Join pattern. Simultaneously, based on the KBPP model, aggregating some QoS effects, we describe the key performance of business with specific features, and various parameters using dynamic decision of key business process performances. All these works will help developers to improve the service selection process in a dynamic and uncertain environment of web services with the validity and efficiency.

Revised View. BPM Center Report BPM-06-22, BPMcenter.org, (2006).

- [11] C.S. Park, S. Park, Efficient execution of composite web services exchanging intensional data, Information Sciences 178 (2) 317-339(2008).
- [12] J. O'sullivan, D. Edmond, A.T. Hofstede, What's in a service? Towards accurate description of non-functional service properties, Kluwer Academic Publishers Distributed and Parallel Databases 12 117-133 (2002).
- [13] Stefano Gallotti, Carlo Ghezzi, Raaela Mirandola et. al., Quality Prediction of Service Compositions through Probabilistic Model Checking. QoSA2008, LNCS 5281,pp.119-134 (2008)
- [14] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services Version 1.1. Technical report, 2003. Accessed at http://xml.coverpages.org/ BPELv11-May052003Final.pdf on 27 November 2004. (2003)



Ru Liang Xiao Hunan Province, China. Birthdate: July, 1966. is Computer Software Engineering Ph.D., graduated from State Key Lab of Software Engineering of Wuhan University. And research interests on Relaiablity of Software Engineering and Virtual compute theory.

He is a associate professor of Software engineering.