

# Process Reservation for Service-Oriented Applications

HongHui Chen, JianWei Ma, Xianpeng Huangfu,  
Deke Guo

C4ISR Technology Key Laboratory of Science and  
Technology for National Defense  
National University of Defense Technology  
Changsha, P.R. China  
e-mail: cch0808@gmail.com

Stephan Reiff-Marganiec

Dept. of Computer Science  
University of Leicester  
Leicester, UK  
e-mail: srm13@le.ac.uk

**Abstract**— With an increasing use of services sustaining the resources which people need has become more important. In this paper, we propose an effective reservation method, called “BPSR” (Business Process Service Reservation), which aims for the process reservation which to the best of our knowledge has not been studied before. In particular we address for major jobs: service differentiation; service reservation; process reservation and QoS Control. We also describe a flexible policy-based reservation method which aims to increase the success rate of reservation and utilization of service resources. Experimental analysis shows that the BPSR reservation system achieves better results than other reservation methods.

**Keywords:** Service Reservation; Process Reservation; Service Oriented Application

## I. INTRODUCTION

Nowadays, the paradigm in distributed applications is shifting towards Service Oriented Architecture. In the past few years, various web services have been deployed on the Internet, and more are likely to come. The amount of the service users, however, grows in a nonlinear manner. A major challenge faced service-oriented applications is to maximize the utility of those available web services and support the largest number of users. To handle this issue and guarantee the quality of services, researchers propose the idea of service reservation, which is usually implemented by a SLA (Service Level Agreement) protocol. Many service vendors have widely used the scheme of service reservation to improve their profits now, such as airline companies and hotels [1].

In reality, reserving only one web service is no solution to dealing with more complicated requirements which are expressed as business processes. The reasons for this are manifold, but typically relationships between services (such as airline-hotel partnerships) as well as different frame parameters (such as different reservation times) that have an influence on overall requirements (such as cost).

As service oriented applications can be built from single or composite services, we can conclude that the reservation schemes can also be divided into service reservation and process reservation. Many service-oriented applications show that it is more desirable if all web services required for a process can be reserved in advance. Most customers need process level reservation to get complete results (for example trip reservation or scientific workflows). However, less effort

has been paid to the scheme of process reservation in which the service relationships impose constraints on the reservation of each related service.

In this paper, we propose a novel process reservation scheme, named BPSR (Business Process Service Reservation) to solve this problem and increase the profit of both the service provider and the service consumer. The process reservation scheme is a promising solution for the seamless integration of many services to create new value-added services for applications.

## II. METHODOLOGY

### A. Overview of BPSR Architecture Framework

The proposed framework for process reservation, depicted in Fig 1, is based on three key components: Policy Generator, Process Reserator and QoS Controller.

Briefly the Policy Generator is concerned with identifying the customer flexibility and expressing this for the Process Reserator which will provide the user with options of the whole process and their cost. The QoS Controller ensures that current QoS information is being used.

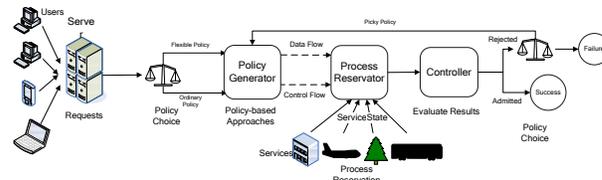


Fig 1 Framework Architecture

### B. Service Reservation

A service is the basic unit of functionality in service oriented computing, it is self-described, can be discovered and services can be joined into larger groups called composite services. We define a service as follows:

#### DEFINITION 1 (Service)

A service is defined as a 3-tuple  $S(I, O, Q)$ , where

$I$  represents the input ports of a service;

$O$  represents the output ports of the service;

$Q$  represents the quality constraints of service;

Service reservation (SR) is fundamental to process reservation and involves three aspects: service discovery, service matching and marking. SR is defined as follows:

## DEFINITION 2 (Service Reservation)

$SR$  is defined as a tuple  $SR(\varepsilon, R)$ , where

$\varepsilon$  is a set of services  $(S_1(I_1, O_1, Q_1), \dots, S_n(I_n, O_n, Q_n))$

$R$  is a set of the same cardinality with  $R = \{0, 1\}^n$ ,

Where 1 denotes that the service has been marked.

### C. Process Reservation

Process reservation ensures that all services needed for a process are reserved – trying to optimize on QoS attributes. Process reservation makes use of policies capturing the user’s flexibility. The policy reservation process uses the results from service reservation, but with the possibility of adjusting these and will ultimately commit to a set of services.

## DEFINITION3 (Process Reservation)

Process Reservation (PR) is defined as a 3-tuple  $PR(C, \varepsilon_{sr}, R_p)$ , where

$C$  represents the composition workflow;

$\varepsilon_{sr}$  represents the service set after SR (each service has an SR mark);

$R_p$  represent a matrix with pair wise dependencies between services, with each entry  $r \in \{\text{Substitution, Contains, adviceTo, Prerequisite, Replace, Exclude}\}$ ;

The rules for process reservation are expressed in terms of keywords showing relationships between services. The relationships can be calculated using semantic distance [2].

We need to consider two different situations with regard to the number of users. (1) For a limited numbers of users, we can assume that there are sufficient service resources available, so we will try to find best reservation for a user. Our method considers service relationships as defined earlier. The method searches for the services that fulfill the workflow and ensure the best QoS. (2) For an unlimited number of users, we cannot assure the best choice of services for all users because there might be a shortfall of resources. A policy-based approach that allows users to express their demands for the process reservation allows to optimize between best service and best utilization rate of service resources. There are three types of policy: (1) ordinary policies with a standard reservation cost and presenting a fixed solution, (2) flexible policies allowing users to express that they are flexible on certain aspects, encouraged by a discount, and (3) picky policies allowing the user to pay extra for getting a higher priority in a reservation to ensure that he gets exactly what he asks.

### III. EXPERIMENTS AND EVALUATIONS

Based on the PR architecture, we implemented a service oriented reservation simulation for the travel process. We consider 3 QoS attributes: cost, response time and throughput. There are 10 services to reserve one-way airline tickets, 10 hotel reservation services and 4 sightseeing tour operators. Prices of service instances are randomly generated within boundary intervals as shown in Table 1. When there exists a “substitution” relation among services, the composite

QoS data is improved by 20 percent. The “adviceTo” leads to a 30% discount and the “Exclude” comes with unlimited premium cost.

TABLE I. ATTRIBUTES OF SIGHTSEEING TICKET ORDERING PROCESS

Service	No of Instances	Price range
Airline Ticket	10	600-800
Hotel Booking	10	300-400
Sightseeing Ticket	4	300-400
Return Ticket	10	600-800

We compare the results gained to those of a standard reservation approach. The metrics for comparison are the average reservation cost, the success rate and the resources utilization of the reservation. We found that a “BPSR” reservation improves vendor’s profits and customers’ satisfaction. For example, Fig 2 presents the average cost of the process, with error bars along the curve of the BPSR approach. The results show that the average cost of process reservation considering the relationships of services decreases with an increase in the relationship between services.

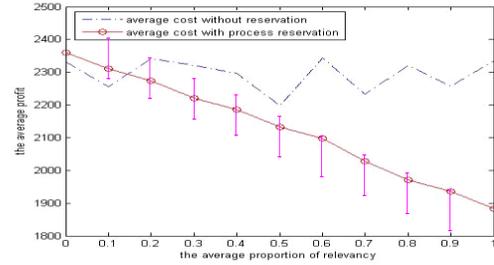


Fig 2 Average cost of different relevance for limited users

### IV. CONCLUSION

In this paper, we presented a process reservation method (BPSR) – both the concept of process reservation as well as the method are novel contributions. This method is based on a process reservation architecture that integrates service differentiation, service reservation, process reservation and QoS control. We focus on the reservation method for the process level. Compared with service reservation, process reservation considers all candidate services as part of a process as a whole, and gains a higher probability to make a successful reservation. In future work we plan to explore identification of process bottlenecks as this could hint at service shortfalls and hence business opportunities. We will also consider whether better QoS data prediction can be achieved using results such as [3].

### REFERENCES

- [1] K. Miyashita, K. Masuda, F. Higashitani. Coordinated Service Allocation through Flexible Reservation. TSC2008, Vol.1, No.2. pp: 117-128.
- [2] C. Bousquet, M.C. Jaulent, G. Chatellier, P. Degoulet. Using semantic distance for the efficient coding of medical concepts. Journal of the American Medical Informatics Association, 2000, pp: 96-100.
- [3] H. Ma, I. King, and M.R. Lyu. Effective missing data prediction for collaborative filtering. In SIGIR, ACM, 2007, pp: 39-46.