http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

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

Information Technology Journal 10 (5): 1024-1030, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1024.1030
© 2011 Asian Network for Scientific Information

Service Selection Constraint Model and Optimization Algorithm for Web Service Composition

[1,2]Xue-Long Wang, [1]Zhang Jing and [2]Huai-zhou Yang
[1]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China
[2]School of Computer Science, Xi'an Shiyou University, Xi'an, 710065, China

Abstract: The Web service composition system with static configuration can not adapt to the failure-prone environment and the variable Quality-of-Service (QoS) of component services. Therefore, a dynamic configuration method of Web service composition, Service Selection Constraint Model, is presented in this study. The candidate component services with same functionality are organized as a service class. The functional dependency relationships between component services are reflected as the service selection constraints. The optimal configurations conforming to multi-objective QoS constraints, known as the Pareto optimal solutions, are searched by a special ant colony optimization algorithm for Web service (ACO4WS). The feasibility and soundness of the method are proved by simulation experiments and corresponding analysis. By using the presented method, not only the QoS of service composition system is greatly improved, but also the multiple functional and non-functional constraints are satisfied.

Key words: Web service, dynamic configuration, modeling, ant colony optimization

INTRODUCTION

The Web service composition provides a mechanism to construct a new value-added service via the integration and interaction of the existing component services. In contrast with the traditional component systems, in Web service composition, there probably exist a lot of candidate component services, which provide the same functionality but differ in QoS attributes. The service composition becomes a decision problem on which component services should be selected such that the user's functional and non-functional requirements can be satisfied. In the Internet-based environment, the QoS of composite service is fluctuating due to the quality of the component services are subject to change with the variation of workload or communication delay (Chafle *et al*., 2007). In error-prone environment, most services are third party services whose QoS is not guaranteed. The malfunction of component service can disable a composite service. Therefore, the Web service composition system must have the ability of dynamic reconfiguration in order to adapt to the dynamic change of run-time environment (Yen *et al*., 2008).

In this study, in order to satisfy the requirements of self-adaptive Web service composition and automatic management, a method of service selection constraint for Web service composition is presented. Each abstract activity of service composition is respectively associated to a corresponding service class. Each service class is comprised of the concrete candidate component services with same functionality. The functional dependency relationships between component services are reflected as the service selection constraints between service classes. Finally, a special ant colony optimization algorithm for Web service (ACO4WS) is designed and applied to detect the optimal configurations which satisfy the service selection constraints and multi-objective QoS requirements.

RELATED WORKS

Although the problem of Web service composition optimization has been extensively researched in the past few years, most works regard each service selection as an independent process. However, in actual applications, there often exist dependent relationships between component services. Xiong *et al*. (2008) presented a functional dependency configuration net to reflect the functional dependency relationships (Xiong *et al*.,2008). An optimal configuration can be dynamically got under the precondition that the functional dependency relationships are satisfied. Unfortunately, their research is only suitable for mono-objective QoS optimization. Gooneratne *et al*. (2007) classified the QoS constraints

Corresponding Author: Xue-Long Wang, School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China

as local and global constraints. The former restricts the values of a particular attribute of a single service, whereas the latter simultaneously restricts the values of two or more attributes of multiple candidate services. A global constraint is strictly dependent if the values of restricted QoS attributes can be uniquely determined once a value is assigned to one of them. A composite service which conforms to strictly dependent global constraints can be easily optimized in polynomial time (Gooneratne *et al.*, 2007; Fang *et al.*, 2009). Associate Petri net (APN) (Fang *et al.*, 2009) and extended color Petri net (eCPN) (Liu *et al.*, 2009a) are used to describe the independent global constraints. Genetic algorithm is applied to search an optimal service composition (Liu *et al.*, 2009a; Fang *et al.*, 2009). The solutions presented in these works still remain some obvious flaws. First, the optimization processes are often dependent on some simplified QoS evaluation functions. These functions are too rough to well evaluate whether the user's QoS requirements are satisfied or not. Second, the functional dependency relationships are hard to be reflected according to above mentioned works. Because the connection relationships between component services are determined by the flow structure of service composition and because the flow constraints are not enough considered in these works, it is difficult to make certain that between which component services there exist functional dependency relationships.

Many meta-heuristic algorithms, such as genetic algorithm (Fang *et al.*, 2009), particle swarm optimization (Liu *et al.*, 2008) and simulated annealing (Liu *et al.*, 2009b), have been applied to accelerate the detection of the optimal service composition. These approaches exhibit the same flaws, i.e., the weak modeling of service constraints and the rough algorithm implementation. The important problem of service selection conflict (Huang *et al.*, 2009) is not solved in these algorithms. Because the meta-heuristic algorithms are only some algorithm frameworks, they often need some modifications to adapt to the special problems. In the above mentioned works, it is not clear what algorithm modifications have been done to adapt to the web service composition optimization. The performance of these algorithms also needs to be improved.

THE SERVICE SELECTION CONSTRAINT MODEL

The service selection constraint model is used to reflect the service selection constraints between component services. We organize the services with same functionality as a service class. The service selection of a service class is often dependent on the service selection of its previous service class. These related concrete services should be selected as a whole in this situation.

Definition 1: The service class: A service class is a service set which is comprised of a number of candidate component services with same functionality.

Definition 2: The service connection: Let C_1 and C_2 represent two service classes which are related to service transition tq_1 and tq_2 respectively. If there is a path from tq_1 to tq_2 and if there are no other service transitions in this path, then there is a service connection $C_1 \rightarrow C_2$, which means that C_1 and C_2 are adjacent and C_1 is a previous service class of C_2.

Definition 3: The service selection constraint model of composite service: A service selection constraint model of composite service is a tuple (S, C, R_c, R_S), where:

- S is a set of component services. C is a set of service classes
- $S = \bigcup_{i=1}^{n} C_i, C_i \in C, n = |C|. \forall C_i \in C, C_i \neq \varnothing. \forall C_1, C_2 \in C, C_1 \cap C_2 = \varnothing$
- $R_c \subset C \times C$ is a set of service connections. If $<C_1, C_2> \in R_c$, then there is a service connection $C_1 \rightarrow C_2$
- $R_s \subset S \times S$ is a set of functional dependency relationships of component services
- If $<s_1, s_2> \in R_s$, then $\exists C_i, . C_j, \in C, s_1 \in C_i, s_2 \in C_j, C_i \neq C_j, <C_i, C_j> \in R_c$. Two component services with functional dependency relationship respectively belong to two different service classes with service connection
- To $\forall C_1, C_2 \in C$, if $C_1 \rightarrow C_2$, then $\forall s_1 \in C_1, |\{<s_1, s>| s \in C_2\}| \geq 1$ and $\forall s_2 \in C_2, |\{<s, S_2>| s \in C_1\}| \geq 1$. To any component service, there exist at least one pre-service and one post-service which have functional dependency relationships with this component service

The service selection constraint model is illustrated in Fig. 1b. It is a directed acyclic graph (DAG). In Fig.1a, the process model is used to reflect the flow constraints of composite service and it can be regarded as a composition plan which is comprised of some abstract services. In the process model of composite service, the system states and the execution conditions are represented by places. The logical control activities are represented by dummy transitions. The activities of component services are represented by service transitions. The task of composite service is represented by token. Every service transition is associated with multiple QoS attributes. The flow constraints are compactly reflected by the process model. A process model of composite service can be transformed to a DAG.

Fig. 1: The illustration of the service selection constraint model. (a) The process model and (b) the service selection constraint model

In the DAG, the set of vertexes is C and the set of directed edges is R_c. There are two kinds of constraints between service classes, mono-constraint and multi-constraint. For example, C_2 is only constrained by C_2 while C_9 are simultaneously constrained by C_3, C_4 and C_8. A service connection includes many service functional dependency relationships. The problem of composite service is how to select a suitable component service from each service class under the precondition that functional dependency relationships and multi-objective QoS constraints are satisfied simultaneously.

ACO4WS

Ant Colony Optimization (ACO) (Dorigo and Caro, 1999) has exhibited its effectiveness to solve several different NP-hard combinatorial optimization problems, such as traveling salesman problem (TSP) (Dorigo and Gambardella, 1997). In ACO, the optimization problems are often modeled by a construction graph. The artificial ants lay pheromone on edges and/or vertices when they walk randomly on the graph. The probability to select a path is determined by the pheromone which has been laid on the path by the ant colony. The pheromones decrease progressively via., evaporation. The optimal solutions are detected through pheromone-based indirect communications between artificial ants. In this study, a special ant colony optimization algorithm ACO4WS is designed to achieve service composition optimization under multiple functional and non-functional constraints. We suppose the readers have the basic concepts of ACO. The characteristics of ACO4WS are presented as:

The multi-objective programming of service composition: The selection of optimal service configuration can be described as a multi-objective programming (MOP):

$$\text{Min}\, f(x) = (f_1(x),\ldots,f_p(x))^T, \tag{1}$$

$$\text{subject to, } g\,(g) \le 0,\ I \in \Gamma, \tag{2}$$

$$x \text{ conforms to } R_s. \tag{3}$$

Vector $x = (x_1, x_2, \ldots, x_n)^T$ is called decision variable. x_i ($I = 1,2, \ldots,n$) represents the service class C_i. The value of x_i is determined by which component service is selected from C_i. Vector function f(x) is comprised of multiple QoS objective functions $f_i(x)$ ($i = 1,2, \ldots,p$). g_i (x) represents multiple QoS constraint functions. The set of constrained QoS attributes is denoted by Γ. F_i (x) and g_i (x) can be constructed with the help of the QoS model of composite service. Condition (3) indicates the functional dependency relationships between component services need to be satisfied. R_s has been defined in Definition 3. Given a feasible solution x^*, if there does not exist another feasible solution x satisfying $f(x) \prec f(x^*)$, then x^* is an efficient solution of MOP (i.e., the Pareto optimal solution) (Guntsch and Middendorf,2003). ACO4WS is designed to find the Pareto optimal solutions in this study.

The construction of optimal service configuration: When selecting the services in DFS&SSCA, the QoS of component service is not considered. In order to search the service configuration with optimal QoS, ACO4WS is designed through adding pheromone-based optimization mechanism (Dorigo *et al*., 1996) to DFS&SSCA. In DFS&SSCA, the set of current candidate component services is determined by following expression:

$$H = \begin{cases} \{s \mid s \in C_c \wedge (\forall s' \in M, <s,s'> \in R_s \vee <s',s> \in R_s)\} & \text{if } C_c \text{ is not initial service class} \\ C_c & \text{if } C_c \text{ is initial service class} \end{cases} \tag{4}$$

where, Cc is the service class being visited by the ant. M is a set of component services. Every $m \in M$ is a component service which has been selected from visited service class C_m. There must exists service connection relationship between C_m and C_c, i.e., $C_m \rightarrow C_c$ or $C_c \rightarrow C_m$.

The service selection based on pheromone: Because the goal of ACO4WS is to select suitable component services for service composition, pheromone is attached to every component service in ACO4WS. $H_{C_c}^k$, the set of current candidate component services, is still determined by expression (4) when an ant k visits a service class C_c. But free service selection labeled with "SS:" in DFS&SSCA is changed to probabilistic service selection. The probability to select a service $H_{C_c}^k$ is:

$$P_i^k = \frac{[\tau_i]^\alpha[\eta_i]^\beta}{\sum_{j\in H_{C_c}^k}[\tau_j]^\alpha[\eta_j]^\beta} \quad (5)$$

where, τ_i represents the pheromone of service i. $\eta_i = 1/\xi_i$ represents the heuristic information of service i. α and β are two parameters, which determine the power of influence of pheromone and heuristic information respectively. ξ is a weighted sum of multiple QoS attributes. The smaller is the value of ξ, the better is the QoS. The weights and ξ can be calculated as following steps (Hwang and Yoon, 1998):

- Construct a matrix $Q = (q_{ij})_{n\times m}$. q_{ij} represents the QoS attribute j of service i. M is the quantity of QoS attributes. N is the quantity of all component services. To larger-is-better QoS attributes, such as throughput and reliability, q_{ij} is the reciprocal of these QoS attributes' values. For example, q_{ij} = 1/throughput
- Normalize Q to $R = (r_{ij})_{n\times m}$ to allow a comparable scale for all QoS attributes
- Calculate R to get a column normalization matrix:

$$r'_{ij} = r_{ij} / \sum_{i=1}^{n} r_{ij}$$

- Calculate the information entropy of every QoS attribute:

$$E_j = -\frac{1}{\ln n}\sum_{i=1}^{n} r'_{ij}\ln r'_{ij}$$

- Calculate the weight of every QoS attribute:

$$\omega = (\omega_1, \omega_2, ..., \omega_m), \quad \omega_j = (1-E_j) / \sum_{k=1}^{m}(1-E_k)$$

- After getting the weights of QoS attributes, ξ_i is calculated as:

$$\xi_i = \sum_{j=1}^{m} r_{ij}\omega_j$$

The initial pheromone τ_0 needs to be set as a value which is slightly larger than the average pheromone laid by the ants in an iteration cycle (Dorigo and Caro, 1999). The value of τ_0 can be estimated as $\tau_0 = m/\xi^{nm}$. m represents the quantity of ants. ξ^{nm} can be calculated as following steps:

Step 1: Run DFS&SSCA once and select the component service with the smallest ξ in every selection

Step 2: Calculate the value of every QoS attribute of composite service according to the selected component services and corresponding QoS model

Step 3: Calculate the ξ of composite service and let ξ^{nm} equal the result value

Every ant k keeps a memory unit Q^k which records the visited service classes and selected component services.

The pheromone updating: At the end of an iteration cycle, when every ant has constructed a solution, the pheromone of every service will be updated. First, the pheromone of every service decreases a value through the evaporation of pheromone. Second, pheromones are added to the services, which have been selected by the ants. The evaporation of pheromone is accomplished according to following expression:

$$\tau^i \leftarrow (1-\rho)\,\tau^i \quad (6)$$

where, ρ represents the evaporation rate of pheromone. $0<\rho\le 1$.After the evaporation of pheromone, every ant lays pheromone on the services selected by it:

$$\tau_i \leftarrow \tau_i + \sum_{k=1}^{m}\Delta\tau_i^k \quad (7)$$

where, $\Delta\tau_i^k$ represents the pheromone which is laid by the ant k on the services selected by it. $\Delta\tau_i^k$ is defined as:

$$\Delta\tau_i^k = \begin{cases} 1/\xi^k, & \text{if service i is selected by ant k;} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

where, ξ^k represents the QoS of the service configuration selected by ant k. If a service is selected by many ants and the configurations including this service have better QoS, then more pheromone will be laid on this service.

```
Input:A service selection constraint model and a service composition QoS model
Output:A QoS optimization configuration conformable to the MOP constraints
Begin
  BestConfig.QoS =+∞
  BestConfig.Config = NULL  //BestConfig is a structure variable to store the configuration with best QoS.
   Set the quantity of ants as m.
   Set the iteration times as x.
   Calculate the QoS attribute weights $\omega = (\omega_1,\omega_2,...,\omega_n)$.
   Calculate $\xi_i$ for every component service.
   Calculate $\xi^{nm}$ and set initial pheromone $\tau_0 = m/\xi^{nm}$ for every component service.
   For j =1 to x
     For k =1 to m
       Pick a service class randomly and lay the ant k on it.
       Run DFS&SSCA, select services according to expression (5) and construct a configuration $Q^k$.
     Next
     For each service i, the pheromone of is updated to $\tau i \leftarrow (1-\rho)\tau i$.
     For k=1 to m
      Calculate every QoS attribute of composite service according to configuration $Q^k$ and QoS model.
      If MOP QoS constraint condition (2) is satisfied then
        Calculate $\xi^k$ of composite service under configuration $Q^k$.
        For each service in configuration $Q^k$, the pheromone of I is updated to $\tau_i \leftarrow \tau^i + 1/\xi^k$
      If $\xi^k <$ BestConfig.QoS Then
          BestConfig.QoS = $\xi^k$
          BestConfig.Config = $Q^k$
        End If
      End If
   Next
  Next
  Return BestConfig
End
```

Algorithm 1: The implementation of ACO4WS

The implementation of ACO4WS: The implementation of ACO4WS is described in Algorithm 1. In ACO4WS, the ants construct the solutions via DFS and SSCA, which guarantees these constructed solutions satisfy the MOP constraint condition (3). In addition, whether the solutions satisfy the MOP QoS constraint condition (2) is verified in ACO4WS. Only when a solution satisfies the MOP QoS constraint condition (2), the corresponding ant is permitted to add pheromone to the services selected by it. After running ACO4WS, we can get an optimal service configuration.

EXPERIMENTS ON SERVICE CONFIGURATION OPTIMIZATION

To evaluate the efficiency of the proposed modeling method and ACO4WS, we take Fig. 1a as an abstract service composition example and optimize its configuration through simulation. We generate a number of candidate component services for each activity of service composition. In every service class, suppose the QoS attributes of services conform to normal distribution, we randomly generate the values of QoS attributes for each service. Two kinds of QoS attributes, response time and cost, are considered in the experiments. The distribution parameters of QoS attributes are shown in Table 1.

In the experiment example, the probabilities to choose t_3 and t_4 are set to 0.5. The loop probability to choose t_6 is set to 0.5. We can define the corresponding QoS mode as follow:

$$\text{Response time} = R_t(t_1) + \text{Max}\{[R_t(t_2) + 0.5R_t(t_3) + 0.5R_t(t_4)], 2R_t(t_5) + R_t(t_6) + R_t(t_7) + R_t(t_8)]\} + R_t(t_9) \tag{9}$$

$$\text{Cost} = \sum_{i=1}^{9} \text{Cost}(t_i) \tag{10}$$

The QoS constraints of composite service are set as Response time<35 and Cost<50. We randomly generate the functional dependency relationships between the component services with service connections. A service is associated to a random number of post services. According to the research in (Dorigo and Caro,1999), α and β in expression (5) are set as 9 and 10, respectively.

When the evaporation rate of pheromone is set as $\rho = 0.3$, the quantity of ants is set as $m = 3$ and the quantity of candidate services in each service class is set as 5, the optimization process of ACO4WS is shown in Fig. 2. After each iteration cycle, three service configurations are selected by the ants. The average QoS values of the selected configurations are calculated. ξ is a weighted sum of multiple QoS attributes. The experiment results indicate that the multi-objective QoS optimization of service composition can be achieved by using ACO4WS. Along with the increase of iteration times, the service configurations selected by the ants gradually tend to an optimal configuration.

In order to evaluate the performance of ACO4WS, we set the quantity of candidate services in each service class as different values and record the average execution time of an iteration cycle. The quantity of ants (m) is changed in each group of experiments. The experiments are made on a desktop computer with 2.4 GHz CPU and 1 GB RAM. The experiment results are shown in Fig. 3. The experiments indicate that the execution time of algorithm is mainly determined by the quantity of candidate services in each service class. Along with the increase of candidate services in each service class, the execution time of algorithm increases rapidly due to the exponential growth of service dependency relationships. Therefore, ACO4WS still need to be improved to adapt to the situation that there are a lot of candidate services in each service class.

Table 1: The distribution parameters of response time and cost

Parameters	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Response time									
Mean (μ)	8.5	6	11.5	10	4	3	7	11	5
SD (σ)	7	5	10	8	3	2	6	7.5	4
Cost									
Mean (μ)	8.5	8.5	5	5	4	3	7	6	3
SD (σ)	7	8	4	3	3	2	6.5	5	2

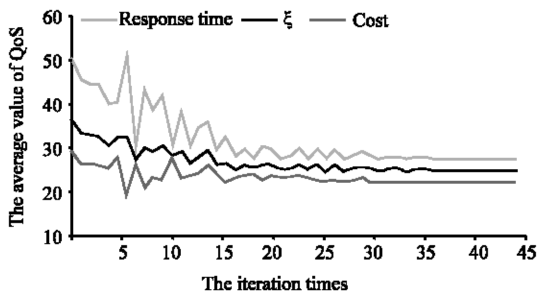

Fig. 2: The optimization process of ACO4WS

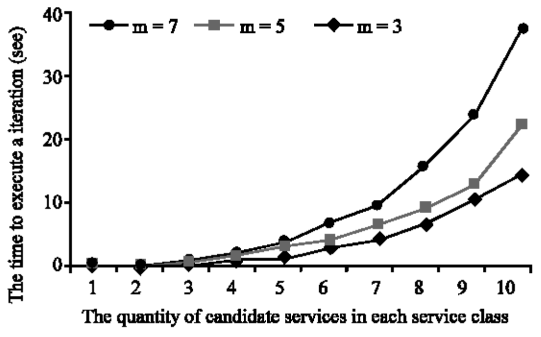

Fig. 3: The performance of ACO4WS

CONCLUSIONS

Due to the failure-prone run-time environment and the inconstant QoS of component services, a composite service must be able to be configured dynamically. In order to satisfy the multiple functional and non-functional constraints, suitable component services need to be selected for service orchestration process. We propose a service selection constraints configuration modeling method, which synthetically reflects the flow constraints and the functional dependency relationship constraints of composite service. Due to the complexity of multiple constraints, a special ant colony optimization algorithm ACO4WS is designed to accelerate the detection of optimal configuration. By using ACO4WS, We can get an optimal service configuration which strictly conforms to multi-objective QoS constraints and service functional dependency relationships.

ACKNOWLEDGMENT

This research was partially sponsored by the National High-Tech Research and Development Plan of China under Grant No.2007AA010305 and Science and technology project of Shaanxi Province under Grant No.2006K04-G10.

REFERENCES

Chafle, G., S. Chandra, N. Karnik, V. Mann and M.G. Nanda, 2007. Improving performance of composite web services over a wide area network. Proceedings of IEEE Congress on Services, July 9-13, IEEE Computer Society, pp: 292-299.

Dorigo, M. and G. Di Caro, 1999. The Ant Colony Optimization Meta-Heuristic. In: New Ideas in Optimization, Corne, D., M. Dorigo and F. Glover (Eds.). McGraw Hill, New York, ISBN: 0077095065, pp: 11-32.

Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput., 1: 53-66.

Dorigo, M., V. Maniezzo and A. Colorni, 1996. The ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybernet., 26: 29-41.

Fang, X.W., C.J. Jiang and X.Q. Fan, 2009. Independent global constraints for web service composition based on GA and APN. Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC), June 12-14, Shanghai, China, pp: 119-1265.

Gooneratne, N., Z. Tari and J. Harland, 2007. Matching strictly dependent global constraints for composite web services. Proceedings of the European Conference on Web Services, Nov. 26-28, Washington, DC, USA., pp: 139-148.

Guntsch, M.G. and M. Middendorf, 2003. Solving multi-criteria optimization problems with population-based ACO. Lecture Notes Comput. Sci., 2632: 464-478.

Huang, A.F., C.W. Lan and S.J. Yang, 2009. An optimal QoS-based web service selection scheme. Inform. Sci., 179: 3309-3322.

Hwang, C.L. and K. Yoon, 1981. Multiple Attribute Decision Making and Applications. Springer-Verlag, New York.

Liu, L.P., Z.G. Chen and A.X. Liu, 2008. Research on web services composition based on particle swarm optimization. Chin. J. Comput. Eng., 34: 104-112.

Liu, X.W., Z.C. Xu and L. Yang, 2009a. Independent global constraints-aware web service composition optimization based on genetic algorithm. Proceedings of International Conference on Industrial and Information Systems (ICIIS), April 24-25, Haikou, China, pp: 52-55.

Liu, Q., S.L. Zhang, R. Yang and X.J. Lian, 2009b. Web services composition with QoS bound based on simulated annealing algorithm. Chinese J. Southeast Univ., 24: 308-311.

Xiong, P.C., Y.S. Fan and M.C. Zhou, 2008. QoS-aware web service configuration. IEEE Trans. Syst. Man Cybern., 38: 888-895.

Yen, I., H. Ma, F.B. Bastani and H. Mei, 2008. QoS-reconfigurable web services and compositions for high-assurance systems. Computer, 41: 48-55.