Asian Journal of Information Technology 13 (5): 282-286, 2014
ISSN: 1682-3915

Architecture for Service Selection Based on Consumer Feedback (FBSR) in Service Oriented Architecture Environment

[1]S. Thangam, [2]E. Kirubakaran and [3]J. William
[1]Anna University, Chennai, India
[2]Bharat Heavy Electricals Limited, Tiruchirappalli, Tamil Nadu, India
[3]MAM College of Engineering and Technology, Tiruchirappalli, Tamil Nadu, India

Abstract: Internet has become an important media in world. Web servies can be located, published and invoked through web. There are number of services published by various organizations. The appropriate service required by the user is important. Selecting this appropriate service is an challenge. This study proposes a new methodology Feedback Based Service Ranking with QoS (FBSR-Q) which helps the consumers to select the appropriate service as per their quality requirement. Service discovery results in more than one service based on user request. This study proposes an architecture which enhances the UDDI. Depending on the execution environment, the services are assigned priority and then a mathematical model PROMETHEE is used to rank the services and further based on the consumer feedback the ranking of services are further updated. The performance of proposed architecture outperforms with respect to functional and non functional web services.

Key words: UDDI, PROMTHEE, ranking, feedback, service selection

INTRODUCTION

SOA is the methodology for achieving interoperability and reuse. SOA is realized with help of Web services (Fig. 1). The services need to communicate with one another. Web services which are a software system which supports interoperable machine to machine interaction over network. Web services developed, deployed and published mean nothing unless the consumer can search, locate and bind them. There are three important participants as web service provider, agency or middleware interacting with registry and web service consumer forming a triad.

Web service provider defines the WSDL of the web service and an interface to access it. It can be directly given to consumer but it is not an feasible approach since provider do not know who is the potential consumer. So, WSDL is provided to discovery agency who publishes it. Discovery agency is associated with UDDI which maintain detail of service published. When a consumer request a service, the find operation is initiated to retrieve the WSDL from the agency. Using WSDL the consumer binds with service provider. Since, multiple services are returned back an appropriate mechanism to do effective service selection is required (Blum and Carter, 2004).

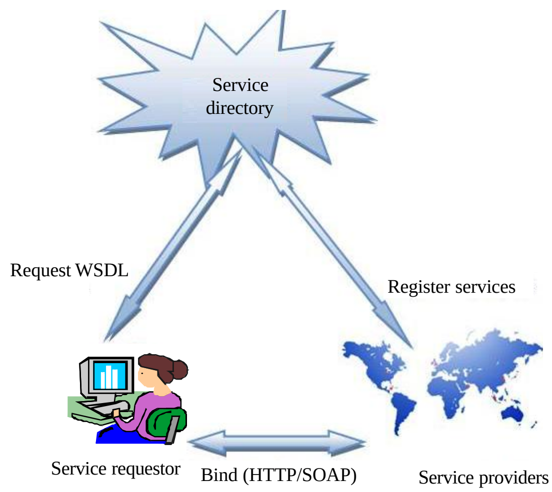

Fig. 1: Web service architecture

Literature review: Enhancement of service discovery can be done in three locations. First at customer side, second at producer side and third at UDDI. Many researches has been done on three sides. First at consumer side, it increases the complexity of the consumer and if old selection algorithms have to be updated all the consumers have to be updated. At the producer side there is no guarantee that they will provide the QoS stated by them. UDDI seems to be the reliable and efficient place where the service selection and discovery takes place. Major efforts include WSLA by IBM, WS Policy and ontology web language for service.

Corresponding Author: S. Thangam, Anna University, Chennai, India

Zou *et al*. (2009) proposed a web service description model that considers service QoS information and then present an overall service selection and ranking framework with QoS (WSSR-Q). Cao *et al*. (2005) proposed cost reduction driven Web service selection. Cost was taken as primary concern. Genetic algorithm was used for optimization. The selection model based on multi-agent platform optimization lack in performance and time.

Liu *et al*. (2005) proposed a strategy GODDS (Global Optimization of Dynamic Web Service Selection) to realize web service selection with QoS global optimization. Multiple constraints such as cost, time, reliability and reputation were considered. Tao *et al*. (2007) proposed efficient algorithms for web service selection with end to end QoS constraints. Researcher used Heuristic algorithm (MMKP and MCOP). The QoS metrics considered were cost, time, availability, reliability and performance. The problem is modelled in two ways as combinatorial and graph model. The algorithm was designed for 2 flow structures as one for service process with sequential flow structure and other for composition structures including loops, conditional and parallel operations.

Xu *et al*. (2007) presented a model of reputation-enhanced QoS-based web service discovery which combines UDDI registry to publish the QoS information and reputation scores are assigned to services. It failed to filter the services prior to rankin. Liu *et al*. (2004) proposed QoS computation model. But the QoS metrics values are not limited in a definite range. The problem with this model is internal impact is less since its QoS value is small (Liu *et al*., 2004).

Chen *et al*. (2003) proposed broker based architecture QCWS which deploys a QoS broker between web service clients and web service providers. This architecture proposes a QoS broker collects the QoS information about service providers that may offer qualified web services to client (Chen *et al*., 2003). Ran (2003) proposed a web services discovery model with functional and nonfunctional requirements and concept of the certifier is introduced.

Kritikos and Plexousakis (2009) focused on analysing the requirements of a semantically rich QoS-based WSDM and provided SW and constrained based mechanisms for enriching syntactic QoS-based WS Discovery (WSDi) algorithms. The roadmap of extending WS standard techniques for realizing semantic, functional and QoS-based WSDi was presented. Menasce (2002) explained various QoS issues in web services. Zhu *et al*. (2008) utilizes certain rules to sort the examples of Web services found by dynamic adaptive template.

Liu *et al*. (2012) mentions that Service Oriented Architecture enable a multitude of service providers to provide loosely, coupled and interoperable services at different Quality of Service (QoS) as more and more Web services become available, QoS is becoming a decisive factor to distinguishing composite Web services. A New Optimization algorithm called C-MMAS is proposed by integrating Max.-Min. Ant System into Culture algorithm framework and is used to solve the problem of composite web services selection.

MATERIALS AND METHODS

This architecture provides service selection model which provides the required services to the requestor based on their QoS requirements. The proposed architecture is shown in Fig. 2. Components of architecture are explained:

Web service provider: The web service provider creates the web service with corresponding QoS and stores the WSDL file in service repository.

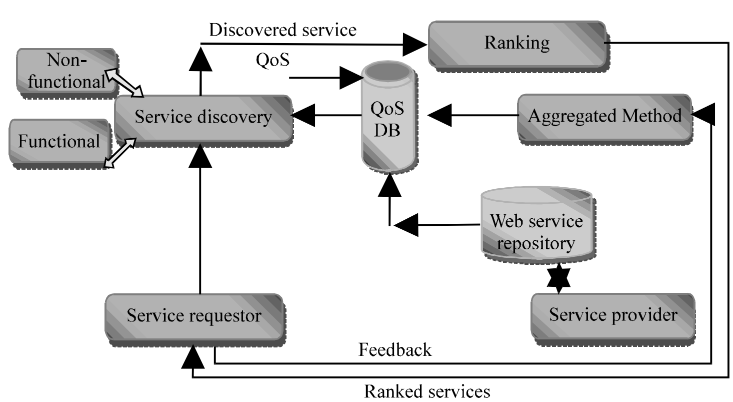

Fig. 2: Web service architecture

Table 1: Web service selection procedure

Web service	Selection procedure
Service consumer	Step 1: The service requestor requests a service based on QoS
	Step 2: Services are selected from service database
	Step 3: The services are selected based on functional and non functional requirement
Service producer	Step 1: The service producer publishes the web service
	Step 2: Service details are stored in service data base
	Step 3: The services QoS are stored in QoS database
Service Selection Model	Step 1: The Services based on functional attributes the services are selected
	Step 2: The servies are ranked based on promthee methodology.
	Step 3: The service data base is updated based on this ranking
	Step 4: The feedback from the customer is considered and based on the aggregated ranking the results are returned back to the customers

Table 2: Notation explanations

Notation	Description
S_D	Discovered service set
Q_R	QoS requirement
S_S	Selected Web service
S_x, S_y	Two services x, y
Pr	Preference function
Π	Aggregated preference
$\Pi(S_x, S_y)$	Preference of service x over service y
ϕ^+	Positive outranking flow
ϕ^-	Negative outranking flow
$C_1, C_2, ...$	Criteria
$W_1, W_2, ...$	Weights assigned to services

Web service requestor: The service requestor requests the service with user preferences.

Service data base: The services created by the producer are registered in service database.

QoS database: Based on the user defined QoS the values are stored in the database. Also, based on the feedback of the customers it is updated.

Service ranking-promthee: A method PROMTHEE is used to rank the services and the best service is delivered to the client.

Aggregated Method: Based on the feedback got from the client the aggregated method is used to aggregate results of promthee and feedback of consumer.

Quality rating: The updated ranking from Promthee and aggregated method is finally stored in QoS database. The service provider registers the available services in service data base of UDDI. The consumer when it requires a service places its request. The service database is fetched for the service which matches the keywords and the services selected are further refined using PROMTHEE methodology. The ranked services are returned back to the consumer. Further based on the feedback the services database is updated (Table 1).

Web Service Selection algorithm: Table 2 explain the notations used in the methodology.

Algorithm for proposed service discovery framework:

Input: SD and QR
Output: SS

Step 1: Get the information from the user: The weighted user preference for each criteria Cr provided by user is retrieved and a set $\{W_j, j = 1, 2, ..., k\}$ representing the preference of different criteria in terms of weight is tabulated and normalized to 1

Step 2: Get the information from service provider: An evaluation table is formed in service provider side. It contains the corresponding services weight

Step 3: Compute preference function $Pr_j(S_x, S_y)$ for each user requirement based on pair wise comparison: Preference function gives the decision maker a preference of a action a with regard to b. The value is between 0 and 1

$Pr_j(S_x, S_y) = F[dj(S_x, S_y)] \quad S_x, S_y \quad S$

Where, $dj(S_x, S_y) = Cj(S_x)\text{-}Cj(S_y)$

for $0 \le Pr_j(S_x, S_y) \le 1$

The intensity of preference increases linearly until deviation equal m and after the preference is strict

Step 4: Compute aggregated preference indexes $\pi(S_x, S_y)$: Let S_x, S_y S where, S is set of services, then aggregated preference index is given by:

$$\pi(S_x, S_y) = \sum_{j=1}^{k} Pr_j(S_x, S_y) W_j$$

where, $\pi(S_x, S_y)$ express the degree in service Sx is preferred over S_y

Step 5: Compute positive outranking flow and negative outranking flow: Each service S_x is compared with n-1 services. Positive outranking flow is calculated as:

$$\phi^+(S_x) = \sum_{x \in S} \prod (S_x, S_y)$$

Negative outranking flow is calculated as:

$$\phi^-(S_x) = \frac{1}{n\text{-}1} \sum_{x \in S} \prod (S_x, S_y)$$

Step 6: Compute net out ranking flow:

$$\phi(Sa) = \{\phi^+(S_x) - \phi^-(S_x)\}$$

Step 7: Select the top ranked service: The service which has highest net outranking flow is the best service which has been selected by user's preferences and requirement

Step 8: Aggregated Method: The services are being used by the consumer and the feedback based on their satisfaction is taken and aggregated method is used and the services are rated and stored in database

RESULTS AND DISCUSSION

The services are created using Sun Java application Server (Graham *et al*., 2004). The services S1, S2, S3 are created. The access time and dependability are calculated and the services are discovered as below using the traditional SS Com (Radhakrishnan *et al*., 2004) Method. The services access time, availability and dependability were measured and the priority of services were assigned as in Table 3. Weight assigned for each metric as per user requirement is given in Table 4.

Function for each user requirement based on pair wise comparison is calculated. Aggregated preference indexes $\pi(S_x, S_y)$. Positive outranking flow, negative outranking flow and net flow using the equation:

$$\phi^+(S_x) = \sum_{x\in S}\prod(S_x, S_y)$$

Table 3: SSCOM results

S-Id	S-Name	Access time	Avail	Dependability	Priority
1	S1	10.7	0.90	0.5275860	1.80
2	S2	5.8	0.36	0.1454500	1.70
3	S3	9.4	0.70	0.0328236	1.36

Table 4: Weight assigned

QoS	Minimum dependability (%)	Maximum avaliability (%)	Minimum access time (nsec)
Weight	50.0000000	30.00	20.0
Service1	0.5275860	0.90	10.7
Service2	0.1454500	0.36	5.8
Service	0.0328236	0.70	9.4

Table 5: Phi values

Services	ϕ^+	ϕ^-	ϕ
1	0.00	0.13	-0.13
2	0.20	0.00	0.20
3	0.03	0.10	-0.07

Table 6: Customer feedback

No. of access	1st choice	2nd choice	3rd choice
30	S1	S2	S1
25	S1	S1	S3
10	S3	S2	S2

Table 7: Feedback points

No. of access	1st choice (3 points)	2nd choice (2 points)	3rd choice (1 point)
30	S1:90	S2:60	S1:30
25	S1:75	S1:50	S3:25
10	S3:30	S3:20	S2:10

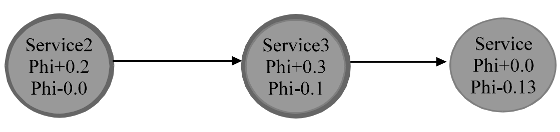

Fig. 3: Netflow

$$\phi^-(S_x) = \frac{1}{n\text{-}1}\sum_{x\in S}\prod(S_x, S_y)$$

$$\phi(Sa) = \{\phi^+(S_x) - \phi^-(S_x)\}$$

is given in Table 5. The service which has highest net outranking flow is the best service which has been selected by user's preferences and requirement (Fig. 3). The services are being used by the consumer and the feedback based on their satisfaction is taken and aggregated method is used and the services are rated and stored in database.

Then, feedback from the client is collected and from the rating given by users the service ranking database is updated. The services ranked are returned to the consumers. Using the Aggregated Method the rank is updated in database in the Table 6 and 7. The aggregated points are calculated as:

Service 1 == 90+75+50+30 = 245

Service 2 = =60+20+10 = 90

Service 3 = =30+25 = 55

Since, the service1 has the highest points it is considered the best service. Therefore, based on the customers feedback too the services are being rated and thus this architecture seems to be the suitable one.

CONCLUSION

Today world is moving towards green environment. The technological improvements has facilitated the human to lead a sophisticated life. The services now are available in cloud and people wishing to avail the service gets it without needing the infrastructure. The base for this is the service oriented architecture. The architecture uses a mathematical model to rank the services .The QoS factors as availability, reliability, access time were considered. The ranked services were further refined using Aggregated Method based on consumers feedback. The experimental results shows improved web service selection as per users preferences

REFERENCES

Blum, A. and F. Carter, 2004. Representing web service management information. January 2004. http://www.oasis-open.org/committees/download.php/5144.

Cao, L., J. Cao and M. Li, 2005. Genetic Algorithm Utilized in Cost-Reduction Driven Web Service Selection. In: Computational Intelligence and Security, Hao, Y., J. Liu, Y.P. Wang, Y.M. Cheung and H. Yin *et al.* (Eds.). Springer-Verlag, Berlin, Heidelberg, pp: 679-686.

Chen, H., T. Yu and K.J. Lin, 2003. QCWS: An implementation of QoS-capable multimedia web services. Proceedings of the 5th International Symposium on Multimedia Software Engineering, December 10-12, 2003, Taichung, Taiwan, pp: 38-45.

Graham, S., D. Davis, S. Simeonov, G. Daniels and P. Brittenham *et al.*, 2004. Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI. 2nd Edn., Pearson Education, USA.

Kritikos, K. and D. Plexousakis, 2009. Requirements for QoS-based web service description and discovery. IEEE Tran. Services Comput., 2: 320-337.

Liu, S.L., Y.X. Liu, N. Jing, G.F. Tang and Y. Tang, 2005. A dynamic web service selection strategy with QoS global optimization based on multi-objective genetic algorithm. Proceedings of the 4th International Conference on Grid and Cooperative Computing, November 30-December 3, 2005, Springer-Verlag Berlin, Heidelberg, pp: 84-89.

Liu, Y., A.H. Ngu and L.Z. Zeng, 2004. QoS computation and policing in dynamic web service selection. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters, May 17-22, 2004, New York, USA., pp: 66-73.

Liu, Z.Z., Z.J. Wang, X.F. Zhou, Y.S. Lou and L. Shang, 2012. A new algorithm for QoS-aware composite web services selection. Proceedings of the 2nd International Workshop on Intelligent Systems and Applications, May 22-23, 2012, Wuhan, pp: 1-4.

Menasce, D.A., 2002. QoS issues in web services. IEEE Internet Comput., 6: 72-75.

Radhakrishnan, T., Kirubhakaran and J. William, 2004. Analysis and enhancement of service discovery framework. Res. J. Applied Sci. Eng. Technol., 7: 807-811.

Ran, S.P., 2003. A model for web service discovery with QoS. Newslett. ACM. SIGecom Exchanges, 4: 1-10.

Tao Yu, T., Y. Zhang and K.J. Lin, 2007. Efficient algorithms for web services election with end-to-end qos constraints. ACM Trans. Web, Vol. 1. 10.1145/1232722.1232728.

Xu, Z., P. Martin, W. Powley and F. Zulkernine, 2007. Reputation-enhanced QoS-based web ervices discovery. Proceedings of the IEEE International Conference on Web Services, July 9-13, 2007, Salt Lake City, UT., pp: 249-256.

Zhu, Z.D., X.H. Dong, Y.H. Hu and Z.Z. Li, 2008. Selection mechanism of composite web services based on usability. Proceedings of the International Conference on Computer Science and Software Engineering, Volume 2, December 12-14, 2008, Wuhan, Hubei, pp: 372-376.

Zou, G., Y. Xiang, Y. Gan, D. Wang and Z.B. Liu, 2009. An agent-based web service selection and ranking framework with QoS. Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, August 8-11, 2009, Beijing, pp: 37-42.