Cost Dependent QoS-based Discovery of Web Services

Dessislava Petrova-Antonova

Sofia University, Faculty of Mathematics and Informatics
125 Tzarigradsko shose Blvd., 1113 Sofia, Bulgaria
d.petrova@fmi.uni-sofia.bg

Abstract. The role of the web services for development of distributed applications steadily increases over time. However, the rising number of available web services with the same functionality embarrasses clients during selection of such a one that fits best their requirements. To solve this problem, web service selection process needs to concern not only functional but also nonfunctional (QoS) properties of web services. The clients need to know the quality of the offered web services as well as what will be the price that they should pay for that quality. This paper contributes to this challenge by presenting an algorithm that allows clients to select the web service with an optimal correlation between quality and price.

Keywords: QoS, web service discovery

1 Introduction

The role of the web services for development of distributed applications steadily increases over time. However, the rising number of available web services with the same functionality embarrasses clients during selection of such a one that fits best their requirements. To solve this problem, web service selection process need to concern not only functional but also nonfunctional (QoS) properties of web services. The current web service architecture is based on Web Service Description Language (WSDL) and Universal Description Discovery and Integration (UDDI) standards that support only functional web service description, publication and discovery. That is why many efforts point to developing of QoS models and ontologies as well as QoS enhanced repositories and selection algorithms.

The clients need to know not only the quality of the offered web services, but also what will be the price that they should pay for that quality. The expression "you get what you pay for" is widespread, but it's not always true [10]. This motivates us to continue our previous work [14] by designing an algorithm that will help clients to select the web service with an optimal correlation between quality and price. We rely on Justin McHenry's value equation: Quality/ Price=Value. He suggests number 1 as a base, and anything above one is good value, anything below is bad value. The equations High quality/ High price = 1, Low quality/ Low price = 1 and Medium quality/ Medium price = 1 mean that the client got what he/she paid for. But what the client is looking for are the "1+" products and services, where the quality beats the price.

The remainder of this paper is structured as follows. Section 2 introduces related work. Section 3 describes the proposed web service selection algorithm. Section 4 shows experimental results that prove the algorithm. Finally, Section 5 concludes the paper with contributions and future work.

2 Related Work

Web service selection brings a challenge to the clients due to the fact that web services with the same functionality have different QoS properties. This section presents a review of various approaches that meet this challenge.

Zhang et al. [23] extend web ontology language for services (OWL-S) with QoS descriptions. The new ontology, called OWL–QoS, is a base for service discovery in the Universal Network. The authors propose P2P-based semantic service discovery with OWL–QoS and a matching algorithm between advertisements and requests described in OWL–QoS. The OWL-S is also used by Tondello and Siqueira [18] to design a QoS-enabled search tool for web services. The tool executes SPARQL queries on the ontology containing QoS description of web services.

FUSION Semantic Registry [7] is an extension of the UDDI registry, which stores semantically annotated descriptions of service interfaces, defined with SAWSDL. OWL is used for modeling service characteristics and performing fine grained service matchmaking via DL reasoning. It is also applied by Day [5] in order to describe QoS information, which then is analyzed by an expert system written in JESS. Zhou et al. [24] introduce the QoS matchmaking algorithm with multiple matching degrees based on DAML-QoS ontology, which is designed as complement ontology to provide additional QoS information for DAML-S. Keskes et al. [6] define a model for an automatic selection of best service provider that is based on mixed context and QoS ontology for a given set of QoS properties. Chaari et al. [3] apply ontological concepts to WS-Policy in order to enable semantic matching.

Pilioura et al. [13] propose a framework for unified publication and discovery of semantically enhanced services over heterogeneous registries named PYRAMID-S. The framework uses an extension of WSDL (PS-WSDL) for describing web services and USQL for service discovery.

Reddy D et al. [15] apply a Greedy approach to evaluate the QoS properties and estimate the criteria rank for each of them. Tran et al. [19] adopt Analytic Hierarchy Process (AHP) method as an underlying mechanism for QoS-based web service ranking.

Xu et al. [22] extend UDDI registry with QoS information. They propose a reputation manager that collects feedback regarding the QoS of the web services from the service consumers, calculates reputation scores, and updates these scores in the Rating database. The discovery agent The QoS scores are calculated based on dominant QoS attribute, which is specified by the consumer to be the most important in the search criteria, and the reputation scores.

Ahmadi and Binder [1] propose a language which lets the client define the semantics of matching by defining the repositories for service lookup, the non-functional properties for service selection, and a utility function for service ranking. The client uses an XPath expression for each type of service specification and is provided with an interface for interoperating with UDDI and third-party repositories.

Al-Masri and Mahmoud [2] define Web Service Relevancy Function (WsRF) in order to measure the relevancy ranking of a particular web service. Firstly, the maximum normalized value for each set of QoS parameters is calculated. Secondly, each QoS measured value is compared against the maximum in its corresponding set. Finally, a web service QoS manager computes WsRF values for all available services related to client request.

Pathak et al. [12] propose approach that relies on user-supplied, context-specific mappings from user ontology to relevant domain web service ontologies. It allows client to define QoS properties valuable for his/her purpose. The framework provides the notion of ranking attributes and a function for ranking the candidate web services.

Vu et al. [20] present a web service selection and ranking approach that uses trust and reputation evaluation techniques to predict the future QoS. The prediction is based on user reports on QoS of all services over time and the quality promised by the service providers. The user feedback is also used by Chuckmol et al. [4] who propose a collaborative tagging-based environment for web service discovery, allowing users to tag or annotate a Web service using keyword or free-text.

Wang et al. [21] define a web service selection algorithm that uses a QoS matrix to represent the values of QoS properties for candidate web services. These values are scaled into a range of [0, 1] based on uniformity analysis and multiplied with a weight value specified for each QoS property. The QoS evaluation of each web service is computed by summing the recalculated values of each QoS matrix's row. Similar algorithm is proposed by Taher et al. [16]. It calculates the final QoS evaluation of each web service using Euclidian distance to find the distance between the specified values of QoS properties by the client and the values in QoS matrix. In contrast of these authors, Liu et al. [8] perform two normalizations of QoS matrix depending on how the quality criteria are grouped. The purpose of normalization is to represent each of QoS property value by uniform index that can be 0 or 1.

The QoS selection algorithm proposed by Maximilien et al. [10] is based on trust function that uses the collected quality values while taking into account the quality preferences of the client and the advertisements of the provider.

Samper et al. [17] define matchmaking algorithm that allows finding services based on their similarities. Semantic values are used to describe each one of the service capabilities. Makris et al. [9] propose an adaptive algorithm performing selection among similar web services located at different infrastructures. The best candidate web services is selected based on network latency between the client and the web service, and the number of other distinct web services, functionally related to the web service in terms of business environment. The adaptive selection is performed using online QoS ratings and the availability conditions of the infrastructure.

In summary, we can conclude that there are various web service discovery algorithms concerning QoS, but there is no approach that selects web services searching for optimal correlation between the quality and cost.

3 QoS-Enhanced Selection Algorithm

In this section, we propose a web service selection algorithm searching for optimal correlation between the quality and cost. Once a client retrieves from UDDI registry a list of web services that fit his/her functional requirements, he/she needs to compare them according to their QoS properties.

We consider measurable QoS properties, which can be divided in two groups: lower bound (LB) like Response time and upper bound (UB) like Availability. This division is made because the clients are interested in low values for some QoS properties and high values for another. For example, from the client's perspective, the value of Response time need to be as low as possible and the value of Availability need to be as high as possible.

For each candidate web service the algorithm computes QoS value, which gives the web service quality in relation to its Cost. The algorithm is based on Justin McHenry's value equation, described in Section 1. In order to calculate high QoS value, it requires lower values for the lower bound QoS properties and higher values for the upper bound QoS properties. The Cost's value has to be as low as possible.

We assume that there is a set of web services that have the same functional properties, defined as follows:

$S = \{S_1, S_2, \ldots S_i, \ldots S_n\}$, where $n\ (1 \le i \le n)$ is a number of candidate web services.

If the client has requirement for the maximum cost of the web service, the algorithm filters the list of candidate web services according to that requirement. Our goal is to order these services according to their QoS value (Q_{S_i}).

We suppose that the values of QoS properties for each web service are collected through monitoring and stored into QoS database. That is why on the first step our algorithm needs to calculate average values of QoS properties for each web service. Thus, we define a set of values for QoS properties for candidate web services as follows:

$P_{S_i} = \{P_{S_i 1}, P_{S_i 2}, \ldots, P_{S_i j}, \ldots P_{S_i m}\}$ where $m\ (1 \le j \le m)$ is a number of QoS properties, which are of interest for the client.

Since each QoS value in the set P_{S_i} has different measure and the values vary within a large range, the set P_{S_i} needs to be normalized. Before normalizing the set P_{S_i}, we need to calculate the maximum and minimum value for each QoS property among web services, respectively $P_{j\min}$ and $P_{j\max}$. Thus each element in the set P_{S_i} will be normalized using the following equation:

$$P_{S_i j, norm} = \begin{cases} \dfrac{P_{S_i j} - P_{j\min}}{P_{j\max} - P_{j\min}}, if\ UB\ \ QoS\ \ property \\ 1 - \dfrac{P_{S_i j} - P_{j\min}}{P_{j\max} - P_{j\min}}, if\ LB\ \ QoS\ \ property \end{cases} \qquad \textbf{(1)}$$

After normalization the values of QoS properties will be presented in the range of [0, 1].

QoS value for each web service then can be calculated as follows:

$$Q_{S_i} = \frac{\frac{1}{m}\sum_{j=1}^{m} P_{S_i j, norm}}{C_{S_i norm}} \qquad \textbf{(2)}$$

where $C_{S_i norm}$ is a normalized Cost property of the web service S_i. Its value is calculated according to the following equation:

$$C_{S_i, norm} = \frac{C_{S_i} - C_{\min}}{C_{\max} - C_{\min}} + 1 \qquad \textbf{(3)}$$

where $C_{\max}$ and $C_{\min}$ are, respectively, the maximum and minimum Costs among all web services' Costs.

The QoS values of all web services form a set $Q = \{Q_{S_1}, Q_{S_2}, \ldots Q_{S_n}\}$.

Finally, the algorithm sort the set Q in descending order, so the web service with the highest quality will be shifted at the beginning of the set and the web service with the lowest quality will be shifted at the end of the set.

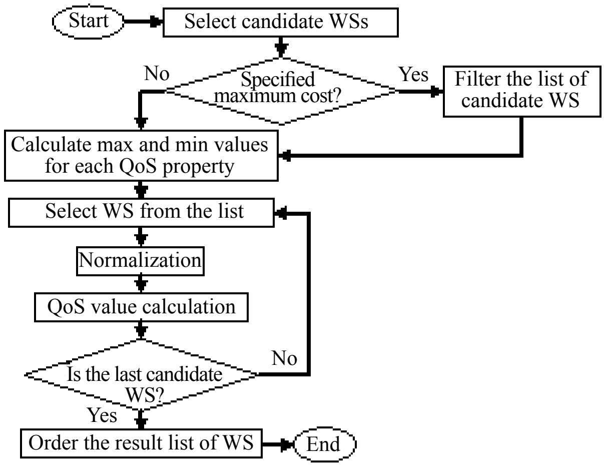

Fig. 1. Block diagram of the algorithm.

A block diagram of the algorithm is shown in Fig. 1.

4 Experimental Results

In order to verify the proposed algorithm, we used experimental data, presented in [2]. The data are retrieved during execution of email verification web services that are listed in XMethods.net, XMLLogic, and StrikeIron. Table 1 shows measurements of six QoS properties concerning these services: Response Time (RT), Throughput (T), Availability (AV), Accessibility (AC), Interoperability Analysis (IA), and Cost (C).

Table 1. Experimental data, Al-Masri and Mahmoud (2007).

Service Provider	RT	T	AV	AC	IA	C
XMLLogic	720	6.00	85	87	80	1.2
XWebServices	1100	1.74	81	79	100	1
StrikeIron	710	12.00	98	96	100	1
StrikeIron	912	10.00	96	94	100	7
CDYNE	910	11.00	90	91	70	2
Webservicex	1232	4.00	87	83	90	0
ServiceObjects	391	9.00	99	99	90	5

The data presented in Table 1 are stored in a local database on MS SQL Server 2005. The algorithm is implemented with Visual Basic.NET and integrated in a web application that reports the results from its execution. Table 2 shows these results. From the table, we can see that the web service provided by Webservicex is free, but it is next to the last in the result list of web services. This is due to the fact that Webservicex's web service has too high Response time in comparison with other web services.

Table 2. Ranking list of web services.

Service Provider	QoS	Cost
StrikeIron	0,772	1
ServiceObjects	0,510	5
StrikeIron	0,376	7
CDYNE	0,371	2
XMLLogic	0,340	1.2
Webservicex	0,284	0
XWebServices	0,202	1

Fig. 2 presents the relationship between the computed QoS value of each web service and it's Cost.

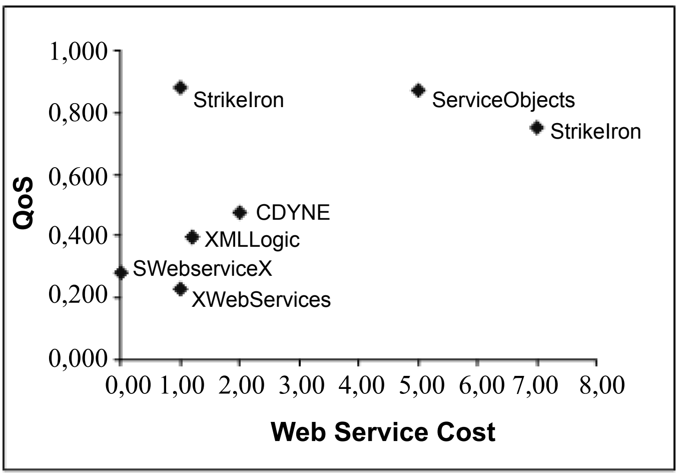

Fig. 2. QoS/Cost relationship.

5 Conclusion

This paper contributes to the challenge of web service discovery based on QoS properties by presenting an algorithm that allows clients to select the web service with an optimal correlation between quality and price. The algorithm is proved through experimental results on data retrieved during execution of email verification web services.

Future work includes the execution of additional tests in order to evaluate the algorithm in different scenarios. We plan to integrate the algorithm in a QoS Discovery Agent that is a part from a QoS-enhanced framework for web service publication and discovery. The framework is in process of research and this algorithm is a first step of its implementation.

Acknowledgements. This paper has been supported by the Project Creative Development Support of Doctoral Students, Post-Doctoral and Young Researches in the Field of Computer Science (No. BG 051PO001-3.3.04/13) of the HR Development OP of the European Social Fund 2007-2013.

References

1. Ahmadi, N., Binder, W.: Flexible Matching and Ranking of Web Service Advertisements. In: MW4SOC'07, Newport Beach, CA, USA, pp. 30-35 (2007)
2. Al-Masri, E., Mahmoud, Q.: Discovering the Best Web Service. In: WWW 2007, Banff, Alberta, Canada, pp. 1257-2589 (2007)
3. Chaari, S., Badr Y., Biennier, F.: Enhancing Web Service Selection by QoS-Based Ontology and WS-Policy. In: SAC'08, March 16-20, Fortaleza, Ceará, Brazil, pp. 2426-2431 (2008)
4. Chukmol, U., Benharkat, N., Amghar, Y.: Enhancing Web Service Discovery by using Collaborative Tagging System. In: 4th IEEE International Conference on Next Generation Web Services Practices, 20th - 22nd October, Seoul, South Korea, pp. 54-59 (2008)
5. Day, J.: Selecting the best web service. In: CASCON'04, pp. 293 – 307 (2004)
6. Keskes, N., Lehireche, A., Rahmoun, A.: Web Services Selection Based on Context Ontology and Quality of Services. In: International Arab Journal of e-Technology, Vol. 1, No. 3, pp. 98-105 (2010)
7. Kourtesis D., Paraskakis, I.: Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery. In: ESWC'08, The Semantic Web: Research and Applications, vol. 5021, Springer, Heidelberg, pp. 614-628 (2008)
8. Liu, Y., Ngu, A. H. H., Zeng, L.: QoS Computation and policing in dynamic web service selection, In: WWW 2004, May, pp. 66-73 (2004)
9. Makris, Ch., Panagis, Y., Sakkopoulos, E., Tsakalidis, A.: Efficient and adaptive discovery techniques of web services handling large data sets. In: Journal of Systems and Software, Volume 79, Issue 4, pp. 480 – 495, ISSN: 0164-1212 (2006)
10. Maximilien E., Singh, M. P.: Multi agent System for Dynamic Web Services Selection. In: IEEE Internet Computing, vol. 8, Issue 5, pp. 84-93 (2004)
11. McHenry, J., The Price/Quality/Value Equation, http://www.finance-weblog.com/50226711/ the_pricequalityvalue_equation.php
12. Pathak, J., Koul, N., Caragea, D., Honavar, V.: A Framework for Semantic Web Services Discovery. In: ACM 7th Intl. workshop on Web Information and Data Management, at 14th ACM Intl. Conference on Information and Knowledge Management (2005)
13. Pilioura, T., Tsalgatidou, A.: Unified Publication and Discovery of Semantic Web Services. In: ACM Transactions on The Web, Vol. 3, No. 3, Article 11, pp. 11:1-11:44 (2009)
14. Petrova-Antonova D., S. Ilieva: Towards Unifying View of QoS-EnhancedWeb Service Description and Discovery Approaches. In: European Yong Researchers Workshop on Service Oriented Computing (YR-SOC), Pisa, Italy, pp. 99-113 (2009)
15. Reddy, D K., Thirumaran, M., Maralla, K., Kumar G, R.: A Greedy Approach with Criteria Factors for QoS based Web Service Discovery. (2009)

16. Taher, L., El Khatib, H., Basha, R.: A Framework and QoS Matchmaking Algorithm for Dynamic Web Services Selection. In: The Second International Conference on Innovations in Information Technology (IIT'05), Dubai, UAE, Sept 26 - 28, (2005)
17. Samper, J. Javier, Adell, F. Javier, Van de Berg, L., Martinez J. Jose: Improving semantic web service discovery. In: Journal of Networks, Vol. 3, No. 1, January, pp.35-42 (2008)
18. Tondello, G., Siqueira F.: The QoS-MO Ontology for Semantic QoS Modeling, In: SAC'08, March 16-20, Fortaleza, Ceará, Brazil, pp. 2336-2340 (2008)
19. Tran, V.X., Tsuji, H., Masuda, R.: A new QoS ontology and its QoS-based ranking algorithm for Web services. In: Simulation Modelling Practice and Theory, pp. 1378-1398 (2009)
20. Vu, L., Hauswirth, M., Porto, F, Aberer, K.: A Search Engine for QoS-enabled Discovery of Semantic Web Services. In: International Journal of Business Process Integration and Management, Volume 1, No. 4, pages 244-255 (2006)
21. Wang, X., Vitvar, T., Kerrigan, M., Toma, I.: A QoS-Aware Selection Model for Semantic Web Services. In: Proceedings of ICSOC'06, pp. 390-401 (2006)
22. Xu, Z., Martin, P., Powley, W., Zulkernine, F.: Reputation-Enhanced QoS-based Web Services Discovery, In: ICWS, pp. 249–256 (2007)
23. Zhang, Y., Huang, H., Yang, D., Zhang, H., Chao, H., Huang Y.: Bring QoS to P2P-based semantic service discovery for the Universal Network. In: Springer-Verlag London Limited, pp. 471-477 (2009)
24. Zhou, Ch., Chia, L., Bilhanan, S, Lee, B.: UX - An Architecture Providing QoS-Aware and Federated Support for UDDI, In: ICWS'03, pp. 472-479 (2003)