Measuring the qualitative feature of combined services

Aalia Hemmati, Sima Emadi

Computer Engineering department, Islamic Azad university of Meybod, Yazd, Iran

Email address:

Aaliahemmati64@gmail.com(A. Hemmati), au_emadi@yahoo.com (S. Emadi)

Abstract: The combination of web services is the result of complex and increasing needs of the users and disability of single web services in resolving the user's needs. One of the important challenges in the field of web 2.0 is the combination of web services based on their qualitative features. Since it is probable that there would be several different combinations of services for achieving a specific goal, choosing the service is based on some qualitative features like combining, availability, acceptability, service cost and security. One of the important issues is the quantitative survey of combining rate of the two services shared on the combination so that they have the ability to combine with each other, correctly. In order to measure the combining ability of services, in the first stage, the more number of effective factors on combining features of services are surveyed in comparison with the present methods. In the second stage, metric is introduced for the effective factors, and in the third stage, an appropriate weight for each factor is found and finally, based on their relationships with each other, a more accurate rate of combining is obtained.

Keywords: Web Service, Qualitative Feature of Service, Combining, Metric

1. Introduction

Service-oriented systems have special importance because of the possibility of working in heterogeneous distributed environments. The users of such systems use the web services that provide system components. In some cases, the needs of the users are not met with the single web service. However, each combination of services is not always the best possible solution and some criteria and features of the new combined service such as the quality of the new service can be a criterion for choosing the appropriate service. In fact, one of the important challenges in the field of web services is the combination of web services considering their quality. The need to combine web services is the result of complex and increasing requirements of users and disability of web services in responding to the goals of the users. Since it is likely that there would be several different combinations of services in achieving to a specific goal, choosing the service is based on some qualitative features like combining, availability, acceptability, service cost and security. One of the important issues that has been noted less is the accuracy quantitative survey of combination rate of the two services shared on the combined platform.

This feature brings the ability of a service for combining with other services in the runtime so that the combinations work out successfully. In fact, the rate of services' adaptation with each other for combination without facing problems in the runtime is a desired goal. After that the user applies for the required service, the component identifies the basic services with the combination of which it can get to the desired service and search for them inside the service store. Because of the diversity of different producers and services, it's likely that several candidate services be found for each desired request.

Considering the quality rate of combination that the candidate services have, the optimum service is selected. Combining two services that are combined in series is affected by some factors such as similarities of input and output, observing the preconditions, reusability, adaptability, loose coupling, etc. In [1], a framework is presented for choosing the service in terms of qualitative features in which the combination rate of services is focused. In [9], the combining feature is surveyed absolutely, in the way that the two services are combining with each other or not.

In this article, by surveying the available weaknesses in the presented method in [9], by surveying the more number of effective factors and the importance level of them and presenting a metric for each, a more accurate rate would be obtained for the combining feature. Some existing methods

have more focused on similarities of services' input and output for surveying the combining rate of the two services and in cases that the other effective factors on combining is surveyed, no metric has been presented for its measuring. This article has evaluated various number of effective factors on combining feature such as the reusability and adaptability in comparison with other methods and finally a more accurate rate of combining has been obtained for each service. The structure of the article is in this way that in the second section, the effective factors on combining feature of services would be surveyed and metrics would be introduced for each. In the third section, based on the obtained values of the effective factors, the combining rate would be measured. In the fourth section, conclusion and future activities would be discussed.

2. Effective Factors on Combining Feature

In this study, web service is considered as a black box and there is access only to its interface. When the service is implemented as a web service, WSDL descriptions are the most common documents of service description. Since the available web services use WSDL, a semantic description in OWL-S language. OWL-S description of service comes in three part of service profile, service model and service support [9].

The service profile part includes input, output, name and service description and service support which includes transfer protocol and web service address are considered in our study. Also, it is supposed that services based on SOAP protocol communicate with each other [9].

Effective factors on combining are in two groups. One of them are the factors that are obtained from the communication between the desired candidate service with the previous existing service of the combination, like the surveying parameters of the similarity level of input and output parameters, effects and preconditions and the other are the factors surveyed on candidate atomic service. Table 1 shows the effective factors on combining with this condition that the importance level and the two qualitative parameters of reusability and adaptability are added, in order that a more accurate rate of combining is obtained. Furthermore, the effective factors on combining feature would be surveyed.

Table 1. *Effective factors on combining, adopted from [9]*

Effective Factors on Combining in Atomic Terms _ Importance level	Effective Factors on Combining in Combining Services' Aspect
Granularity	The similarity level of input- output
Reusability _ medium	The observing level of preconditions and effects
Adaptability_ high	The similarity level of input- output
Availability	The observing level of preconditions and effects
Well-defined interfaces_ high	The similarity level of input- output
Loosely-coupled _ medium	The observing level of preconditions and effects
Surveying the quality level of service	

2.1. Surveying the Similarity Level of Input and Output Parameters

When two services are combined with each other , the output of the first service is usually considered as the inputs of the second service. The input and output show the message interactions in services. Here, the two services have been showed as S_1 and S_2. The major purpose is surveying the similarity level of input and output parameters in the two services, considering that the m input s_2 corresponds with the n output s_1. Equation 1 has been defined for surveying the similarity level of input and output type in the two services [9]:

$$\lambda 1(S1,S2)=\sum_{i=1}^{n}\frac{P_i}{m} \qquad (1)$$

Equation 1: The similarity level of input- output parameters

$$p_{i=}\begin{cases} 1, \text{if Type}\big(\text{Output}(s1.i)\big) = \text{Type}(\text{Input}(s2.i))) \\ \text{or Type}\big(\text{Output}(s1.i)\big) \subset \text{Type}(\text{Input}(s2.i)) \\ \epsilon, \text{Type}\big(\text{Input}(s2.i)\big) \subset \text{Type}(\text{Output}(s1.i)) \\ 0, \text{otherwise} \end{cases} \qquad (2)$$

Equation 2: input- output parameters
If 's_2' service input is exactly equal to s_1 service output or 's_1' service output is a subset of 's_2' service inputs, the amount 1 would be returned as result. But when 's_1' service outputs is a subset of s_2 service input, it is probable that some problems would be created in runtime for compatibility between two services.

2.2. Similarity Level of Preconditions and Effects

Preconditions show logical conditions that must be supplied before replying to the service. Effects are events that happen by the successful performance of a service. For example, "having more than 1000 dollar credits" is a precondition for the service of buying a book and "I'm the owner of a book" is the effect of performing this service.

According to relation 3, for measuring the effects and preconditions, suppose that k component is proposed as precondition in the second service that can be covered by the first service as its effects. According to Equation 4, the similarity level of preconditions and effects would be measured and also the number of all preconditions of the second service is equal to m [9].

$$Ki = \begin{cases} 1, if\ PC(S2) \cap ET(S1) \neq \emptyset \\ 0, otherwise \end{cases} \qquad (3)$$

Equation 3: preconditions and effects

$$\lambda 2(S1,S2) = \begin{cases} if\,PC(S_2) \neq \varnothing\,,\ \sum_{m}^{i=1} Ki/m \\ if\,PC(S_2) = \varnothing,\,1 \end{cases} \quad (4)$$

Equation 4: Similarity level of preconditions and effects

2.3. Service Granularity

One of the measuring criteria of service granularity is service size and the other is the number of operations that service operates. The number of operations is shown by the word "operation" in WSDL file and "process" in OWLS file. The measuring criterion of this feature is stated as Equation 5 [5]:

$$\lambda_3(S_1,S_2) = 1/\text{number of "atomic process" in owls file} \quad (5)$$

Equation 5: Service granularity

In [5], a metric is presented for computing granularity rate which measures the level in which a service is independent of other services. Equation 6 measures this metric.

$$\lambda_3(S_1,S_2) = 1-(\text{Num}_{SRVOpWithDependency}/\text{Num}_{TotalSRVQp}) \quad (6)$$

Equation 6: Service granularity

That in relation 6, $\text{Num}_{SRVOpWithDependency}$ is the number of operations dependent to other services and $\text{Num}_{TotalSRVOp}$ is the whole number of service operations and their amount interval is between 0 to 1.

2.4. Availability

This feature measures the availability rate of a service when the user applies for it. The reason why we use this feature is that the service availability is the first condition of a successful performance of a service. So, if only one service of the combination is not available, the whole combination would be considered unavailable [2].

Availability is defined as a level of a system or its component's being operational and available exactly when the user needs it. Services' availability is a concern for the success of service-oriented architecture from the aspect of both the user and the producer. In user's viewpoint, availability means that a service set becomes available for doing the functional needs of a system. Now, if one of these services become unavailable (even temporarily), it is because of the disorder success of service-oriented systems. From the service producer's aspect, the services must be available when needed, otherwise, the producer's capital and popularity would be affected. (Especially when the service is unavailable, he must pay the damages). This metric is computed by the Equation 7 [2]:

$$\lambda_4(S_1,S_2) =WSOT/(WSOT+WSRT) \quad (7)$$

Equation 7: Availability

Where WSOT is the time of service availability and WSRT is the time of repairing the system. We can also obtain the availability from the treaty file of the service level. Since in the existing format, no independent place is forecasted for index and definition of all variables, it's necessary that all the variables be defined immediately after its proposal in the article.

2.5. Loosely-Coupled Factor

The word coupling is a part of the information technology's dictionary. Whatever that communicates, has the coupling and whatever that has the coupling can be dependent to each other [8]. One of the common methods of defining the coupling is comparing it to the dependence. The level of coupling between two things is equal to the dependence level that exists between the two. For example, the relationship between a software program with others shows its coupling level or the relationship between the technical contract of the program with the logic of the solution it shows, measures the coupling level [11].

Whatever the coupling between the services be looser, the rate of service intelligibility, reusability and flexibility increases. With increase in service's reusability, the capability of service' combining increases either.

Assume that the metric ISCI be the number of services that are called by the desired service and let SOCI be the number of operations that are called by the service [4], Equation 8 is used for computing the loose coupling:

$$\text{Loose coupling}= ISCI+SICI \quad (8)$$

Equation 8: loosely-coupled

In [8], loose coupling has been defined as a concept that is related to scalability, flexibility and fault tolerance and has stated different types for loose coupling. One of these types is the dependence of data model or the data types in which the services that use the complex data types have the strong coupling and the services that use the simple data types have a loose coupling. In this research, the number of complex data types that have been introduced in interfaces, have a reverse relation with loose coupling. The rate of loose coupling is computed according to Equation 9:

$$\lambda_5(S_1,S_2)= 1/(\alpha_x \text{ Number of Complex data types} + \beta_x \text{ Number of service calls}) \quad (9)$$

Equation 9: loosely-coupled

βو α are the factors the amounts of which are obtained experimentally.

2.6. Reusability

In Equation 10, a formula is presented for reusability according to the rate of qualitative parameters of modularity, the rate of adaptability and universality in profession, ability to detect and alignment with standard SC [5]:

$$\lambda_4(S_1) =BCM*(MD*W_{MD}+AD*W_{AD}+SC*W_{SC}+DC*W_{DC}) \quad (10)$$

Equation 10: Reusability

Metric W is a weight that gives an assessor to each metric and the total weights is equal to 1. Whatever the amount of the weight be more, it shows that the reusability is higher.

In Equation 11, a formula is presented for the reusability feature according to qualitative criteria of loose coupling and

the rate and parameters of granularity [6]:

$$\lambda_6(S_1, S_2) = -0.5 * Coupling + 0.61 * Service\ Granularity + 0.61 * Parameter\ Granularity \quad (11)$$

Equation 11: Reusability

In relation 11, parameters of loose coupling and the rate of granularity and the number of granularity parameters are measured based on the number of used messages, the number of simultaneous and asynchronous operations.

2.7. Adaptability

In [5], the adaptability of a service is measured by the internal adaptability. The internal adaptability would be measured if the internal variable of service can be adapted to the user's needs well. Assume 'n' be the number of change points in a service. For each change point, it's probable that the difference the service user expects be provided or not. If the default amount of a change point meet the expectations of the user, it is considered as a present change point. As it goes on, Equation 12 computes the adaptability of this case that how many change point can be adapted as the needs of the users:

$$\lambda_7(S_1, S_2) = Num_{Consumers\ Satisfied\ Variants} / Num_{TotalApplicableConsumer} \quad (12)$$

Equation 12: Adaptability

The numerator is the number of the users that the default change points meet their needs and the denominator is the number of all the users that are dependent to change points. The computed amount is in intervals of 0 and 1 and whatever the amount is more, the adaptability is higher as well.

2.8. Well-Defined Interfaces

When the service is implemented, it has the documents of service description. The service can have a document in addition to the mentioned documents that is readable by human, like the treaty file of service that includes the additional descriptions about the qualitative capabilities, limitations and behavior of the service. Although the service contract is well-defined, it makes the user have a more accurate and convenient perception of the service and as a result, using the service will be easier. Since the services would be at the disposal of the users as a black box component, the service contract is the only solution by which you can have an accurate perception of the service [9].

By well-defined interfaces, we mean the surveying of the fields such as conditions, effects, input, output, service categorizing and describing service performance in file WSDL in terms of being available completely. When the semantic web is used, the service interface is OWL-S that again the mentioned cases are proposed for the way of filling the semantic web field and the accuracy of them. The ontology profile determines the following feature for referring to IOPE: (hasParameter, hasInput, hasOutput, hasPrecondition , hasResult)

Naturally, whatever the service contract has a higher standard, using and working with the service is easier and the

success is higher in that combination. This metric is computed by Equation 13 [9]:

$$\lambda_8(S_1, S_2) = \sum_{i=0}^{4} Xi \quad x_i \in \{0, 0.2\} \quad (13)$$

Equation 13: Well-defined interfaces

$$xi = \begin{cases} 0.2, hasparameter \\ 0, no\ parameter \end{cases} \quad (14)$$

$$x_0 = input,\ x_1 = output,\ x_2 = precondition,\ x_3 = result,\ x_4 = category \quad (15)$$

Equation 15: Well-defined interfaces parameters

2.9. Surveying the Quality Level of the Service

Generally, service quality is of great importance. Service applicant states limitations such as replying time, cost and so on for service (As stating limitations for service efficiency). If the quality limitations are not observed, the service is not appropriate for performing. Quality features that are proposed by the applicant are usually determined as a range (by determining the minimum and maximum of the feature amount). In this level of the combining, we can survey that how much of the user's quality features are supplied by the candidate service [9]. This metric is computed by the Equation 16:

$$\lambda 9(S1, S2) = \left(\sum_{Qi \in neg} W_i \frac{Q_i^{max} - Q_i}{Q_i^{max} - Q_i^{min}} + \sum_{Qi \in pos} W_i \frac{Q_i - Q_i^{min}}{Q_i^{max} - Q_i^{min}} \right) \quad (16)$$

Equation 16: quality level of the service

In Equation 16, the amount of Q_i is equal to the measure of quality feature of I th that is presented by the service. Q_i^{max} is equal to maximum amount of quality feature of I that is presented by the user and Q_i^{min} the minimum amount of quality feature of I that is presented by the user.

3. Metric for Measuring the Combining Feature

This metric is presented for measuring the combining rate of the two continuous services that are combined with each other. The input of the issue is a workflow including some duties that are to be performed by the real services. These services are shown as S_1 S_2.

Assume that for the duty x_1 a real service called s_1 has been found. Now, the issue is that among the candidate services for the duty x_2, which real service is better to be selected. The choosing criterion is the service that would have a higher combining capability with the real service s_1. This metric is computed by Equation 18 [9]:

$$Composability(S_1) = \lambda(S_1) \quad (17)$$

Equation 17: Composability

$$\lambda(S1, S2) = \begin{cases} 0, & if \ \prod_i \lambda_i(S_1) = 0 \\ \sum a_i \lambda_i(S_1), & if \ \prod_i \lambda_i(S_1) \neq 0 \end{cases} \quad (18)$$

Equation 18: Composability

In which, I is the number of effective factors, i = {1…9}, ai is the weight allocated to each criteria (between 0 and 1).

Finally, the obtained amount for combining is divided into the number of effective factors in formula so that the obtained number be normal. $\lambda i(S1)$ shows the amount of each mentioned parameters for service combining. $\lambda(S1) \in [0,1]$ is the combining rate of the desired candidate service. In order to normalize the acquired amount for combining, $\lambda(S1)$ is divided into $\sum_{i=1}^{n} ai$.

And finally, from among the candidate services, the service is selected that the combining rate of it be more than the defined threshold for combining. The combining threshold is the minimum amount that a service must have in order to face no problem in runtime when combining with other services. In this research, the combining threshold has determined 0.6 in the normal situation.

4. Discussion, Conclusion and Future Work

In this article, by surveying the available weaknesses in Rokni method [9], the two effective factors of adaptability and reusability were added to the other effective factors on combining feature and a more accurate rate of combining was obtained. Each of these features is effective on combining feature with a weight. Here, the weight of all cases is the same and equal to 1. Finally, by computing the combining metric for a candidate service, you can obtain its rate of combining. While in Rokni method [9], without considering the effective factors of adaptability and reusability, the rate of combining is obtained. Also, in most similar tasks, the time is spent on surveying the only effective factor on combining and no metric is presented for them. In line with this research, it's possible that in future, the effective qualitative parameters of autonomy on the rate of combining be estimated and a metric be presented for it. A more accurate surveying of the allocated weights to each combining parameters is another task that can be surveyed in the future.

References

[1] Q. Yu, M. Rege, A. Bouguettaya,B. Medjahed, M. Quzzani , A Two-Phase Framework For Quality-Aware Web Service Selection, Service Oriented Computing and Applications Journal, Vol 4, No.2, pp. 63-79, 2010.

[2] S. Choi, S. Jin Sun and S. D. Kim, QoS Metrics for Evaluating Services from the Perspective of Service Providers, in IEEE International Conference, e-Business Engineering, ICEBE, pp. 622-625, 2007.

[3] J. Fang, S. Hu and Y. Han, A Service Interoperability Assessment Model for Service Composition, in Proceedings of the 2004 IEEE International Conference on Services Computing, IEEE Computer Society, pp. 153-158, 2004.

[4] R. Sindhgatta, B. Sengupta and K. Ponnalagu, Measuring the Quality of Service Oriented Design, in Proceedings of the 7th International Joint Conference on Service-Oriented Computing, Springer-Verlag: Stockholm, pp. 485-499, 2009.

[5] S. Choi and S. Kim, A Quality Model for Evaluating Reusability of services in SOA, 10th IEEE Conference on E-commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and EServices,2008.

[6] B. Shim, S. Choue, S. Kim and S .Park, A Design Quality Model for Service Oriented Architecture, 15th Asia Pacific Software Engineering Conference, pp.403-410, 2008.

[7] G. Feuerlicht, Design of Composable Services, in Service-Oriented Computing, ICSOC Workshops, 2009.

[8] N.M. josuttis, SOA in practice the Art of distributed system design, oreilly: united states. pp. 35-46, 2007.

[9] Z. Rokni, " choosing the service in service-based architecture based on qualitative features" MA thesis, Shahid Beheshti university, 2011

[10] A. Kazemi, "logical phase application in selection, evaluation and combination of services in service-based architecture", MA seminar, Shahid Beheshti university, 2010.

[11] A. Rostam Pour, "surveying the methods of service evaluation in service-based architecture", MA thesis, Shahid Beheshti university, spring 2010.