WSLA Schema for Functionality Based Weight fixing of Non-Functional Parameters of Web Services

M.Mohemmed Sha, C.Rajalakshmi, S.Pavithra, M.Yasotha Priya
Department of Computer Science
Achariya School of Business and Technology
Pondicherry, India
sahalshas@gmail.com, rajalakshmimbaasbt@gmail.com, pavithragandhi7@gmail.com,
yasothapriya2010@gmail.com

Dr.K.Viveknanandan
Department of Computer Science
Pondicherry Engineering College
Pondicherry, India
kvivek27@yahoo.com

***Abstract*—Recently Web services have evolved as a cost-effective solution for exchanging information between distributed applications over different operating system, platform, and software environment. The success of such a system is mainly depends on the quality offered by the service and the extent to which the web service meets the customer requirements. So while selection of a service the top priority is given to the functionality related aspects. From the list of service with same functionality the selection of a service is made by evaluating the QoS by considering the domain specific and domain independent attribute that leads to the achievement of maximum functionality. But while monitoring the performance of a service, the quality is mainly evaluated by measuring the non functional parameters such as response time, throughput, availability, reliability and sucessiblity. Satisfied level for these non functional values shows that the web service met the customer needs. In this paper we are proposing a method and a schema to assign functionality related weight to the non functional parameters and thereby evaluate the expected level of QoS at the time of selection itself. Apart from the guaranteed level of non functional values asserted in the WSLA, the weights are also assigned to the non functional parameters to evaluate the actual QoS.**

Keywords-QoS,WSLA,

I. Introduction

The success of SOA implementation is realized when appropriate services are discovered based on the functional requirements. After finding a list of functionally-equivalent web services, the most suitable web service is selected. Several semantic Web service discovery algorithms have assumed that the web service functionality is characterized by their inputs and outputs. Semantic matching of inputs and outputs is not sufficient for discovering relevant services. Each web service can provide different quality (non-functional) levels and the selection is about locating the web service that provides the best quality criteria match. Web services in different domains can have different quality metrics, which are called Domain Specific QoS metrics. There can also be some QoS criteria that can be applied to services in all domains irrespective of their functionality or specialty. These criteria are called Domain Independent QoS metrics. Both kinds of QoS metrics need shared semantics for interpreting them as intended by the service provider and requester [4]. This can be achieved by using ontology that defines the domain specific and domain independent QoS metrics [1]. While selection of the web services all the functional domain specific and independent attributes are considered to assure the quality of the web service as effective as possible. The weight of this attributes relative to its domain and the user preferences [2]. This would determine the impact of this attribute on the final decision regarding a provider. During the usage (run time), to check whether the web service reached its actual functionality is mainly based on the non-functional parameter values. So the functionality based weight is fixed for all the non functional parameters for the assertion of expected QoS and is mentioned in the WSLA. The measured metrics from the system instrumentation at runtime can be compared with the asserted values in the WSLA to monitor the actual performance of the web service.

II. FUNCTIONALITY BASED WEIGHT ASSIGNMENT

The QoS selection tree T of the web service S is a tree with nodes of QoS groups or QoS metrics. The functional parametric attributes are represented by the nodes in various levels based on its importance. The main restriction on the tree: in each level l of the tree, the sum of the weights of the nodes that belong to the same parent must equal to one. The root node also has weight equal to 1. Another restriction is that a QoS metric or group cannot appear twice in the tree. A final restriction is that the leaf nodes of T must only be QoS metrics. This tree can be used by the web service requester in order to create groups of QoS metrics that share a common feature [4]. Figure 1 shows the QoS selection tree T that can be reduced to a weighted sum of all the leaf-nodes of the tree.

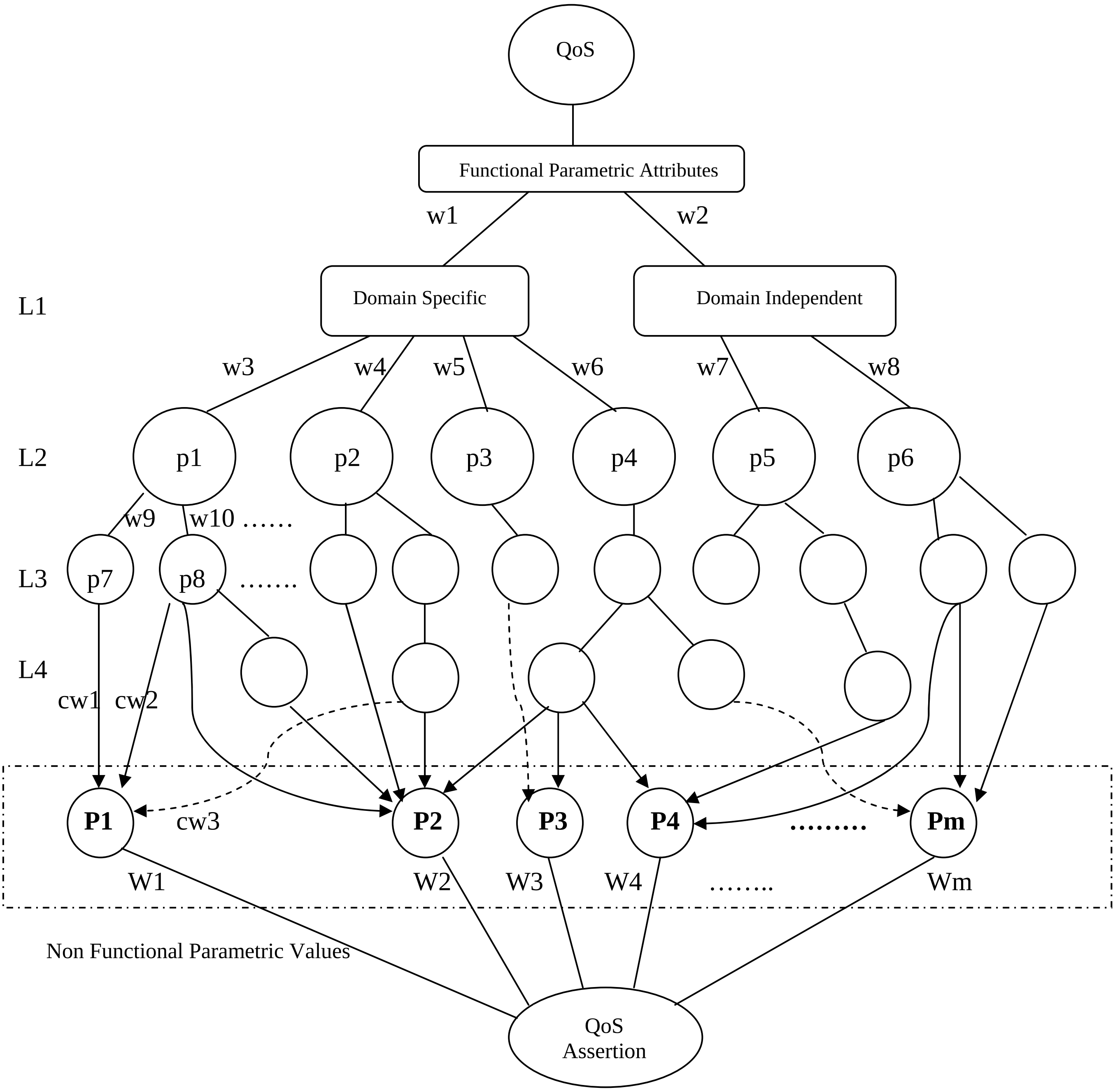

Figure 1: Functionality based weight assignment tree

Let F= {f1, f2, f3 ….fn} are the priority wise functionalities of the web service to reach the customer requirements.

{p1, p2, p3,……..pn} are the functional attributes considered for the achieving the maximum performance of the web service.

{w1,w2,w3,……..wn} are the weights assigned for the functional attributes relative to its domain and the user preferences.

The weight of a particular weight is obtained by multiplying all the individual weight s through its path.

i.e., Weight (p7) = w1 x w3 x w7.

In the same manner weight for the entire leaf node that represent the matrices are calculated. Finally the leaf nodes that have the influence on the non functional parameter to reach the user functionality are mapped with the non functional parametric values. The combined weights of all the mapped matrices are the weight assigned for the non functional parameter.

i.e., W1= cw1+cw2+cw3.

For example

The weights w1=0.6, w2=0.4, w3=0.3, w9=0.3, w10=0.7 are considered as the metrics as per user preference.

The weight of the leaf node cw1=w1 x w3 x w9 = 0.54

The same can be followed to calculate cw2= 0.36, cw3=.61

Here W1=cw1 + cw2 + cw3 = 1.51 is the weight assigned for the non functional parameter P1. In the same manner the weights are assigned for all the non functional parameters P1, P2,…………Pn.

The weights W1,W2, W3…..Wn are normalized to 1to evaluate the QoS and its assertion at the time of selection of a web service.

III. WSLA SCHEMA FOR FUNCTIONALITY BASED WEIGHTS

Web Service-Level Agreement (WSLA) is a standard for unambiguous and clear specification of Service Level Agreements that can be monitored by the service provider and customer [5]. The non-functional attributes can be represented in the form of a Service-Level Agreement (SLA), i.e., a formal agreement or contract between a consumer and a service provider. The agreement is on the level of service that is to be provided in terms of the non functional parametric guarantees and the functional weights of the service [7]. The WSLA definition for the assignment of weight is mentioned in the constant part and the non functional parametric values in the metric section as follows

Type Definition (Constant):

```
<xsd:complexType name="WeightsType">
<xsd:choice>
        <xsd:element name="String" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Integer" type="xsd:integer" minOccurs="0"/>
        <xsd:element name="Float" type="xsd:float" minOccurs="0"/>
</xsd:choice>
<xsd:attribute name="name" type="xsd:string"/>
</xsd:complexType> <xsd:element name="Weight" type="wsla:WeightType"/>
```

Example:

The weights W1=0.79, W2=0.88, W3=0.84, W4=0.93, W5=0.95 are assigned for the non functional parameters response time, throughput, availability, reliability and successibility.

```
<Constant name="ResponseTimeWeight">
        <Float>0.79</Float>
</Constant>
<Constant name="ThroughputWeight">
        <Float>0.88</Float>
</Constant>
<Constant name="AvailabilityWeight">
        <Float>0.84</Float>
</Constant>
<Constant name="ReliabilityWeight">
        <Float>0.93</Float>
</Constant>
<Constant name="SuccessibilityWeight">
        <Float>0.95</Float>
</Constant>
```

Type Definition (Metric):

```
<xsd:complexType name="MetricType">
<xsd:sequence>
        <xsd:element name="Source" type="xsd:string"/>
        <xsd:element name="MetricURI" type="xsd:anyURI" minOccurs="0"/>
        <xsd:choice>
        <xsd:element name="MeasurementDirective" type="wsla:MeasurementDirectiveType"
        minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Function"type="wsla:FunctionType"minOccurs="0" />
        <xsd:element name="MeasurementDirectiveVariable" type="wsla:MDVariableType"/>
        </xsd:choice>
        </xsd:sequence>
<xsd:attribute name="name" type="xsd:string"/>
<xsd:attribute name="type" type="wsla:Type"/>
<xsd:attribute name="unit" type="xsd:string"/>
</xsd:complexType>
<xsd:element name="Metric" type="wsla:MetricType"/>
```

Example:

```
<Metric name="ResponsetTime" type="float" unit="millisecond">
        <Source> strikeiron </Source>
        <MeasurementDirective xsi: type="email-verification" resultType="Float">
        <RequestURI>http://www.strikeiron.com/product-list/email/email-verification </RequestURI>
        </MeasurementDirective>
</Metric>
<Metric name="Availability" type="float" unit="percentage">
        <Source>  strikeiron </Source>
        <MeasurementDirective xsi:type=" email-verification " resultType="integer">
        <RequestURI>http://www.strikeiron.com/product-list/email/email-verification </RequestURI>
        </MeasurementDirective>
</Metric>
<Metric name="Throughput" type="float" unit="numberpersecond">
        <Source>  strikeiron </Source>
        <MeasurementDirective xsi:type=" email-verification " resultType="integer">
        <RequestURI>http://www.strikeiron.com/product-list/email/email-verification </RequestURI>
        </MeasurementDirective>
</Metric>
<Metric name="Successability" type="float" unit="percentage">
        <Source>  strikeiron </Source>
        <MeasurementDirective xsi:type=" email-verification " resultType="integer">
        <RequestURI>http://www.strikeiron.com/product-list/email/email-verification </RequestURI>
        </MeasurementDirective>
</Metric>
<Metric name="Reliability" type="float" unit="percentage">
        <Source>  strikeiron </Source>
        <MeasurementDirective xsi:type=" email-verification " resultType="integer">
        <RequestURI>http://www.strikeiron.com/product-list/email/email-verification </RequestURI>
        </MeasurementDirective>
</Metric>
```

IV. CONCLUSION

In this paper the QoS of a web service is viewed as the set of non functional characteristics/attributes that may impact the quality of the service offered by the web service. QoS can also give web service providers a significant competitive advantage in the business domain to increase of their reputation. The WSLA document defines assertions of the quality attributes that are considered at the time of selection and measures to be taken in case of deviation and failure to meet the asserted service guarantees. Here the assignment of weight at time of service selection leads to achieve maximum functionality and thereby satisfy the user requirements. Apart from fixing the expected level of non functional parameter values, the corresponding weights also assigned as per the user preference. This would determine the impact of this attribute on the final decision regarding a provider. If there is a deviation from the guaranteed level, the signing parties can take necessary action to update the service by reconsidering the domain specific and independent attributes.

V. REFERENCES

[1] I.V. Papaioannou, D.T. Tsesmetzis, I.G. Roussaki, and E.A. Miltiades, “A QoS Ontology Language for Web-Services”, pp. 101-106, 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06), 2006.

[2] E. Maximilien and M. Singh, “Conceptual model of Web service reputation,” ACM SIGMOD Record, 31(4), 2002.

[3] X. Wang , T. Vitvar, M. Kerrigan and I. Toma, “A QoS-aware Selection Model for Semantic Web Services”, ICSOC 2006

[4] Kyriakos E. Kritikos, “QoS-based Web Service Description and Discovery”, Research work for PhD thesis, Dec 2008

[5] H. Ludwig, A. Keller, A. Dan, R. King and R. Franck, Web Service Level Agreement (WSLA) Language Specification, Version 1.0, Revision wsla–2003/01/28, http://www.research.ibm.com/wsla/documents.html, 2003.

[6] Kona, S., Bansal, A., Simon, L., Mallya, A., Gupta, G., Hite, “A Service-Semantics Description Language for Automatic Web Service Discovery & Composition”, Intl. Journal of Web Services Research, Jan. 2009.

[7] L.-H. Vu, M. Hauswirth, and K. Aberer, “QoS-Based Service Selection and Ranking with Trust and Reputation Management,” Proc. 13th Conf. Cooperative Information Systems, 2005.

[8] J. Skene, A. Skene, J. Crampton, and W. Emmerich, “The Monitorability of Service-Level Agreements for Application Service Provision,” Proc. Sixth Int’l Workshop Software and Performanc, pp. 3-14, 2007.

[9] E. Rahm and P. Bernstein, ”A survey on approaches to automatic schema matching,” The International Journal on Very Large Databases 10(4), pp. 334-350, 2001.

AUTHORS PROFILE

M.Mohemmed Sha isaresearchscholarofManonmaniamSundaranarUniversity, Tirunelveli,Tamilnadu,India. He holds a B.Sc Degree in Mathematics, Masters Degree in Computer Applications at Manonmaniam Sundaranar University, Tirunelveli and M.Phil degree in Computer Applications at Bharathidasan University, Trichi. His field of interest is Web Service and Neural Networks.

Dr. K.Vivekanandan obtained his B.E degree in Electronics and Communication Engineering from Coimbatore Institute of Technology in 1986, M.Tech in Computer Science and Engineering from Indian Institute of Technology in 1991 and Ph.D in 2005 from Pondicherry University in the area of software engineering. He has 21 years of teaching, research and administration experience. He worked in Indian Institute of Science, Bangalore and Central Research Laboratory, Bharat Electronics Limited, Bangalore. Since 1992 he has been working in Pondicherry Engineering College and currently he is a Professor in Department of Computer Science and Engineering. He had also headed the Department from September 2006 to September 2009.

C. Rajalakshmi is a Assistant Professor in Management Studies, Achariya School Of Business And Technology, Pondicherry, India. She holds a B.Sc Degree in Mathematics, Master of Business Administration.

S. Pavithra is a Assistant Professor in Computer Science, Achariya School Of Business And Technology, Pondicherry, India. She holds a BCA, Master Of Computer Applications.

M. Yasotha Priya is a Assistant Professor in Computer Science, Achariya School Of Business And Technology, Pondicherry, India. She holds a B.Sc (CS), Master Of Computer Applications, M.Phil.