International Journal of Web Engineering 2014, 3(1): 1-8
DOI: 10.5923/j.web.20140301.01

QoS Measurement Tool for Web Service Selection

Susila[1,*], S. Vadivel[2]

[1]Senior lecturer, computer science dept., Bits Pilani Dubai, Dubai
[2]Professor, HOD, computer science dept., Bits Pilani Dubai, Dubai

Abstract The propagation of web services into our businesses and day-to-day lives has made quality of services (QoS) a very important aspect for both the service provider and the consumers. With a large number of web service providers providing similar services, Web service selection based on QoS classification becomes crucial for the consumer. This paper deals with ranking and selection of Web services on the basis of Entropy-Based Discretization with the help of using QoS constraints values and classifying them under corresponding service classifier. Using ranking (service classifier), client can easily choose the Web service for his need.

Keywords Web Services, Quality of Service, Entropy Based Discretization, Decision Tree Rule Induction Algorithm

1. Introduction

With the increasing use of web services in standardization of basic content integration, support of complex service-oriented architectures, provision of seamless integration of business processes and applications etc has lead to a boost in numbers of both web service consumers and providers. This means Quality of Service (QoS) becomes a very important aspect in distinguishing the success of a web service provider. [5]

From a consumer's point of view, knowing the QoS provided by the service provider plays a crucial role in choosing a particular web service over its alternatives. Therefore, knowing where the QoS of a web service ranks becomes vital for both the web service provider and the web service consumer.

This has called for tools to measure and classify web services based on various aspects of the service provided. Also, with more and more service providers providing similar functionality the customers job of searching the UDDI registry for a web service that suits his requirements while ensuring desired QoS becomes difficult.

In this project, we aim to design and implement a tool which will be able to classify web services based on their QoS and also assist customers in web service discovery based on functionality and QoS. This tool will be then implemented as a web service which will increase ease of use and improved integration. The remainder of the paper is organized as follows: Section 2 presents related work done in selection of web services on the basis of Qos attributes of the web service. Section 3 explains about web service and the need to select web service. Section 4 describes the dataset used for web service selection. Section 5 explains our feature selection ideas and reviews a core discretization algorithm [25] that is used in our method for the first round of selection and proposes tree classifier consisting of cascading decision trees. Section 6 reports our results to show that our feature selection method is effective in the selection of web service.

* Corresponding author:
susam999@gmail.com (Susila)
Published online at http://journal.sapub.org/web

2. Related Work

Selection of Web Services on the Basis of Quality of Service Constraints was proposed in the paper [13]. In that QoS Manager had a role of being a moderator amongst the provider and the client. In paper [4] they discuss about Qos broker publish system that Extract the quality of service constraints in the issued WSDL and the values extracted are stored in QoSDB and the fundamental features are issued in the UDDI registry. Service matching procedure is applied, and finally, service with the highest quality selected and proposed to the service requester. In paper [8] they formulates a robust QoS semantic framework for Web Services into three layers QoS-ontology, which can provide a standard model to formally describe arbitrary QoS parameters and exhibits properties [18] [19] [20]. Paper [21] dealt with, prominent works that apply fuzzy theory for representing imprecise QoS constraints and preferences and for developing QoS based ranking algorithm for Web services which can deal with fuzzy QoS values.In [22], the service selection for a Web service composition problem PWSDCP is dignified as contentment difficulty that belongs to a fuzzy attribute.

3. Web Services

Web Services are outcome of the advancement of the web into a means of scientific, commercial and social exchanges. A Web service can be described as a way of calling a function which is inside software from another software. The software which makes the call is called as the client and the software which services the client is called a server. The two softwares might have been programmed using different languages and could be running on different machines but have to be connected by a network. Web services have an interface expressed as the WSDL (Web services description language) file. WSDL file can be seen a contract for communications between the web service client and server.

With the rapid growth in the field of web services, Quality of service (QoS) becomes one of the important factors based on which the service requestor distinguishes the success of service providers [15].

Some of the aspects of QoS include response time, reliability, scalability, throughput, robustness, success ability, exception handling, reliability, accuracy, integrity, accessibility, availability, interoperability, security, and network-related QoS requirements.

The aim of this project is to classify the web services based on the response time, availability, throughput, success ability and reliability and hence help the service requestor choose a web service which best suits his requirements.

1. Response time(ms):

Response time is a factor of web services which tell us how fast a web service can service our request. Response time is a vital performance measure which is used to grade web service an lesser response latency is preferred by web service consumers.

Response Time = Time taken to complete the response – Time taken for user request.

2. Availability (%):

Availability is the probability that the system is up and read for immediate consumption when invoked. Service providers have to guarantee high availability ratio of their web service so as to ensure customer satisfaction. A web service with is prone to frequent outages or unavailability might lead to adverse effects on the business of the customers and on the standing of the web service provider.

Availability = 1 – (Down time/ Unit time)

3. Throughput(invokes/second):

Throughput represents the number of invocation per unit time for a given web service. This corresponds to the ability of a web service to accept an incoming request and to counter it in accordance with the requisite performance constraints.

Throughput = Max. Completed requests / Unit Time

4. Successability(%):

Success ability represents the ratio of the number of successful responses to the number of requests. It tells us how often a web service responds given a certain number of requests or how often the web service fulfills its service commitments.

Successability = No. Of responses / No. Of requests

5. Reliability (%):

Reliability corresponds to the ability of a web service to execute its required functions under the given conditions for a particular time interval. [16] Reliability is computed as a function of 6 characteristic features which are Accuracy (C), fault tolerance (F), testability (T), interoperability (I), availability (A), and performance (P).[21]

Reliability = f(aC, bF, cT, dI, eA, fP)

Where a, b, c, d, e, and f are the weight associated with the contribution of each attribute to the reliability of the web service.

4. Entropy

Initially, the range of a continuous variable, from a database sample, is divided into intervals which contain at least one case each. This is done after sorting on the variable values. At most, there would be *m* intervals (*O* (*m)*) for *m* cases. This converts the continuous variable into a discrete one, with *O* (*m)* values [16]. Entropy, or information, is maximized when the frequency probability distribution has the maximum number of values. Since there is a discrete partition for every distinct value in the continuous distribution in the database, there is no information or entropy loss from the database sample.

The entropy of a discrete random variable *X* is defined as

$$\text{Entropy (t)}=\sum_{i=0}^{c-1} p\left(\frac{i}{t}\right) \log 2\ p\left(\frac{i}{t}\right) \quad (1)$$

We can use a measure called Information Gain, which calculates the reduction in entropy (*Gain in information*) that would result on splitting the data on an attribute, A.

$$\text{Gain (S,A)} = \text{Entropy (S)} \sum_{ve\ A} \frac{|Sv|}{|S|} \text{ Entropy (Sv)}$$

Where v is a value of A, |Sv| is the subset of instances of S where A takes the value v, and |S| is the number of instances

Using Entropy-based Discretization, classification of web services could be done easily. Using this classification the requester could choose the most suitable web service according to his/her requirements and preferences. Using Entropy-based Discretization we obtained a tree (Figure 6.1). In this tree, nodes belong to the QoS attributes. Tracing these nodes, we could reach leave nodes, which represent the classification of the web service into four classes.

In order to present the most suitable service to the service requester, we used the Quality of Service attributes, as they completely define a web service. In this paper, we proposed to use Entropy-based Discretization in order classify web services into four classes, i.e. Poor, Good, Average and Excellent. Using this method we obtained a classifier tree. Using this tree, we could classify any new web service into the four classes mentioned earlier by tracing the tree according to their QoS attribute values.

Many features are irrelevant to the classification. Takingsuch features into account during classification

increases the dimensionality of the problem, raises many computational difficulties, and potentially introduces noise effecting the classification accuracy [12, 22]. So, how to select important features for classifications a problem that has been attracting tremendous research effort previously and currently. So we have considered only five attributes namely throughput, response time, success ability, reliability and availability.

Decision tree construction

The entropy based discretization is used to construct the decision tree using a variant of the ID3 algorithm. ID3 algorithm looks through the attributes of the training set and extorts the attribute that best splits the given examples. If the attribute perfectly classifies the training sets then the ID3 algorithm stops; else it recursively executes on the split subsets to get their "best" splitting attribute. The algorithm exploits a greedy search technique. The “best” splitting attribute is determined by means of entropy and information gain. [7]

The variant of ID3 used has been modified to handle continuous attributes in the test and training sets.

```
BuildDecTree (Training_Set ) returns node_ptr
  IF all entries of the Training_Set all of the same class
THEN  return a pointer to a LEAF node with class = one with the most
examples
  ELSE
     choose split_point  S based on information gain.
     IF splitting fails
     THEN return pointer to LEAF node as above
     ELSE
split_examples( S, Training_Set, +examples, -examples)
       LET +b = BuildDecTree( +examples )
       AND -b  = BuildDecTree( -examples )
       return a pointer to a BRANCH
                  with criteria   =  S
                     +branch =  +b
                     -branch  =  -b
     END {if choosing fails}
  END {if examples of same class}
```

Figure 1. Pseudo code of the algorithm

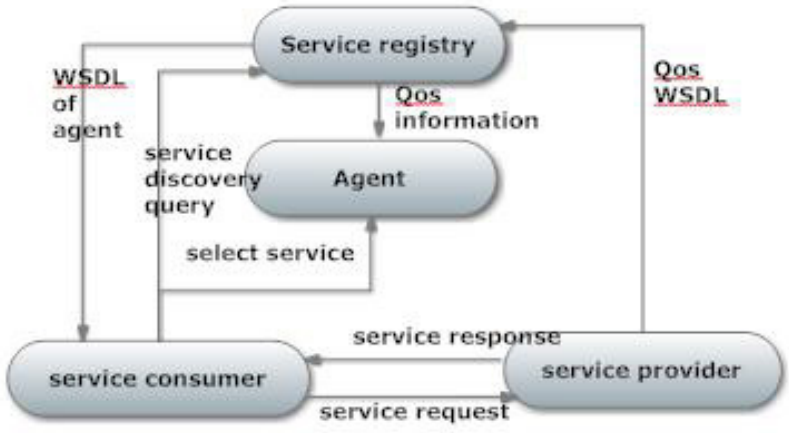

Figure 2. Broker based architecture for web service selection

Once the decision tree has been constructed using the algorithm, the decision tree could be used to classify new web services based on their QoS parameters namely response time, availability, throughput, success ability and reliability.

In this project we assume that the QoS information is stored in the WSDL document which are registered in the UDDI registry and is updated by the service providers in case of amendments. Our service discovery agent/broker obtains the QoS information from the WSDL documents, stores them in its database and classifies the web service based on its QoS information. Therefore, a service requestor who is looking for a service can send a service discovery request to the agent with his requirements and specification and the agent will look up through its data base and suggest the available services which meet the requestor's requirements.

Input Data Set

The input data set used for the decision tree induction and QoS classification is the "QWS Dataset"[12]. The QWS dataset consists of data from over 5000 web services out of which the public dataset consists of a random 365 web services which have been chosen and nine QWS(Quality of Web Service) attributes have been measured. Each web service was tested for over a duration of over ten-minutes for three successive days.

ID	Parameter Name	Description	Units
1	Response Time	Time taken to send a request and receive a response	ms
2	Availability	Number of successful invocations/total invocations	%
3	Throughput	Total Number of invocations for a given period of time	invokes/second
4	Successability	Number of response / number of request messages	%
5	Reliability	Ratio of the number of error messages to total messages	%
6	Compliance	The extent to which a WSDL document follows WSDL specification	%
7	Best Practices	The extent to which a Web service follows WS-I Basic Profile	%
8	Latency	Time taken for the server to process a given request	ms
9	Documentation	Measure of documentation (i.e. description tags) in WSDL	%
10	WsRF	Web Service Relevancy Function: a rank for Web Service Quality	%
11	Service Classification	Levels representing service offering qualities (1 through 4)	Classifier
12	Service Name	Name of the Web service	None
13	WSDL Address	Location of the Web Service Definition Language (WSDL) file on the Web	None

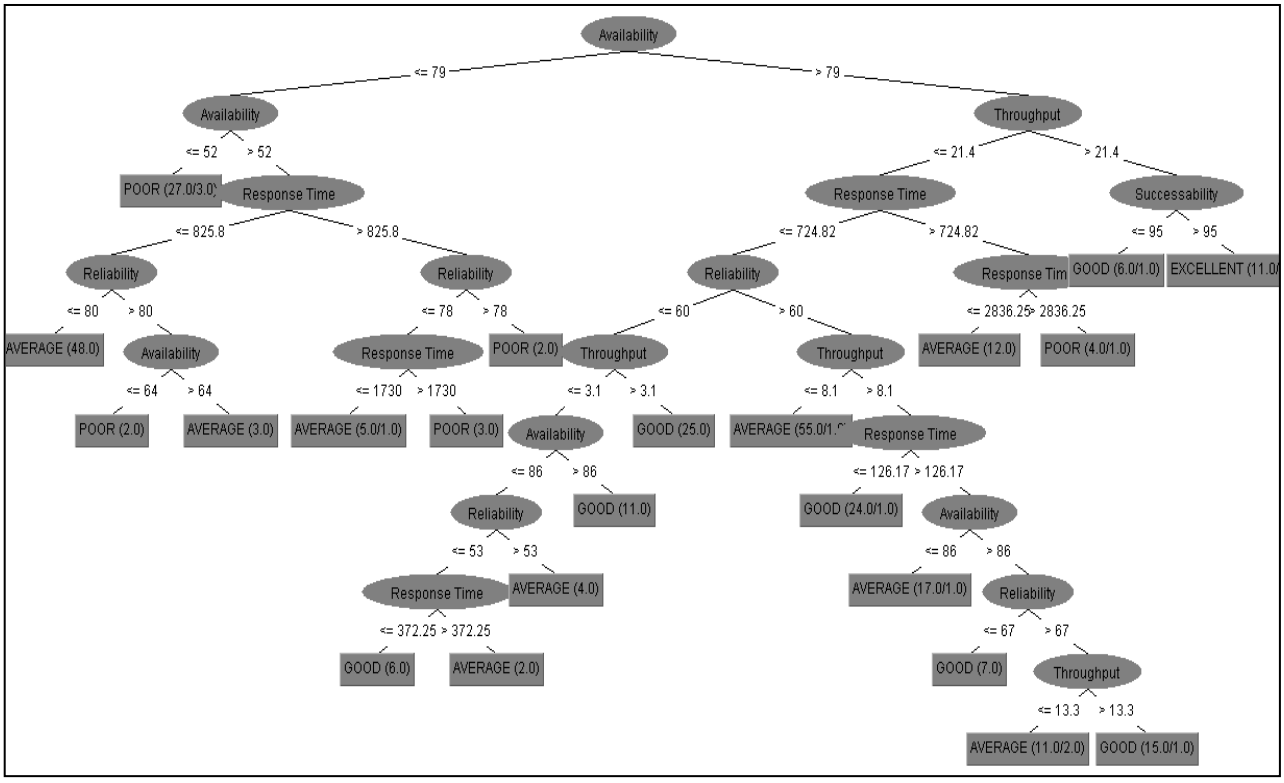

Figure 3. Decision tree classifier

5. Algorithm and Result

Decision tree rule induction algorithm using entropy based discretization.

Modification in the proposed algorithm design:

ID3 algorithm has been used because of its quick build time and accuracy comparable to that of other algorithms like C4.5 for numeric datasets. ID3 inherently uses entropy based discretization for creating pure bins out of the training dataset. But the drawback with ID3 algorithms is, the performance of ID3 in case of continuous attributes deteriorates to a high extent. But in our modified algorithm, it uses a variation of ID3 algorithm to induce the decision tree to enable decision tree classification for continuous datasets.

Implementation:

The Java based implementation of this algorithm has been carried out and results of the same have been recorded. The result consists of a decision tree which is represented in the form of rules. This java program is also capable of accepting inputs of QoS parameter test values from the user, traversing the decision tree and giving out the classification of the test web service. The decision tree when processed and represented in a tree format will look like figure 4.

Design

The rules generated using the Decision tree rule induction algorithm are consolidated and made into a SOAP based web service.

Implementation

The web service uses Java API for XML Web Services (JAX-WS) which is an significant part of the Java EE 5 and EE 6 platforms. The web services is deployed on the glassfish server. On successful deployment, the web service is then tested using the IDE's tester page which runs on the local browser[13].

6. Testing and Output

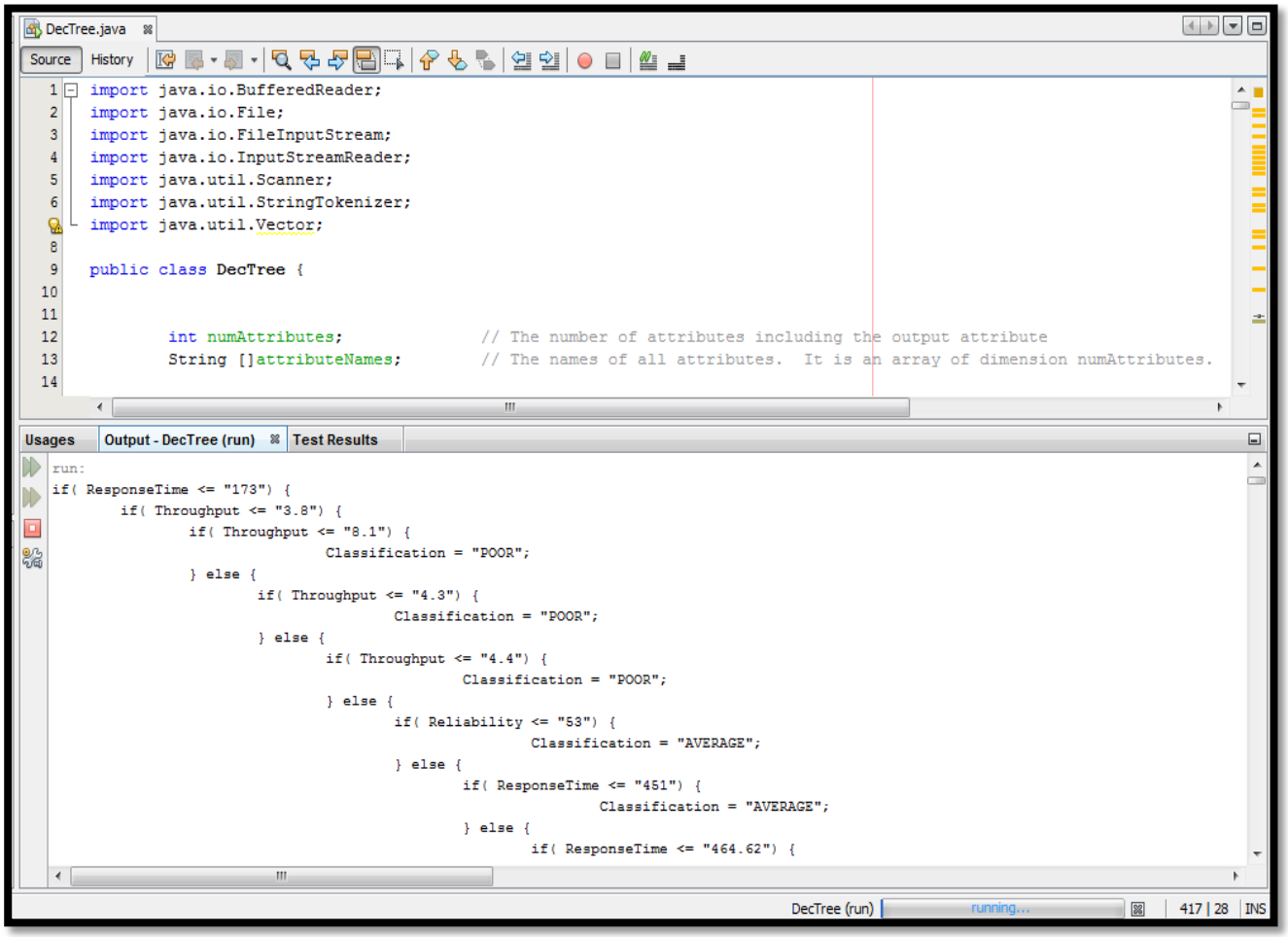

Figure 4. Screenshot showing output of the Decision tree rule induction algorithm

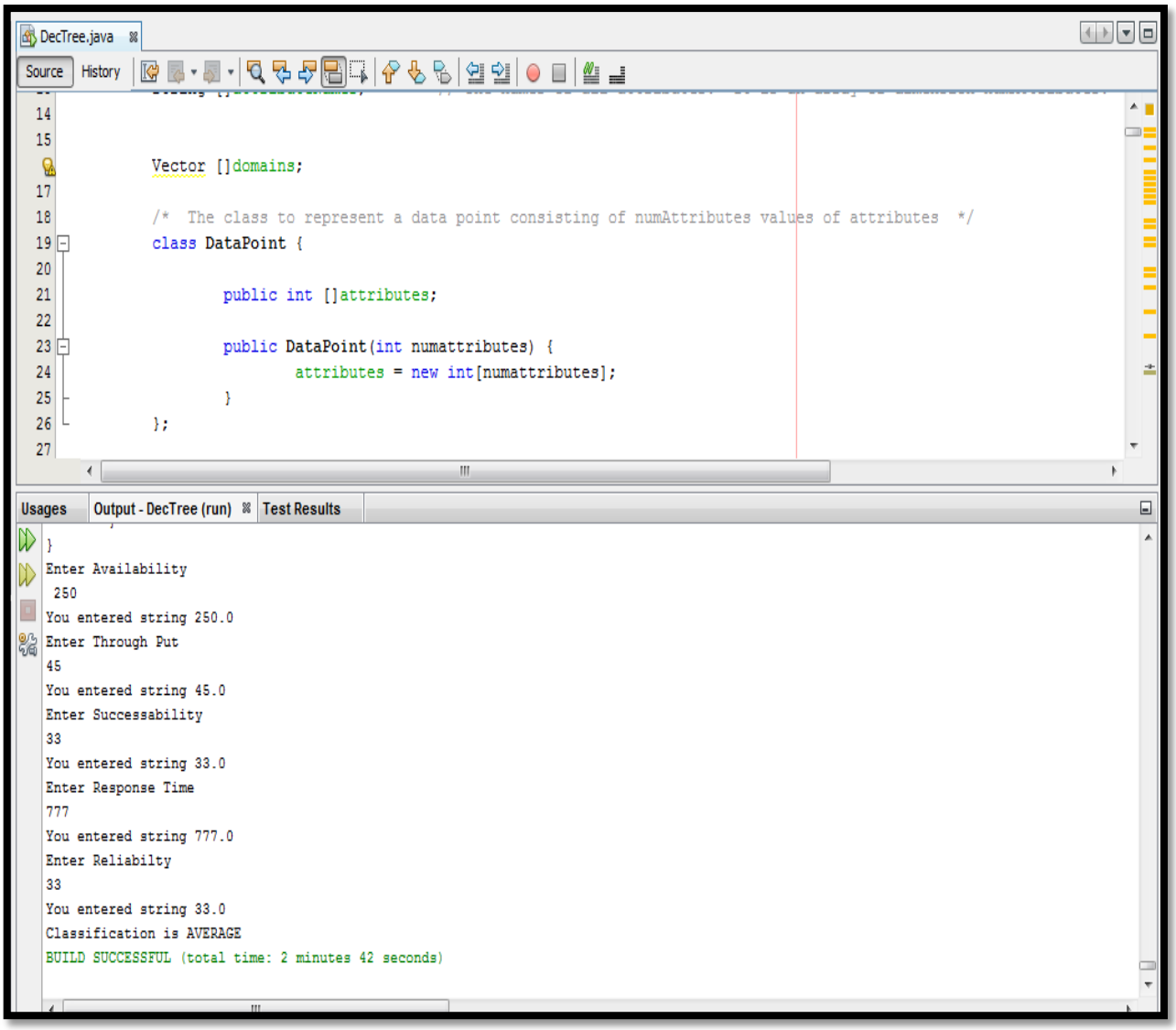

Figure 5. Decision tree traversal and classification using Decision tree rule induction algorithm

Figure 6. IDE's tester page for the QoS classification web service

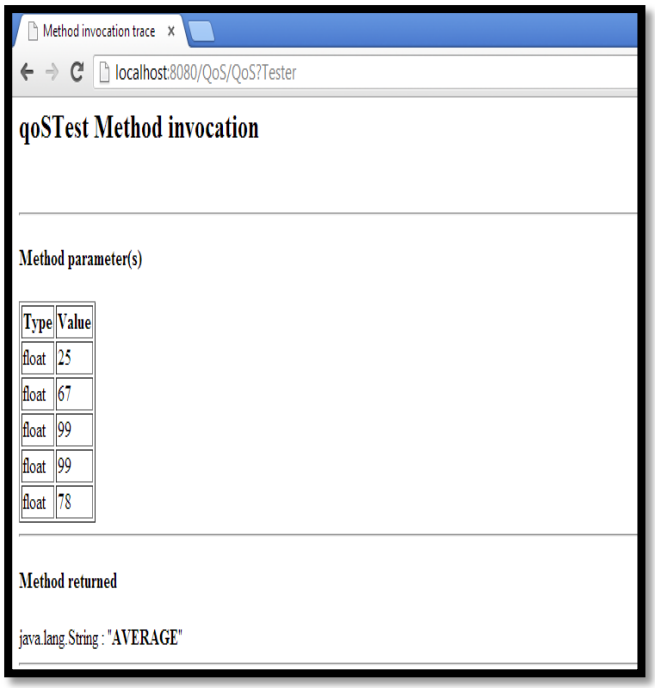

Figure 7. IDE's Method invocation trace for the QoS classification web service

This fig shows, the IDE's tester page for the QoS classification web service. Here we expect the client to know the values for the attributes that he is looking for throughput, response time, success ability, reliability and availability. As he enters the values for these attributes, according to the mining results we have, classification of the web service is done.

Clippings of the SOAP request and response file for the QoS classification web service is given fig 7 and fig 8 below,

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:QoSTest xmlns:ns2="http://qos.me.org/">
            <Availability>25.0</Availability>
            <ResponseTime>67.0</ResponseTime>
            <Throughput>99.0</Throughput>
            <Reliability>99.0</Reliability>
            <Successability>78.0</Successability>
        </ns2:QoSTest>
    </S:Body>
```

Figure 8. SOAP request for the QoS classification web service

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:QoSTestResponse
xmlns:ns2="http://qos.me.org/">
            <return>AVERAGE</return>
        </ns2:QoSTestResponse>
    </S:Body>

</S:Envelope>
```

Figure 9. SOAP response for the QoS classification web service

7. Conclusions

Web Services are becoming increasingly used and a large number of consumers are building their business solutions using web service technology. The need for QoS specifications for web services have arisen due to consumer's prospect for superior web service performance and Services provider's obligation to provide high quality service so as to improve the usability and utility of their services which in turn decides their standing in the market. This report dealt with the design and implementation of a QoS classifier which takes in the QoS parameters as input and yield the classification of web services based on those aspects as the result and the same classifier was also implemented as a web service as to improve ease of access. A service discovery agent was also implemented and deployed as a web service. Even though this implementation wasn't a full-fledged one, it does give us an intuitive overview of a service discovery agent to enable us understand the basic working of such a system.

REFERENCES

[1] Liang – Jie – Zhang, Jia Zhang, "Services Computing" Springer Berlin Acidelberg, Newyork.

[2] Thomas Erl, "Service Oriented Architecture: A Field Guide to Integrating XML and Web Services", Prentice Hall Publications, 2004.

[3] Jeberson R. Retna Raj, Sasipraba T., "Web Service Selection Based on QoS Constraints", IEEE, 2010.

[4] Liu Sha Guo Shaozhong Chen Xin Lan Mingjing Zhengzhou,

" A QoS Based Web Service Selection Model", International Forum on Information Technology and Applications, 2009.

[5] Manish Godse, Umesh Bellur, Rajendra Sonar, "Automating QoS based Service Selection", 2010 IEEE International Conference on Web Services PP: 530-540.

[6] Kyriakos Kritikos and Dimitris Plexousakis, "Requirements for QoS Based Web Service Description and Discovery", IEEE Transactions on Services Computing, Vol. 2, 2009.

[7] Sidney Rosario, Albert Benveniste, "Probabilistic QoS and Soft Contracts for Transaction-Based Web Services Orchestrations", IEEE Transactions on Services Computing, 2008.

[8] Qu Li-li, Chen Yan, "Qos Ontology Based Efficient Web Services Selection", International Conference on Management Science & Engineering, 2009.

[9] Serhani M. A., Dssouli R., Hafid A., Sahraoui H., "A QoS broker based architecture for efficient web services selection", IEEE International Conference on Web Services, 2005.

[10] Yu T., Lin K. J., "Service selection algorithms for Web services with end-to-end Qos constraints", Proceeding of Information Systems and E-Business Management, 2005.

[11] Tran Vuong Xuan, Tsuji Hidekazu, "QoS based ranking for Web Services: Fuzzy Approach", International Conference on Next Generation Web Services Practices, 2008.

[12] Architecture of the World Wide Web, First Edition, W3C Working Draft, I. Jacobs, 9 December 2003 See http://www.w3.org/TR/2003/WD-webarch-20031209/.)

[13] Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001 (http://www.w3.org/TR/wsdl#_service).

[14] Service Oriented Architecture (SOA) and Specialized Messaging Patterns, Adobe Technical White Paper, Duane Nickull, Laurel Reitman, James Ward, Jack Wilber.

[15] http://www.service-architecture.com/web services/articles/web_services_explained.html.

[16] http://www.ibm.com/developerworks/webservices/library/ws-quality/index.html.

[17] http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/#qos-2.

[18] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm.

[19] http://www.hiraeth.com/books/ai96/QBB/id3.html.

[20] Al-Masri, E., and Mahmoud, Q. H., "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258.

[21] Rajesh Sumra and Arulazi. D.,Quality of Service for Web Services-Demystification, Limitations.http://www.developer.com/services/article.php/10928_2027911_2/Quality-of-Service-for-Web-ServicesmdashDemystification-Limitations-and-Best-Practices.htm.

[22] Jia Zhang and Liang-Jie Zhang, Criteria Analysis and Validation of the Reliability of Web Services-oriented Systems, IEEE International Conference on Web Services (ICWS'05).

[23] Al-Masri, E., and Mahmoud, Q. H., "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258.

[24] Getting Started with JAX-WS Web Services, www.netbeans.org.

[25] Dr. Iiavarasan Egambaram, G. Vadivelou, S. Prasath Sivasubramanian., qos based web service selection.

[26] Bernhard Borges, Kerrie Holley and Ali Arsanjani, IBM, Service-oriented architecture, http://searchsoa.techtarget.com/news/1006206/Service-oriented-architecture.