Ontology Based Qos Driven Web Service Discovery

R.Suganyakala,M.Aarthilakshmi[1], Dr.G.R.Karpagam[2] and Ms.S.Maheswari[3]

[1] Students,Department of ComputerScience and Engineering,PSG College of Technology
Coimbatore, Tamil Nadu, India

[2] Professor, Department of ComputerScience and Engineering,PSG College of Technology
Coimbatore, Tamil Nadu, India

[3] Assistant Professor,Department of ComputerScience and Engineering,PSG College of Technology
Coimbatore, Tamil Nadu, India

Abstract

In today's scenario web services have become a grand vision to implement the business process functionalities. With increase in number of similar web services, one of the essential challenges is to discover relevant web service with regard to user specification. Relevancy of web service discovery can be improved by augmenting semantics through expressive formats like OWL. QoS based service selection will play a significant role in meeting the non-functional user requirements. Hence QoS and semantics has been used as finer search constraints to discover the most relevant service. In this paper, we describe a QoS framework for ontology based web service discovery. The QoS factors taken into consideration are execution time, response time, throughput, scalability, reputation, accessibility and availability. The behavior of each web service at various instances is observed over a period of time and their QoS based performance is analyzed.

Keywords: *Web services, Semantic Web Service Discovery, Ontology, Quality of service, QoS based WSD, QOS parameters.*

1. Introduction

Basically web services are used as a means for businesses to communicate with each other and with clients , Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. When there are many web services that provide the similar functionalities finding the relevant service becomes a tedious task.The Semantic Web plays an important role in making the Web more relevant. The aim of semantic Web services is to create a semantic Web service whose properties, capabilities, interfaces and effects are unambiguously described and used by machines. The current UDDI registries only support Web services discovery based on the functional aspects of services. Quality of service (QoS) which is a set of non-functional attributes (for example, response time and availability) that may have impact on the quality of the service provided by Web services. QoS ranking provides a facility to refine the search and get the web service that matches the best to user requirements

2. Web Service

The term Web services describes a standardized way of integrating web based applications using XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Web services allow different applications from different sources to communicate with each other without time-consuming custom coding and also because all communication is in XML, Web services are not tied to any one operating system or programming language hence it is both language and platform independent in nature. Traditional RPC-style middleware, such as RPC, CORBA, RMI, and DCOM, relies on tightly coupled connections. A tightly coupled connection is very brittle, and it can break if you make any modification to the application. In contrast, Web services support loosely coupled connections. Loose coupling minimizes the impact of changes to the applications. A Web service interface provides a layer of abstraction between the client and server. A change in one doesn't necessarily force a change in the other. The abstract interface also makes it easier to reuse a service in another application. Loose coupling reduces the cost of maintenance and increases reusability. The major problem faced by web service users is that finding the most relevant service that matches their requirements as there are numerous web services with similar functionalities is available. Hence semantics helps to make the web services to improve the relevancy.

3. Transition from Syntactic To Semantic Web Services

Syntactic Web describe the current, mostly HTML-based World Wide Web. The semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. WSDL, SOAP and UDDI are Web Service related standards. On taking a closer look, these standards allow for describing a Web Service in a standardized format and publishing it in a UDDI repository. however, UDDI service descriptions merely provide some meta-data on the service and are often even natural language based, making it extremely difficult for automated algorithms to understand the exact semantics and capabilities of a service and choosing the correct service in order to complete a task. A WSDL document on the service does not help much here, it merely contains information on the syntax of accessing services, again, the information does not suffice for an automated algorithm to choose a service in order to reach a particular goal. The standards allow for composing services and creating more complex services , the workflow models resulting from such compositions are executed in a deterministic, non-flexible way, with no means of simply declaring a particular service in the workflow as abstract and leaving it to a execution engine to dynamically select a service at run-time. Or, moreover, to replace an existing service in the workflow dynamically by a new service featuring, for example, better quality of service.

- In fact, WSDL, UDDI and SOAP provide solutions to low-level infrastructure problems in the context of Web Service communication, description, discovery and composition.
- Leveraging the power of Web Services, for example enabling automated Web Service discovery and composition, requires connecting the Web Service domain to research areas that deal with searching, processing and combining information.
- These standards do not provide any means of re-planning a Web Services workflow on the way in case a service fails.
- They are not able to choose the best service among a large set of available and semantically similar services.
- They cannot determine the exact capabilities of a service and decide whether the service might be of any use for solving another service's task in combination with further services.

The solution to this problem comes in form of formally defined, linked data on the Web. Machine algorithms require data to be stored in formalized syntax in which terms used are associated with predefined semantics. Semantic Web aims at defining knowledge representation methods which can be used in such manner. A crucial role in these efforts is played by ontologies. Ontologies are a formal, explicit specification of a shared conceptualization of a domain. Essentially, ontologies capture knowledge about a particular domain in a machine-processable way. The approach which is used to bring semantics into web service is, OWL-S, a Web Service description language that semantically describes the Web using OWL ontologies. OWL-S services are then mapped to WSDL operations and inputs and outputs of OWL-S are mapped to WSDL messages.

3.1 Owl in semanticweb

The motivation for the structuring of the ontology of webservices is the need to provide three essential types of knowledge[1] about a service (shown in Figure).

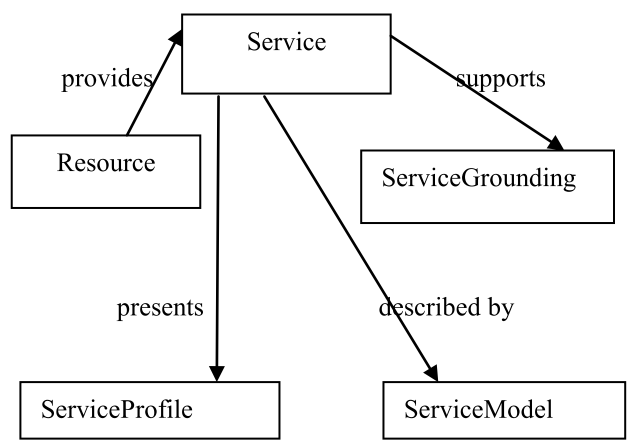

Fig 1 organisation of OWL-S into modules[1]

3.1.1 Service profile

The service profile says what the service does, in a way that is suitable for the client to determine whether the service meets its needs.

3.1.2 Service model

The service model describes to a cl ient how to use the service, by detailing the semantic content of requests.

3.1.3 Service grounding

A service grounding is concerned about the details of how an agent can access a service.

4. Life Cycle of Semantic Web Service

Table 1 describes lifecycle of semantic webservice

Table 1: Phases in the lifecycle of semantic webservice

Phase	**Description**
Service Annotation	To effectively perform operations such as the discovery of services, semantics of the input/output data has to be taken into account. Hence, the data involved in Web service operation is annotated using an ontology
Service Advertisement	After the service is developed and annotated, it has to be advertised to enable discovery. The UDDI registry
Service Discovery	This stage is the process of discovering appropriate services before selecting a specific Web service
Service Selection	After discovering Web services whose semantics match the semantics of the requirement, the next step is to select the most suitable service.
Service Composition	This stage involves creating a representation of Web processes(composition of web services).
Service Execution	The web process is executed.

5. Case Study(Online Shopping Webservices)

There is a remarkable range for growth in trade through electronic interactions, simply because it can eliminate geographical distances in bringing buyers and sellers together. With the Internet dissemination and the e-commerce growth there is a shift from the traditional off-line distribution process based on organization's catalogues to on-line services. A shift that is marked by isolated initiatives guided by the business-to-customer and business-to-business promise of increased profit margins and reduced commission .Therefore, organizations are increasingly faced with the challenge of managing e-business systems and e-commerce applications managing Web services, Web processes, and semantics. So we have chosen online Shopping system as our case study. Web services promise universal interoperability and integration. The key to achieve this relies on the efficiency of discovering appropriate Web services.

5.1 Webservices Implemented

5.1.1 Login service

When a user wants to login, he/she may use Login Service provided by the system which facilitates the user to obtain service of a particular type.

Table 2:Login webservice

Service name	**Description**
Login 1	Uses MD5 for security
Login 2	Uses SHA1 for security
Login 3	Uses SHA256 for security
Login 4	Uses Triple DES for encryption
Login 5	User authentication using image path and id
Login 6	User authentication by RIPMD16 hashing mechanism
Login 7	User authentication by mail id
Login 8	Uses encrypted password
Login 9	Uses minimal details and less time consuming
Login 10	Uses both encrypted username and password

5.1.2 Register service

When a user wants to register to use facilities of online shopping,he/she may use Register Service provided by the system which facilitates the user to perform registration in various types :

Table 3:Register webservice

Service name	**Description**
Register 1	Uses MD5 for storing password details
Register 2	Uses SHA1 for storing password details
Register 3	Uses SHA256 for storing password details
Register 4	Uses Triple DES for encryption
Register 5	User registration using image path and id
Register 6	User registration using encrypted image path and id

Register 7	User registration by getting more details and hence time consuming
Register 8	User registration by getting more details and encrypted password
Register 9	User registration by getting less details and hence less time consuming
Register 10	Uses both encrypted username and password for registration

5.1.3 Productdisplay service

When the user wants to select the products to be purchased then this service can be used to provide the display of products according to the user requirements in the following ways

Table 4:Display product webservice

Service name	Description
Display 1	Displays the products based on the type of category chosen
Display 2	Displays the products based on the cost of the products
Display 3	Displays the products in alphabetical order irrespective of the category
Display 4	Displays the products based on its reputation from the user's rating

6. Qos Framework

Quality of service (QoS) is defined as the non-functional attributes (e.g. response time, availability etc.) that may have impact on the overall performance of any web service.For organizations, being able to characterize Web processes based on QoS has several advantages: a) it allows organizations to translate their vision into their business processes more efficiently, since Web processes can be designed ac-cording to QoS metrics, b) it allows for the selection and execution of Web processes based on their QoS, to better fulfill customer expectations, c) it makes possible the monitoring of Web processes based on QoS, and d) it allows for the evaluation of alternative strategies when Web process adaptation becomes necessary. Quality criteria may have different definitions in different domains. However, in the Web services context, Quality criteria can be defined as a set of non-functional criteria such as availability, performance and reliability that impact the performance of Web services[as shown in fig 2]. Quality is the measure of how well does a particular service perform relative to expectations, as presented to the requester.

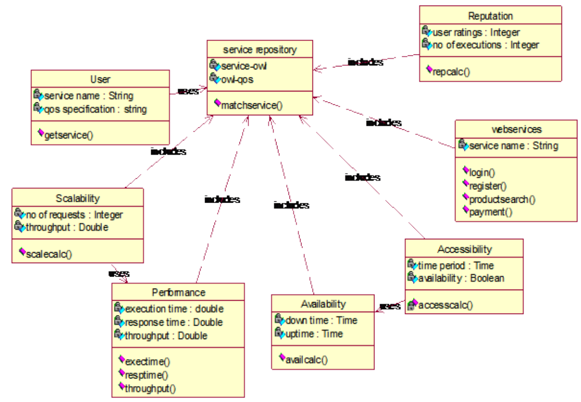

Fig 2 class diagram of the QoS framework

6.1 Performance

The performance of a Web services measure the speed in completing a service request. It can be measured by:

- **Response time-** The maximum time that elapses from the moment that a web service receives a SOAP request until it produces the corresponding SOAP response.[7]
- **Execution (processing) time-** The time taken by a Web service to process its sequence of activities.
- **Throughput**-The number of Web service requests *R* for an operation *o* that can be processed by a service S within a given period of time is referred to as throughput tp(S, o).

It can be calculated by the following formula[7]:

$$t_p = \frac{R}{time\ period(in\ sec)} \quad (1)$$

In general, high performance Web services should provide higher throughput, higher capacity, faster response time, lower latency, and lower execution duration.

6.2 Availability

It is the probability that a service is operating when it is invoked[7]. It is the probability that a service *S* is up and running.Formula used:

$$av(s) = 1 - \left(\frac{downtime}{uptime}\right) \quad (2)$$

- The downtime and uptime are measured in minutes

6.3 Scalability

A Web service that is scalable, has the ability to not get overloaded by a massive number of parallel request[7]. A high scalability value states the probability for the requester of receiving the response in the evaluated response time t_{rt}

Formula:

$$sc(S) = t_r / t_{rt(throughput)} \quad (3)$$

where $t_{rt(Throughput)}$ is the round trip time which is evaluated during the throughput test[7].

6.4 Reputation

It is the measure of trustworthiness of a service, based on the end user's experiences of using the service. Different end users may have different opinions on the same service. The reputation can be defined as the average ranking given to the service by the end users. The value of the reputation is computed using the following expression :

$$q_{rep} = \frac{\sum_{i=1}^{n} R_i}{n} \quad (4)$$

where R_i is the end user's ranking on a service's reputation, n is the number of times the service has been graded. Usually, the end users are given a r ange to rank Web services[4].

6.5 Accessibility

It is the capability of serving the Web Service request. The Web service might be available but not accessible because of a high volume of requests. Accessibility can be represented by the following formula[4]:

$$P_{accessibility} = P_{availability} \text{ at } Time\ T = t \quad (5)$$

7. Interdependencies

There are few interdependencies among the QoS parameters that are described as follows and represented in fig 3.

- In general, high performance Web services should provide higher throughput, higher capacity, faster response time, lower latency, and lower execution duration.
- Here throughput is inversely proportional to execution time and it varies directly with response time.
- Accessibility of a web service depends on the availability of a service at a given time and it s directly proportional to it.[7]
- Scalability of a web service depends on the response time and the time taken for throughput test. Since the response time, execution time, throughput and scalability cannot be implemented as separate services and hence cannot be invoked to calculate these QOS values for any service because they will in turn call their corresponding services hence it leads to a cycle or loop formation. Factors that can be evaluated during execution are execution time, response time, throughput, scalability.
- Factors that can be evaluated after execution are:reputation ,availability, accessibility.

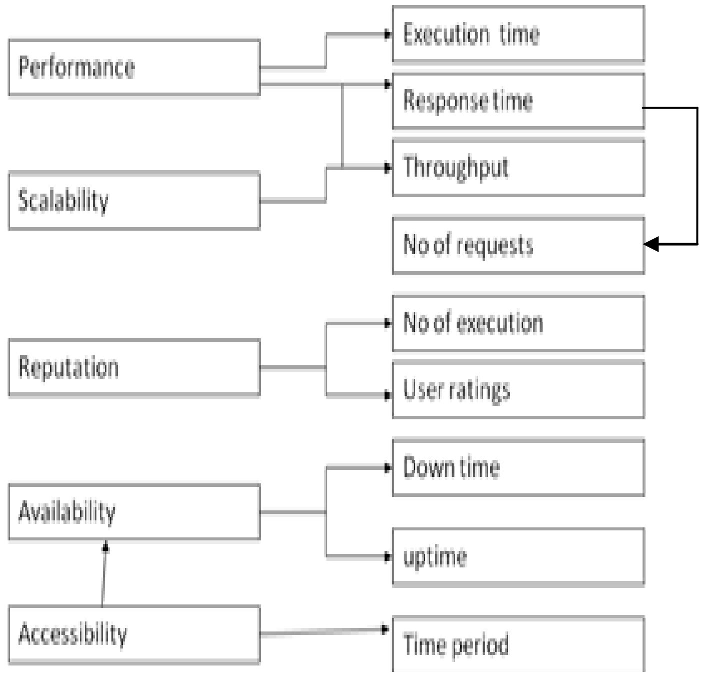

Fig 3 interdependencies between the parameters

7.1 Testing

Testing is a set of activities conducted with the intent of finding errors in software. It is the process of exercising software to verify that it s atisfies specified requirements and to detect errors. Validation is the process of finding errors by executing the program in a real environment with inputs. The main principle of software testing is to remove as many defects as possible before test since the quality improvement potential of testing is limited. Unit testing is concerned with knowledge about testing a program unit, to determine that it is free of data, logic, or standards errors. This unit includes knowledge of dynamic analysis (partitioning, boundary value analysis, logic-based testing and syntax testing) and static analysis (decision testing, condition testing, and data-flow testing). In Unit testing individual components or modules are tested.

Table 5: Details of unit testing

Test case Name	Related requirements	Remarks	Current status Pass /Fail
Execution time	-	Unsuccessful when a variable to calculate time was not declared	Pass
Response time	-	Successful	Pass
Throughput	Response time	Unsuccessful when a w hile loop got executed infinite number of times	Pass
Reputation	-	Unsuccessful when there was no ratings found in database for a particular web service	Pass
Availability	-	Unsuccessful when there was no entry found in database for a particular web service	Pass
Scalability	Response time	Successful	Pass
Accessibility	Availability	Unsuccessful when there was no entry found in database for a webservice	Pass

8. Empirical Evaluation of QOS parameters

In order to find the interdependencies between the parameters and also to find the variation in the values of the QoS parameters for different web services, we have executed the services at various instances of time period and their QoS values are noted. From these observed values, graphs have been drawn to make a valid conclusion about the behavior of each parameter for different services and also its variation for the same service at different instances.

8.1 Analysis of QoS parameters

In order to find the interdependencies between the parameters and also to find the variation in the values of the QoS parameters for different web services, we have executed the services at various instances of time period and their QoS values are noted. From these observed values, graphs have been drawn to make a valid conclusion about the behavior of each parameter for different services and also its variation for the same service at different instances.

8.1.1 Execution Time

The following graph shows the variation of execution time for various types of login service. It has the service name in X-axis and execution time (milliseconds) in Y-axis. It shows that depending upon the complexity of the internal function the execution time varies. For example in this case, the minimum value(15 ms) is obtained for login1 which uses simple encryption whereas login5(968 ms) which uses image storing method for user authentication obtains a maximum value.

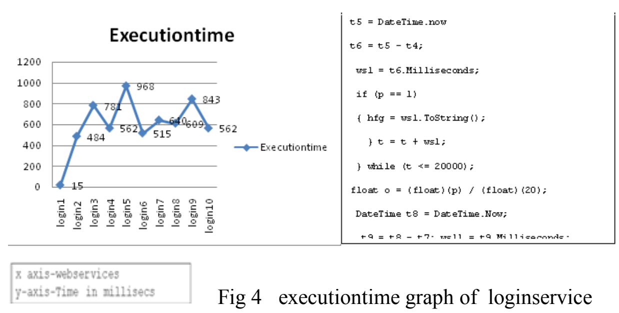

Fig 4 executiontime graph of loginservice

8.1.2 Response Time

The following graph shows the variation of response time for various types of login service. It has the service name in X-axis and response time (milliseconds) in Y-axis. Response time varies with time required to respond to a SOAP request so we have login10 with highest value(823 ms) because both username and password are encrypted here and minimum for login4 (31 ms).

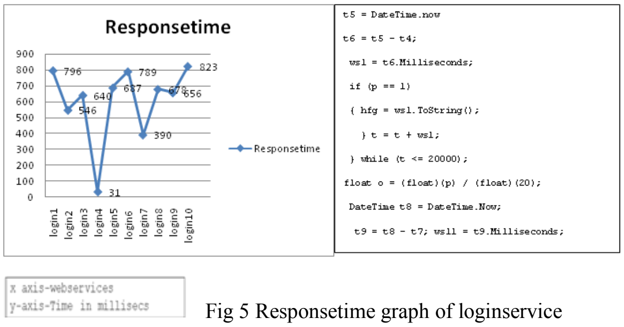

Fig 5 Responsetime graph of loginservice

8.1.3 Throughput

The following graph shows the variation of throughput for various types of login service. It has the service name in X-axis and t hroughput in Y-axis. Here login2 has highest throughput (4.85) and login1(1.2) has lowest throughput.

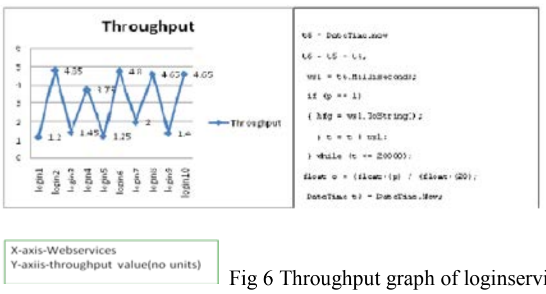

Fig 6 Throughput graph of loginservice

8.1.4 Scalability

The following graph shows the variation of scalability for various types of login service. It has the service name in X-axis and scalability in Y-axis. Web service Login6(0.87) has the highest value and login4 has the lowest value(0.05).

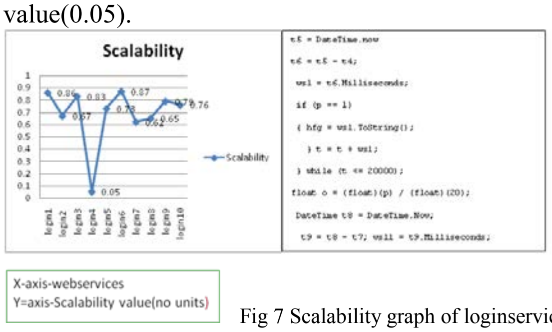

Fig 7 Scalability graph of loginservice

8.1.5 Reputation

The following graph shows the variation of reputation for various types of login service. It has the service name in X-axis and reputation in Y-axis. Here it is inferred that login4(5) has highest reputation because of its security aspect and login9(1) which doesn't have any encryption and needs only less user details has a minimum reputation value.

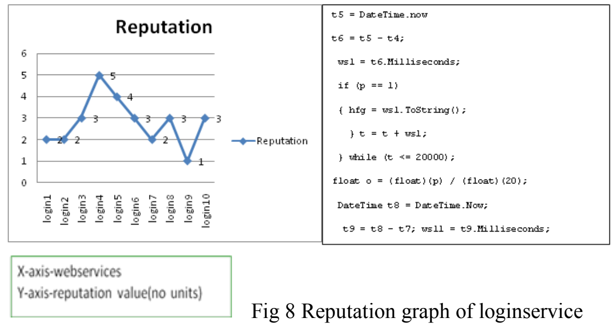

Fig 8 Reputation graph of loginservice

8.1.6 Availability

The following graph shows the variation of availability for various types of login service. It has the service name in X-axis and availability in Y-axis. The web services available at the time of its execution has a value of 1 and the web services which are not available at the time of its execution has a value of 0.

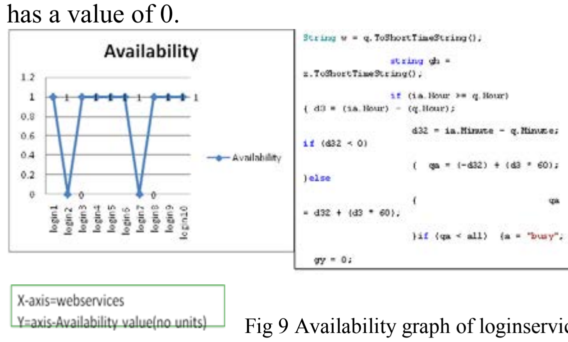

Fig 9 Availability graph of loginservice

8.1.7 Accessibility

The following graph shows the variation of accessibility for various types of login service. It has the service name in X-axis and accessibility in Y-axis. The web services which are unavailable has Zero accessibility value and web services which are available may be accessible.

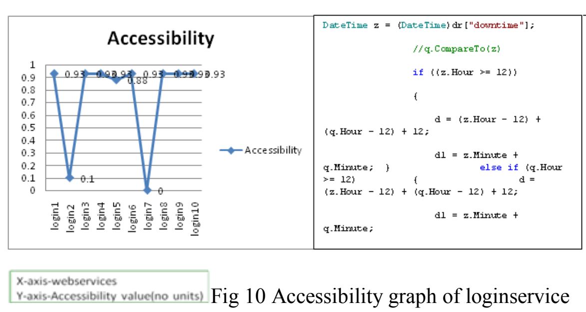

Fig 10 Accessibility graph of loginservice

8.2 Time Based Analysis

The following graph shows the variation of execution time at different instances of display1 service execution. It has different instances of execution in X-axis and execution time (milliseconds) in Y-axis. It is inferred that as the time interval between successive executions is lesser, execution time is also less whereas if time interval is greater, execution time increases. Execution time also depends on cache capacity.

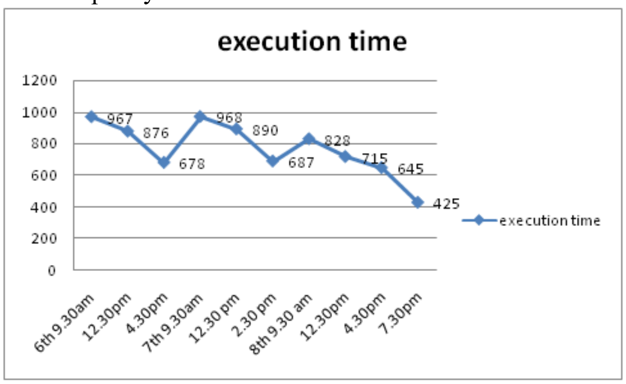

Fig 10 Executiontime graph of display1 service

8.3 Inferences

In general it can be inferred from the analysis that, high performance Web services should provide higher throughput, higher capacity, faster response time, lower latency, and lower execution duration. Here throughput is inversely proportional to execution time and it varies directly with response time. Accessibility of a web service depends on the availability of a service at a given time and it is directly proportional to it. S calability of a web service depends on the response time and the time taken for throughput test.

9. Conclusions

QoS framework for calculating the Quality of Service (QOS) parameters of a w eb service has been developed. These values are represented semantically in ontology using OWL-Qos files that are generated. The case study here is Online Shopping system and the appropriate services for this case study is developed with many alternative services. Whenever the user requests for a service with Qos parameter constraints, the framework will find and rank the relevant services both semantically and also with matches in their quality criteria.

References

[1] Jorge Cardoso, ”On the Move to Semantic Web Services”, World Academy of Science, Engineering and Technology 8 2005
[2] Cardoso.J and Sheth.A, "Introduction to Semantic Web Services and Web Process Composition," Lecture Notes in Computer Science, Springer, 2005
[3] Jorge Cardoso, ”Approaches to Developing Semantic Web Services”, International Journal of Computer Science 1,1, 2006
[4] Amna Eleyan,Liping Zhao, ”Extending WSDL and UDDI with Quality Service Selection Criteria” ,Birzeit University, University of Manchester, Palestine, United Kingdom
[5] D.A.D Mello, V. S. Ananthanarayana ”Quality Driven Web Service Selection and Ranking” In Proceedings of the Fifth International Conference on Information Technology: New Generations. Volume 5, pp. 1175–1176 Chicago, Apr. 2008
[6]Dimitris Plexousakis and Kyriakos Kritikos,” Semantic QoS-based Web Service Discovery Algorithms”, International Journal of Web Services Practices, Vol. 3, No.1-2 (2008), pp. 71-82.
[7] Christian Platzer, Florian Rosenberg,”Bootstrapping Performance and Dependability Attributes of Web Services”,Schahra Dustdar VitaLab, Distributed Systems Group, Information Systems Institute Vienna University of Technology,Austria.

R.Suganyakala and M.Aarthilakshmi are the students of Department of Computer science and Engineering in PSG College of Technology.Their project work includes QOS based webservice discovery and ontologies.

Dr.G.R.Karpagam is professor with 15 years of experience in Department of Computer science and Engineering in PSG College of Technology.She is an IEEE member doing her research in the field of webservices,cloud service discovery,semantic webservices.She has published papers in national/international journal and conferences and reviewer of international journal.She has co-authored monograph for Database Management Systems.She has established industry funded laboratories namely opensource software lab,SOA and cloud computing lab.

Ms.S.Maheswari is a par t-time research scholar pursuing her research in Department of Computer science and E ngineering in PSG College of Technology.Her research focuses on QOS based semantic webservice discovery.She has published paper in international journal and international conference.