Volume 2, Issue 12, December 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Service Discovery Framework with Functional and Non-Functional Information (SDF)

Dr. E. Kirubakaran
AGM, SSTP,
BHEL,
Tiruchirappalli, India,

D. Ravindran
Associate Professor
Dept of C.S.E, St Joseph's College,
Tiruchirappalli, India

Dr. D. I. George Amalarathinam,
Director, MCA,
Jamal Mohamed College,
Tiruchirappalli, India

Abstract— Service Oriented Architecture (SOA) plays an important role in a distributed computing environment based enterprise applications. The processing logic are distributed and implemented with web services. In this scenario, issues are plenty with reference to service providers' interaction with service registry, maintaining information within the registry, searching the registry for service discovery etc This paper proposes a framework, which interacts with service providers, service consumers and service registries. The model proposed will use non-functional data for the process of service discovery and also incorporates consumer's preferences in the process. It also envisages dynamic updation of non-functional information. There are two levels of searches, first one is based on keyword matching and the second level uses the non-functional data and the updation of services' data.

Keywords— Web service, Service Discovery, UDDI, Service Provider, Service Consumer, Service Oriented Architecture (SOA), WSDL

I. INTRODUCTION

In a distributed computing environment, the logical units are deployed into different application servers and they interact with each other, as a single entity, to respond to the clients' requests. Along with data, distributed into different servers, processing logic are also distributed. Component based development is also drafted into the development process in order to gain better independency among the units. These components are implemented as web services, leading to a concept, Service Oriented Architecture (SOA).

Service provider, Service Registry and Service Consumer are the three important entities involved with SOA. Web services are application components of a bigger system and use of such components enhances the distributed system with features like interoperability, reusability, location transparency, scalability, availability, facility for linking Legacy Applications, reduced vendor dependence and loosely coupled architecture. Web services in a distributed environment can be accessed with a help of a service registry feature, called Universal Description Discovery and Integration (UDDI). Service Provider creates a service, creates a service description as a Web Service Description Language (WSDL) document registers the WSDL document in a service registry and service is said to be exposed to the outside world. Now, it is up to the service consumer to search for the service in the registry, get the WSDL document, understand the features of the web service and binds itself to the server and accesses the service.

In this context, discovering the services satisfying clients' request, becomes important. Service discovery mechanism may result in many services because more than one service could satisfy the clients' need. Functional information will be used to discover the services and in order to refine the search, non-functional information may be used. This paper proposes a framework to discover the web services with the help of functional and non-functional information.

Rest of the paper is organized as follows: Section 2 will discuss the related work whereas the Section 3 discusses the framework for service discovery. Section 4 discusses the implementation details of the proposed framework and the results. Section 5 concludes by summarizing the work and opens up the area for further study.

II RELATED WORK

Service Oriented Architecture can be realized with the implementation of web services. Component based development has the problem of objects working within a closed environment and the service technology overcomes this problem, that is, services become totally independent [1]. After the creation of services, service descriptions are created and registered with a service registry. Service registry maintains all information provided in the service description, in the form of WSDL documents. Schahram Dustdar & Martin Treiber [11] proposed peer to peer architecture for the transparent integration of multiple Web service registries and transient Web service providers, based on views. It allows flexible extension of registry entries with value added information without changing the original Web service registry entries.

Once the services are registered with registries, services are said to be exposed to the outside world. Now, clients make requests for service details for accessing a service. Services which will satisfy the requests are to be identified and the service details are to be passed back to the clients. So, service discovery becomes important. Wenge Rong and Kecheng Liu [13], proposed a model for match making based on keywords, syntax, semantics and pragmatics. David Bell et al [2] stressed that the current industrial representation of services is predominantly syntactic and the fundamental

semantic underpinnings are required to fulfill the goals of any semantic-oriented Grid, heterogeneous mix of business software and data. Few semantic web service discovery algorithms based on OWL-S, performs service category matching, service functionality matching and quality of service matching for the discovery of web services [5] [14] [6].

Amin Yousefipour et al proposed a new QoS-aware framework, to improve the semantic Web service discovery, based on broker concept and Zhenqi Wang et al [16], a fully decentralized and interoperable discovery method. Thirumaran et al [9] proposed a Web service discovery architecture using Analytic Hierarchy Process. Demian Antony D'Mello and V. S. Ananthanarayana [3] explored Service Operation Tree (SOT) to store Web service information in a compact way to speed up the discovery process. Yue-an Zhu and Xiao-hua Meng [15] proposed a framework for Service discovery in a pervasive computing environment. The system adopts Three-layer Pervasive Semantic Service Matching Algorithm, which matches service according to category of service, Input/Output parameters and QoS parameters.

Zhu Xilu et al [17] proposed a framework for discovery that support multi-dimension range queries in multiple QoS attribute space, based on distributed decision tree. T. Rajendran and Dr.P. Balasubramanie [10] proposed the agent-based architecture for dynamic Web service discovery which facilitates the requester to specify the QoS requirements along with functional requirements**.** P. Dharanyadevi et al [4] discussed the integration of SOA with the Event Driven Architecture (EDA) in web service discovery and justified by listing total, relevant and non-relevant services, with their implementation. R.Jeberson Retna Raj and T.Sasipraba [7] proposed a model that includes clients' information for discovery. The discovery process quality increases, if the search is incorporated with client's QoS requirements. Uddam CHUKMOL et al [12] proposed a collaborative tagging-based environment for Web service discovery, allowing users to tag or annotate a Web service using keyword or free-text.

III FRAMEWORK FOR SERVICE DISCOVERY

Service discovery is a process of identifying the service descriptions, available in the service registry. Functional information will be used for such identification process and it will be purely a keyword matching process. This discovery process results in selection of many services, of which some may be irrelevant also. To enhance the web service discovery process, if non-functional information also used then the services selected will be meaningful. Many services will be selected because of functional data and these services are filtered and refined afterwards with a help of non-functional data.

This paper proposes a framework for service discovery, which uses functional information and also non-functional information. This framework is depicted in the Figure 1.

Service provider sends the service description to the interface module and the interface sends the functional information to the registries and stores the evaluated or preset non-functional information in a database. When the clients make a request, functional data will be checked first in Level1 and some services are identified. These identified services are refined further with the help of non-functional information in Level2. Only relevant services are identified with non-functional information like reliability, availability, level and access time.

Reliability and availability factors will have values in the range 1 to 5, where 5 stands for the reliability percentage of 80% to 100%, 4 stands for a range of 60% to 80% and so on. Level is used to indicate the level of the service, viz, Level 1 indicates a top level service, whereas Level 2 indicate a slightly lower level service in comparison to level1 service and so on. Response time is also recorded for each service.

This framework takes into account the clients requirements and based on the quality parameters that they send, services are discovered, filtered and refined suitably.

Consider that the non-functional information for a service is coded as S_r (Reliability), S_a (Availability), S_l (Level) and S_{at} (Access time) and is represented as

$$\text{Service S } (S_r, S_a, S_l, S_{at})$$

and the users input also are represented as

$$\text{User U } ((U_r, U_a, U_l, U_{at})$$

After the services are discovered in level1 with functional data, the non-functional information will be compared and the services are filtered based on the condition mentioned as follows:

$$U_r <= S_r;\ U_a <= S_a;\ U_l <= S_l;\ U_{at} >= S_{at}$$

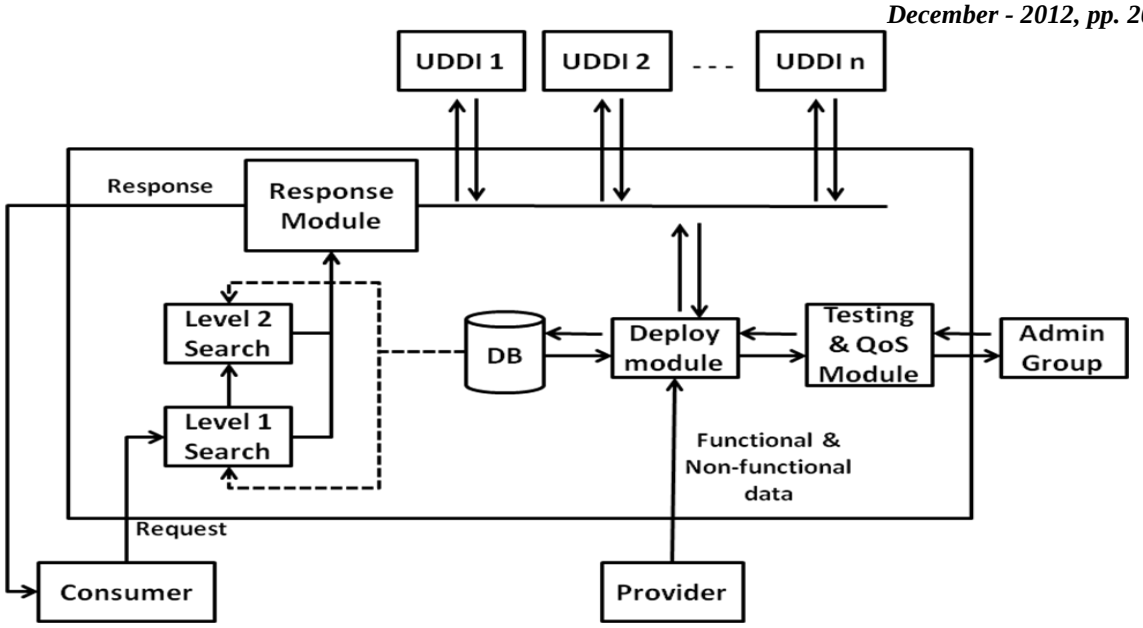

Fig 1 Service Discovery Framework

If the user gives a value 4 for reliability, then the services with a reliability factor of 4 or 5 will be selected. The same constraint is applied to availability and level of the services also. User specifies a minimum reliability and availability parameters that they require and the maximum level that they can afford. Regarding the access time, the time specified is the maximum limit and all the services with lesser or equal response time, will be selected. These non-functional parameters will not be always static, but gets modified after each access.

IV IMPLEMENTATION

The proposed framework has been implemented by creating the services with Java language and deployed within Sun Application Server. The results are summarized in Table 1.

Values specified for the reliability, availability, Level and access time, in the Table 1, are non-functional information given by the client. These values are used to filter the selected services at level1. Total number of services selected at Level 1 and level 2 along with the actual selected services, with service IDs, are also depicted in the Table. Once the access is made, the non-functional parameters are updated for services, which are either accessed or not accessed because of some problem.

Functional information may be stored within single or many registries and that access is not considered in this paper, because the model will specify the exact location of the service descriptions once the services are discovered and refined with level1 and level2 searches.

Table 1 Non-functional data given by clients and services selected

Keywords	Reliability	Availability	Level	Total services (Level 1)	Total services (Level 2)	Access Time	Services at Level 1	Services at Level 2
Gold+price	4	3	3	5	1	28	1 2 4 5 6	1
Gold	4	3	3	3	1	28	1 4 6	1
Price+ Rate	3	3	2	5	1	25	1 2 4 5 6	4
Car	4	4	2	3	0	28	1 5 7	
Room+ Name	2	2	3	1	1	35	3	3
Price+ Rate	3	3	3	5	3	30	1 2 4 5 6	1 4 5
Hotel+Model	4	2	3	2	1	26	5 7	7
Rate	2	2	3	1	1	30	1 2 6	1
Price+ Rate+ Model	1	1	3	6	6	35	1 2 4 5 6 7	1 2 4 5 6 7
Rate	2	2	3	3	0	20	1 2 6	0
Car+ Color	4	4	2	2	0	25	5 7	0

The results are summarized in Figure 2 and Figure 3. Figure 2 shows the non-functional parameters values, excluding access time, and the services selected at level1 (Total1) and level2 (Total2). Figure 3 shows the effect of access time (dotted line) and service selection at level1 (dashed line) and level2 (Solid line).and a screen shot of output is shown in Figure 4.

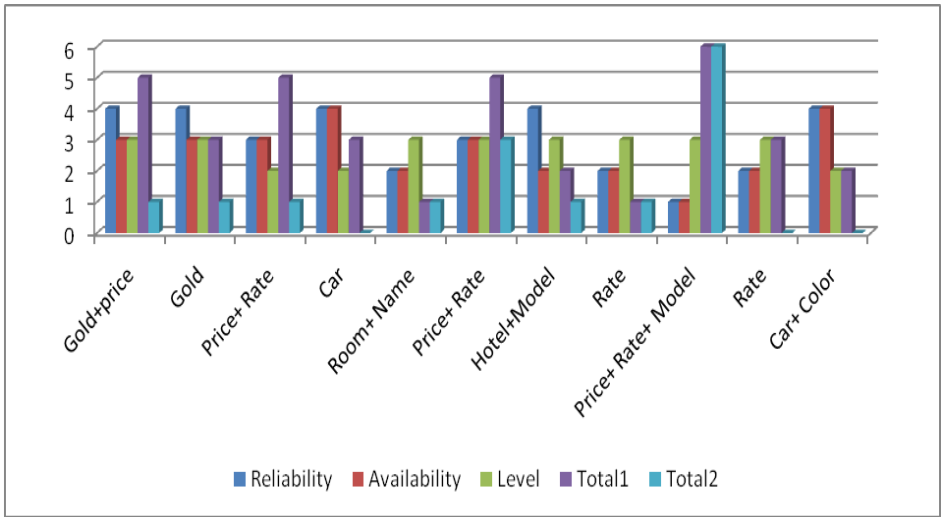

Fig 2 Non-Functional data and service discovery

Based on the details in Figure 2, number of services at Level2 is less than or equal to number of services at Level1. It clearly shows that non-functional information helps in refinement of the search for web services.

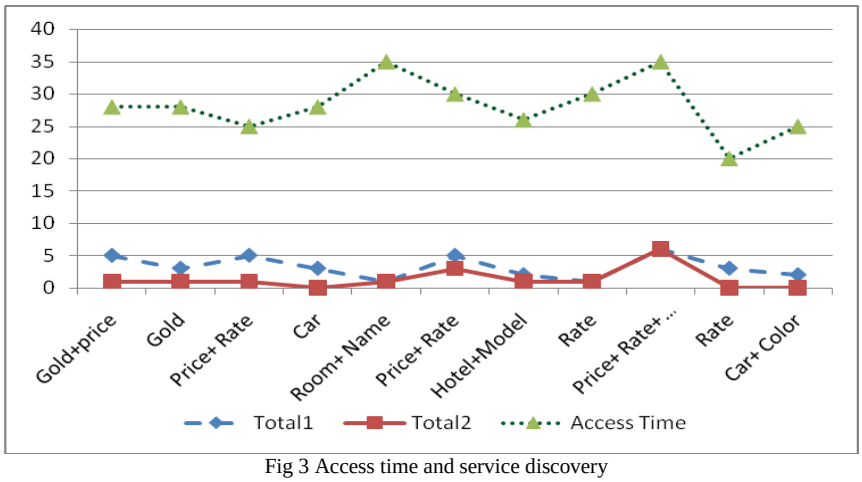

Fig 3 Access time and service discovery

In the Line chart in Figure 3 also, the number of services at level2 is less than or equal to the number of services at level1.

```
Enter your search...   price shirt

Enter the Access Time...   27

Enter the Availability...   5

Enter the Reliablity...   5

Enter the Level...   5

price, shirt,  ¦ 27.0 ¦ 5.0 ¦ 5.0 ¦ 5.0 ¦level 1: 1, 2, 4, 5, 6,  ¦level 2: 1, 4
, 5,

-----------------------Retrieving the Service Details-----------------------

Service ID  :  1     Service Name  :  Service - 1      Location  :  Server-1
Service ID  :  4     Service Name  :  Service - 4      Location  :  Server-4
Service ID  :  5     Service Name  :  Service - 5      Location  :  Server-5
```

Fig 4 output screen shot

The screen shot in Figure 4 displays the input functional and non-functional information, along with the services selected at level1 and level2 and the service details like Service ID, Name and location.

V. CONCLUSION AND FUTURE SCOPE

The service discovery model proposed in this paper consists of interface module interacting with service provider, service consumer and service registries. As the service provider passes functional and non-functional information, the non-functional information is retained in a database and the functional information sent to the service registries. Service consumers are to give functional and non-functional (QoS) data and the discovery was carried out at two different levels, one for keyword matching and the other for non-functional information. When a consumer makes a request, they are supposed to give the QoS parameters also. Once the access is made, the service data will be updated. The feedback from the clients is necessary for the updation process and so this implementation is more suitable in a service composition environment than the open environment. Along with non-functional data used, some more properties also may be considered for better results. QoS weight also could be used in order to give varied importance to the non-functional data.

REFERENCES

[1] Component meets service: what does the mongrel look like?, Bernd J. Krämer, Innovations in Systems and Software Engineering, Volume 4, Issue 4, pp 385-394, 2008.
[2] Enterprise application reuse: Semantic discovery of business grid services, David Bell, Simone, Ludwig, Mark Lycett, Information Technology and Management, Volume 8, Issue 3, pp 223-239, 2007.
[3] A Tree Structure for Efficient Web Service Discovery, D'Mello, D.A. Ananthanarayana, V.S., 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), pp: 826 – 831, 2009.
[4] Event Matchmaking Technique in dynamic web service discovery, Dharanyadevi,P, Dhavachelvan P, Jayakumar S.K.V., Sujatha P, Baskaran R, Venkatachalapathy,, V.S.K, 2011 International Conference on Recent Trends in Information Technology (ICRTIT), pp: 538 – 542, 2011
[5] Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system, Guo Wen-yue; Qu Hai-cheng; ChenHong 2010, The 2nd IEEE International Conference on Information Management and Engineering (ICIME), pp: 601 – 604, 2010.
[6] The research and implementation of Web Service classification and discovery based on semantic, Hang Wu; Chaozhen Guo, 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp: 381 – 385, 2011.
[7] Efficient Web Service Discovery Model Based on QoS and Meta Data Instances, Jeberson Retna Raj, R.; Sasipraba, T, 2011 3rd International Conference on Trendz in Information Sciences and Computing (TISC), pp: 36 – 39, 2011.
[8] Analysis of Various Service Discovery Protocols for Infrastructure-less Networks S. Thangam, Dr.E.Kirubhakaran, International Journal of Computer Applications (0975 – 8887), Volume 12– No.8, December 2010
[9] Parallel Analytic Hierarchy Process for Web Service discovery and composition, Thirumaran, M, Dhavachelvan P, Abarna S, Sheela S, 2011 International Conference on Recent Trends in Information Technology (ICRTIT), pp: 456 – 461, 2011
[10] An optimal agent-based architecture for dynamic Web service discovery with QoS, Rajendran T, Balasubramanie P, 2010 International Conference on Computing Communication and Networking Technologies (ICCCNT), pp: 1 – 7, 2010
[11] View based Integration of heterogeneous Web service Registries – the case of VISR, Schahram Dustdar , Martin Treiber, World Wide Web, pp 457–483, 2006
[12] Enhancing Web Service Discovery by Using Collaborative Tagging System, Chukmol U, Benharkat A, Amghar Y, 4th International Conference on Next Generation Web Services Practices, NWESP '08, pp 54 – 59, 2008
[13] A Survey of Context Aware Web Service Discovery: From User's Perspective, Wenge Rong, Kecheng Liu, 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp: 15 – 22, 2010
[14] A framework and QoS based web services discovery, Yin Baocai, Yang Huirong, Fu Pengbin, Gu Liheng, Liu Mingli, 2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS), pp: 755 – 758, 2010
[15] A Framework for Service Discovery in Pervasive Computing , Yue-an Zhu, Xiao-hua Meng, 2010 2nd International Conference on Information Engineering and Computer Science, ICIECS), Page(s): 1 – 4, 2010
[16] An Approach for Semantic Web Service Discovery Based on P2P Network, Zhenqi Wang, Yuanyuan Hu, 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08, pp: 1 – 4, 2008
[17] QoS-aware web service discovery in P2P network, Zhu Xilu, Wang Bai, Wei Gengyu, 2nd IEEE International Conference on Broadband Network & Multimedia Technology, 2009. IC-BNMT '09, pp: 650 – 654, 2009

