Optimal Selection and Composition of Web Services – a Survey

N. Hema Priya
Asst. Professor & Research Scholar, Department of IT, PSG College of Technology

S. Chandramathi
Phd,Prof & Head, Department of ECE,
Sri Krishna College of Engg & Tech.

ABSTRACT

Internet provides huge number of Web Services and it has become an overhead for the clients to select the appropriate web Service. Different service providers offer the same service with same functionalities, and it is crucial for the users to choose the best. Also some users may need certain service that needs to be combined with already existing atomic services – thus forming a composite service. Many researchers have worked on optimal selection and composition of web services. This paper will provide a clear perspective for the future research.

General Terms

Web Services Composition and selection.

Keywords

Web Services, selection, Composition, techniques, Open source, Survey on optimal selection.

1. INTRODUCTION

To support the changing nature of business environments, SoA (Service oriented Architecture) is widely used for solving many real time problems. The predominant implementation of SoA is Web Services. Web services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. The purpose of a Web service is to provide some functionality on behalf of its owner - a person or organization, such as a business or an individual. The provider entity is the person or organization that provides an appropriate agent to implement a particular service [18]. A requester entity is a person or organization that wishes to make use of a provider entity's Web service.

Web Services follow the fundamental paradigm, publish- find-bind. They are implemented using the standards like UDDI (Universal Description and Discovery Integration), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) etc. Services are made available via the internet by a service provider, and their description is published (using WSDL descriptor files with details stored in UDDI repositories); a service requester will query the UDDI or any middleware agent to find an appropriate service and then use SOAP to invoke that service (Fig. 1).

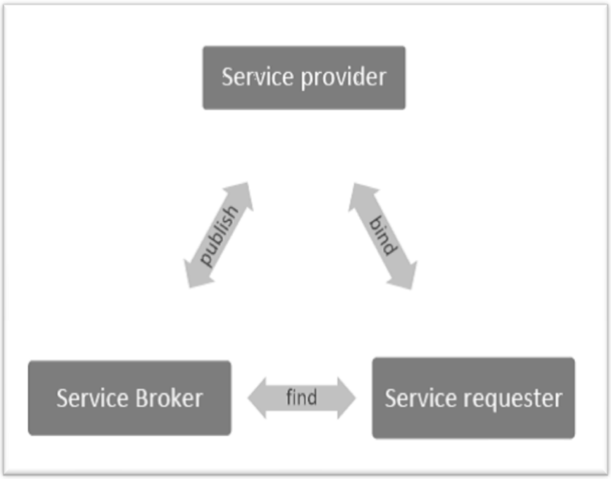

Fig. 1 Web Service Components

The Service requester has to surf any search engine (Eg. Google), a public or a private registry, or a portal and must spend indefinite time to find an appropriate service. The matching service can be found based on either functional or a non functional property. Functional properties (FPs) describe what a service actually does. Based on the capabilities of a web service, it is selected. Non Functional properties (NFPs) of a system are many and varied, including all properties, which are not directly related to the functionality provided. NFPs include quality of service (QoS) as well as other properties such as cost, adherence to standards and obligations on the consumer/provider. QoS is one of the most important subsets of non-functional properties.

In some cases the atomic services may not be sufficient, so there is a need for a composite service. Of course a repository might contain information about the cost of using a service or the service level agreements provided. But, the problem is, the atomic or a composite service searched must be found automatically and it should be optimal, with respect to the client's needs.

There have been a lot of researchers worked on this automated and optimal selection and composition of Web Services. They have employed various algorithms and models. In the rest of the paper we explore and compare the various non semantic based service selection and composition techniques and a clear view is provided.

2. NFP BASED SELECTION

Non-functional properties (NFP) play an important role in all service related tasks, especially in discovery, selection and substitution of services. It is simple to imagine a scenario in which multiple services which provide the same functionality can fulfill a user request. In this case the ability of the user to differentiate between the services depends upon their non-functional properties.

2.1 Algorithm based Selection

Optimal selection and composition is done using evolutionary and non-evolutionary algorithms. Minghui wu et al. in [4] constructed a mathematical model for web service composition and selection. According to these models, service composition problem is considered as Single Objective Multi Constraints Optimization problem. For this an evolutionary algorithm is proposed, GAELS (Genetic Algorithm Embedded Local Searching), which uses the strategies of enhanced initial population and mutation with local searching, to speed up the convergence.

In [17] a tool for Web Service selection and composition is proposed for ubiquitous computing environment. It evaluates quality in three dimensions: quality of services, quality of contexts and quality of devices. It also introduces Event driven Web service Composition (EWC) which automatically prioritizes services, contexts and devices, selects optimal service, monitors and reconfigures the services.

An evolutionary based Bee Colony Optimization Metaheuristic (BCO) is proposed in [4], for local optimization of QoS. The BCO is capable to solve deterministic combinatorial problems, as well as combinatorial problems characterized by uncertainty. Bee swarm behavior in nature is, first and foremost, characterized by autonomy and distributed functioning and self-organizing. Agents – bees collaborate in order to solve difficult combinatorial optimization problem. All bees are located in the hive at the beginning of the search process. During the search process, the bees communicate directly. Each bee makes a series of local moves, and in this way incrementally constructs a solution of the problem. Bees are adding solution components to the current partial solution until they create one or more feasible solutions. The search process is composed of iterations.

Angus F.M. Huang et al. [2] proposed a non-evolutionary mathematical programming based, solution-integer programming for selection and composition of web service. Also an algorithm is designed for efficient matchmaking.

2.2 Extended Architecture

Shrabani Mallick et al [7] proposed an architecture based on x-SOA, that organizes the method of web service discovery in a structured manner, using an intermediary requester friendly layer called Request Analyzer (RA) between the service provider and service requester via a service broker. Also an algorithm is proposed for a complete cycle of web service discovery. A cache based service broker has been proposed that takes lesser time towards discovery path.

Mohamed Adel serhani et al [8] proposed a Community based web service selection, by adding a new support layer for QoS classification. It is done through QoS specification, monitoring and adaptation of web services within the community. A web service will be admitted into the community if QoS is high or rejected if QoS is low. The main focus is on the managerial community to adapt QoS based selection.

There has been a huge work done on agent based discovery of web services. In [11] S.Susila et al. proposed such a scheme, which selects exact web service based on the information from WSDL files. Data mining technique (WEKA) is used here.

In [16] G.Vadivelou et al. proposed a multi layer composition model based on multi agent system. Two layers are added between the service requirement and service candidates. Home agent, Service agent and coordinating agents are used. Different self adaptive mechanisms are proposed corresponding to different environment's evolutions to improve the reliability of web service compositions. Also, an algorithm for dynamic selection based on QoS and composability of participated services. According to the implementation results, better performance is shown.

2.3 Control Flow based Selection

In [10], Chengying Mao et al. adopted a complexity measurement technique based on Petri Nets for web service composition. Based on business process representation, two metric sets are provided through analyzing the compositions' execution logics and dependency relations in workflow. The first one is count based metric set and the second one is execution path based metric set. Then an extension based on cognitive informatics is also discussed.

Petri nets based work flow for describing business solutions is given in [2].

2.4 Reputation aware Selection

In [13] Shenghui Zhao et al. proposed a gradually adjusting reputation evaluation method based on eliminating the collusive behaviors of consumers and a model for service selection is designed. In order to adjust reputations, QoS similarity is computed first according to the difference between promised QoS and delivered QoS. Then reputation is calculated by mining the collusive users. Association rule mining is used here. The experimental results clearly identified malicious customers and thus based on reputation, better services are selected.

3. SELECTION BASED ON USER RATINGS / TAGS

Web services by tags are the best way to find the appropriate web service. In [14] Uddam Chukmol et al. illustrated a collaborative tagging-based environment for web service discovery, allowing users to tag or annotate a service using keyword or free text. Algorithm is developed for both types of tagging, and yet to be implemented. In [6], the responses of two different approaches were evaluated by an experimental setup. The design approach to implement the database consists of tables which contains web services that are already tagged by the web service providers. Based on the tags inputted by the user, the search engine implements the algorithms to find the semantic correlation between the service tags. By applying SEBT (Search Expansion Based on Tags) algorithm, web services which are semantic correlative are obtained. However, it can be observed that some of these web services semantic relationships are poor. By applying

SPBT (Search Pruning Based on Tags) algorithm, web services whose semantic relationships are close, can be obtained.

The sequence is as follows: 1) the client searches the web service by putting the tag in the user interface. 2) The user interface of the system passes the tag to the search engine where the logic SEBT and SPBT is executed. 3) The matched web services are retrieved from the database and are presented to the client via the interface. Results are obtained with varying thresholds and the results obtained by SEBT and SPBT are compared.

In [12], they contributed an algorithm that features the expansion of service requests by user defined demands. The user’s profile is divided into three parts: (i) to decompose goals based on typical usage patterns anticipated by service provider – break down a complex task into simpler goals (ii) user preferred service parameters (cost), (iii) information that a service claims to provide.

In [15] Delnavaz Mobedpour et al. proposed a expressive QoS Query Language, used for service selection. A few unique features of the query language includes time dimension, user defined relaxation order (different from preference order) and support for mixed fuzzy and range requirement. The query is formulated based on user input, and it is sent to the selection and ranking component, finally yielding in a ranked list of matching services.

In [1], to solve problems proposed by keyword based search techniques, WSRF (Web Service Relevance Function) for measuring the relevancy ranking of a Web Service is proposed. In order to provide quality driven ranking, the model uses a crawler engine. The model enables users to search and manage criteria based on their interest. Highest ranked service is considered the most relevant as per the user’s interest. This reduces Cost of service selection.

4. SELECTION BASED ON BEHAVIORAL PROPERTIES

In [3], Zhongnan Shen et al proposed a new behavior model for web services using automata and logic formalisms. Roughly, the model associates messages with activities and adopts the IOPR model in OWL-S to describe activities. A new query language is developed to express temporal and semantic properties on service behaviors. Query evaluation algorithms are developed; in particular, an optimization approach using RE-tree and heuristics is shown to improve the performance. Specifically, experimental results show that the use of RE-tree reduces query evaluation time by an order of magnitude and with heuristics it enhances the performance by two orders of magnitude.

Table 1: A Survey of Web Service Selection Techniques

S. No	**Approach**	**Benefits**	**Pitfalls**
NFP based Selection			
1.	Algorithm based Selection		
	[4] GAELS	Better execution time and selection	Not all QoS parameters are taken into account
	[4]BCO	Better performance	Lacks interoperability
2	Extended Architecture		
	[7] xSoA	Helps even a novice user	Not all QoS parameters are taken into account
	[8], [10], [16] Agent based, Community based	Better throughput, Lesser resource consumption	Not all QoS parameters are taken into account
3	Control Flow		
	[2], [10] Petri Nets	Easy interpretation, better performance	Increase in cost if implemented vastly
4	Reputation based		
	[13]Eliminates collusive users	User friendly, Better performance	Lacks interoperability
User Interest / Tags based Selection			
5	[6],[14] Collaborative tagging	User friendly	Creates ambiguity, Lacks implementation
6	[12] Users’ Profile	Easy to use	Lacks interoperability
7	[15] Query Language – QQL	User friendly	Fully functioning model is not there
8	[1] WSRF	Lesser search Cost	Lacks interoperability
Behavior based Selection			
9	[3] Query Language, RE tree based optimization	Performance is good	Lacks interoperability

6. A DISCUSSION ON FUTURE WEB SERVICE SELECTION

Most techniques discussed deploy proprietary, commercial tools for developing the software, which resulted in increased cost and a compromise in interoperability, if real time applications are considered. For that a few possible low cost open source tools [19] that can positively show a better interoperability is given in Table 2

TABLE 2

S.No	Components	Open Source Tools	Feature
1.	Registry	UDDI	A private or a public directory
		jUDDI	UDDI for java
		UDDI4J	Java class library that provides an API to interact with a UDDI
2.	SOAP	Axis	3rd generation of Apache SOAP (which began at IBM as "SOAP4J").
3.	Invocation	WSIF	A simple Java API for invoking Web services, no matter how or where the services are provided
		Jersey	For building RESTful Web services.
4.	Crawler	WebSPHINX	A Java class library and interactive development environment for Web crawlers that browse and process Web pages automatically.
		Crawler4j	It has been able to download and parse 200 pages per second on a Quad core PC with cable connection

7. CONCLUSION

Different types of algorithms, architectures, user specific techniques employed in optimal selection and composition of Web Services are discussed. The selection is based on functional, non functional and behavioral properties. Most cases did not consider all the major QoS criteria and either cost or interoperability is compromised. And most important thing is that the operating system targeted is not interoperable. Instead, an open source Operating system can be preferred along with its supportive tools (Table 2). As the number of Web service providers and consumers are evolving with modern technologies, considering all the major quality metrics, an interoperable, user friendly and cost effective tool helps better and provides faster selection.

8. REFERENCES

[1] Eyhab al-Masri, Qusay H.Mahmoud, “QoS based Discovery and Ranking of Web Services,”, 16th International Conference on Computer Communications and Networks, ICCCN, IEEE 2007.

[2] Angus. F.M. Huang, Ci-Wei-Lan, Stephen J.H. Yang, “An Optimal QoS based Web Service Selection scheme”, Science Direct, Information Sciences 179 (2009).

[3] Zhongnan Shen, Jianwen Su, “Web Service Discovery Based on Behavior Signatures”, IEEE International Conference on Services Computing – Vol. 01, 2005.

[4] Minghui Wu, Xianghui Xiong, Jing Ying, Canghong Jin, “QoS Driven Global Optimization Approach for Large Scale Web Services Composition”, Journal of Computers, Vol. 6,No.7,July 2011.

[5] D.Palanikkumar,” An Algorithmic Evaluation of Optimal ServiceSelection using BCO” European Journal of Scientific Research, vol.68, No 4, 2012.

[6] Kadambari Dehekar, Amiya Kumar Tripathy, “Discovering Web Services by Search Expansion and Search Pruning,” International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 3, No. 1, February 2012, ISSN: 2079-2557.

[7] Shrabani Mallick, D.S.Kushwala. “An Efficient Web Service Discovery Architecture”, International Journal of Computer Applications (0975-8887), vol. 3, No.12, July 2010.

[8] Mohamed Adel Serhani, Abdelghani Benharref, “Enforcing Quality of Service within Web Services Communities,” Journal of software, Vol. 6, No.4, April. 2011.

[9] Zhongnan Shen, Jianwen Su, “Web Service Discovery Based on Behavior Signatures”, University of California, for NSF Grant.

[10] Chenying Mao, “Control Flow Complexity Metrics for Petri Net based Web Service Composition”, Journal of Software, vol. 5, No.11, Nov. 2010.

[11] S.Susila, S.Vadivel, “Agent based discovery of web service to enhance the quality of web service selection,” International Journal of Computer Science and Network Security, vol. 11, No.2, Feb. 2011.

[12] Shailesh Kapre, D.Chandramohan, “Personalized Web Service Selection”, International Journal of Web & Semantic Technology, Vol.2, No.2, April 2011.

[13] Shengui Zhao, Gioxin Wu, Guilin Chen, Haibao Chen, “Reputation Aware Service Selection based on QoS similarirty” Journal of Networks, Academy publisher 2011.

[14] Uddam Chukmol, Alcha Nabiala, Benharkat, Youssef Amghar, “Enhancing Web Service Discovery by using Collaborative Tagging System,” International Journal of Web Service Practices, vol. 3, No 3-4 (2008), pp. 129-135.

[15] Delnavaz Mobedpour, Chen Ding, Chi-Hung Chi, “A QoS Query language for User-Centric Web Service Selection,” IEEE International Conference on Services Computing, 2010.

[16] G.Vadivelou, “Algorithm for Web Service Composition using multi agents” International Journal of Computer Applications, vol. 13, No.8, January 2011.

[17] Heejung Chang, Kangsun Lee, “Quality driven Web Service Composition for Ubiquitous Computing Environment”, International Conference on New Trends in Information and Service Science, IEEE Computer Society, 2009.

[18] www.W3.org.

[19] www.java-source.net/open source web-services tools in java.