IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Volume-4, Issue-8 E-ISSN: 2347-2693

Survey on Web Compositioon Based on Semantics

G. Narayanan[1*] and Dr. Pon Periasamy[2]

PG and Research Department of computer science,
Nehru Memorial college (Autonomous), Puthanampatti 621 007
Trichy, Tamil nadu, India

Available online at: www.ijcseonline.org

Received: 22/Jul/2016 Revised: 30/Jul/2016 Accepted: 18/Aug/2016 Published: 31/Aug/2016

Abstract— The current service composition techniques and tools are mainly designed for use by Service-Oriented Architecture (SOA) professionals to solve business problems. Little attention has been paid to allowing end-users without sufficient service composition skills to compose services and integrate SOA In this paper provides a brief survey of the approaches to semantic integration developed by researchers in the ontology community. The system focus on the approaches that differentiate the ontology research from other related areas. The goal of the paper is to provide a reader who may not be very familiar with ontology research with introduction to major themes in this research and with pointers to different research projects. We discuss techniques for finding correspondences between ontologies, declarative ways of representing these correspondences, and use of these correspondences in various semantic-integration tasks

Keywords- Ontology, Service-oriented architecture, Service composition, Service discovery and Web services

I. INTRODUCTION

In recent years, the software industry has been dominated by the web services. Most of the enterprises publish their applications on the World Wide Web using web services. Web service technology is the heart of Service Oriented Architecture (SOA) that meets the interoperability demands of the web application. W3C defines web service as “a software system designed to support interoperable machine to machine interaction over a network.” Web services [K. Kritikos, 2009] are self-contained, self-describing, and loosely coupled software applications that can be published, located, and accessed across the web using XML-based open standards, namely, SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), and UDDI (Universal Description, Discovery and Integration).

The three main components in web service architecture are web service provider, web service requester, and UDDI registry [B. Hong, 2011] as given in Figure 1. The web service provider publishes information about the web services in UDDI registry in the form of WSDL file. WSDL of a web service is a machine-readable description that indicates the service invocation parameters of the web service. WSDL is an XML-based language that describes the functionality of the web service like location of the service and operations (or methods) of the service. Universal Description, Discovery and Integration (UDDI) is a platform-independent, Extensible Markup Language- (XML-) based registry for businesses worldwide to list themselves on the Internet, used to register and locate web service applications. Web service consumer uses UDDI to discover appropriate services which meet the requirement using the service invocation parameters provided by the web service provider.

Heterogenous Web resources hinder search engines and users to discover suitable Web resources for fulfilling users’ goal of daily activities. Existing Web resources are described in heterogeneous formats. For example, Web Service Description Language (WSDL) [chinnachi, 2007] is used to describe SOAP-based Web services that make remote procedure calls. WSDL is designed for programmers. It is difficult to be understood by non-IT professional users. HTTP-based APIs are increasingly used by companies in [Carlson, 2008]. They are chosen over SOAP-based services. However, HTTP-based APIs reveal little information about the functionality of the APIs. It is challenging for existing search engines such as Google to discover the HTTP-based APIs due to the lack of functional descriptions.

II. RELATED WORK

In today’s on-line experience, an end-user, who is not familiar with Web services standards and tools, frequently re-visits Web sites and uses online services to perform repeated activities, such as online shopping. The end-user potentially composes an ad-hoc process to fulfil his or her needs. Such an ad-hoc process is characterized by a set of tasks performed by end-users without a strict execution order. For example, planning a trip is an ad-hoc process for many end-users. It involves several tasks, such as searching for flight tickets, booking a hotel, and checking the weather reports for the destination. These tasks can be performed in any order to achieve the goal of trip planning. Currently, an end-user needs to manually browse different Web services, which provides specialized services to perform each of the tasks in order to plan a trip. For example, expedia.com is a specialized website for booking hotels and checking flight tickets.

In (Ran 2003) the main idea is to include a QoS model into UDDI registries so that QoS parameters can be included as search criteria. In fact, they propose to use a QoS model as non-functional requirements to enable a service search based on functional and non-functional (QoS) parameters. They also explain that the current UDDI model limits the service discovery to functional requirements. Due to this limitation, they propose to incorporate a QoS model into UDDI registries. The proposed model will coexist with the current UDDI. If no services are found with these qualities, feed-back is returned to clients and so they can reduce their quality values.

In (Sreenath and Singh 2004) the authors propose a mutual evaluation process between agents to select a web service. It selects the best service based on rates given to providers by agents. A provider is ranked by an agent and the agent's evaluations are, themselves, evaluated by other agents. Thus, selecting a service provider involves getting a list of rated service providers and choosing the best based on a weighted average calculation. The result of the execution of the chosen service is then feedback into the service provider rating mechanism. The main idea in (Cardoso et al. 2004) is an adaptation of Workflow Quality of Services and its transposition to web service technologies. First of all, they propose to characterize workflows based on their QoS in order to better fulfill customers' expectations. The QoS model is composed of: time, cost, fidelity and reliability. Fidelity means how well workflows, instances and tasks are meeting user specifications. Concerning reliability, it is the measure of the likelihood that the component performs a task demanded by a user. These QoS constraints are implemented into METEOR workflow management systems for Genomic Projects.

Ideas in (Zeng et al. 2003) are very close to our proposition regarding the QoS model and also to the resolution method. This work treats the services selection during the execution process and so it takes into account multiple criteria. Thus, the idea is that services are selected by the composite service execution engine based on a set of criteria. This paper presents a quality model that is characterized by non-functional properties: price, duration, reputation and availability. Service selection is then formulated as an optimization problem and a linear programming method is used to compute optimal services execution plans to compose services. This work is an example of objective aggregation approach. In other words, they weight the objectives and then sum them all in order to create a single aggregate objective. The transformed problem is solved using linear programming. Notice that this approach cannot lead to alternative solutions and is not able to handle automatically non-linear constraints.

The most important difference between our work and Zeng et al's work (Zeng et al. 2003) is that, as opposed to their work, we do not give any weight to any objective. We treat all objectives with the same importance using a multi objective optimization approach. Even though our objectives are contradictory, they are taken into account simultaneously by our resolution algorithm. Web resources for fulfilling a user's activity are often distributed in several websites. In today's online experience, users frequently re-visit Web resources distributed across different websites to perform repeated tasks. For example, a person planning a conference trip needs to locate various Web resources to search for transportation, reserve accommodation, and look for local attractions. In current practices, users have to visit multiple websites to find the desired Web resources. The visited Web resources are not recorded. Therefore, users cannot automatically reuse the already performed process for a recurring activity. It is a time-consuming process to manually compose Web resources. The result of service composition may not provide the optimal outcome. For instance, a user may not be able to discover a Web service that provides the most economical air ticket.

III. QOS-BASED WEB SERVICE SELECTION METHOD

Some of the current approaches for composition first rank the web services before selection. But the performance of this approach is less when compared to composition approaches that does ranking followed by selection. In [Shaozhong, 2009], QoS-based web service selection model is discussed. This method is based on weightage and normalization of functionally similar services. Though this method gets the client's weightage, there is a large difference between the QoS values of a group of functionally similar services and QoS value of another functionally similar service. During normalization the difference between the normalized QoS values becomes negligible. Thus user's weightage does not influence the QoS values while ranking the web services.

Other normalization approach [. C.-F. Lin,2011] used for web service selection considers only QoS of the services without users QoS requirements and preferences on QoS aspects. But this approach handles the situation when there is no feasible solution to fulfill QoS constraints set by users. QoS-based web service selection method proposed in [W. Rong, 2009] requires a lot of interactions with users and does not consider user preferences on various concerned QoS aspects.

Web service composition [E. Bertin, 2009] approaches can be classified into manual composition or static composition, semiautomatic service composition and automated web service composition. In manual composition, composition is achieved programmatically through orchestration languages like Web Services Business Process Execution Language (WS-BPEL). Manual composition of services consumes a lot of time and it cannot adapt to the dynamic environment. Semiautomatic service composition involves the end user in the management of composition through a graphical interface, for example, YAHOO PIPES. This kind of approach also requires users with knowledge in the development and consumes a lot of time.

In automatic web service composition, the system processes the user's request and generates the composite service. This paper focuses on automatic web service composition considering functional and non-functional aspect of web services. Advantages of using automatic web service composition approach are that it(i)minimizes user intervention,(ii)accelerates the process of producing a composite service that satisfies the user request,(iii)eliminates the human errors,(iv)reduces cost.Four distinct approaches for automated web service composition are workflow based, model-based, mathematics-based, and AI planning approaches. Some of the other approaches include heuristic approaches which use algorithms like genetic algorithm.

Many automatic approaches do not take QoS attributes into account. Gu et al. [Z. Gu, 2008] proposed an approach without considering QoS aspects and it lacks correctness in composition. The workflow approach proposed by Ardagna et al. [D. Ardagna, 2007] focused on adaptation and flexibility of service composition modeled as business processes. BPEL process is created and then annotated with global and local constraints. The QoS constraints are expressed in the Service Level Agreement (SLA). WSQosX proposed in [M. Spahn,2006] is a workflow engine which calculates an execution plan that maximizes the overall QoS. The main limitation is that many candidate web services for the composition are selected which reduces the efficiency of the approach. Workflows are limited to simple schemas and some of the approaches are manual.

IV. MODEL-BASED QOS APPROACHES

Model-based approaches include [D. Berardi, 2005]. The model-based approach proposed in [12] uses UML activity diagrams to model service compositions. The UML diagrams are then used to generate executable BPEL processes using XSLT transformations. Model-based approaches are time-consuming and not fully automated.

Some of the mathematics-based approaches are [D. Skogan, 2004]. The QoS aware service composition problem has been defined as mathematical problems such as integer linear planning, single objective problem with QoS constraints, and the multiple objective problem with QoS constraints which are the most common ones. For the service composition, LIP can help to get the best solution without constructing all the possible composite services. However, it can only be available for the composition problems with small volumes since the traditional branch and bound technique has its computation limitations. On the other hand, it also asks for the linearization of the objective function and the corresponding constraints.

The AI based approach proposed in [I.-H. Kwon, 2008] is complex and some of the other approaches based on "AI planning" will not endure when any one of the web service within the composition plan fails.

PSR system (precomputing solutions for web service composition in an RDBMS) proposed by Lee et al. [B. Hong, 2011] implemented web service composition using a relational database. This approach is efficient in a large number of services, but it does not consider QoS aspects of composition.

These shortcomings motivated us to build QoS aware automatic web service composition framework using a ontology concepts on semantic web. Our proposed framework is flexible in supporting user preferences over QoS criteria and also allows user to specify the constraints.

V. PROBLEM MODEL

Many authors have studied the problem of web services composition, but only a few have worried about how complex this composition could be. Concerning our Travel problem, consider that we can now have more than ten tasks to be executed and over a hundred candidate services; with the daily growth of the Internet, these figures may soon be realistic. Thus, combining each task, respecting their restrictions and respectively finding the service to execute the tasks can be considered as a combinatory problem. Since we treat our services composition as a combinatory problem it requires optimization, so our Travel problem can be treated as an optimization problem. Optimization problems require basically two elements: a search space composed of potential solutions and an objective function to be optimized. The search space may be restricted by a set of constraints. In our example, prior to execute the services, it is necessary to find optimal composition. In order to achieve optimal compositions we defined four main objectives that should be optimized: cost, time, reputation and availability. In addition to these objectives, we restricted the search space using constraints stating, for example, that one service can only be allocated to one task. Actually these objectives are our QoS model explained earlier. Since each QoS variable will be described inside a service, our optimization problem will retrieve these values in order to make possible combinations. The QoS (non-functional criteria) model was used as the objectives to be optimized because we need to differentiate candidate services with identical functionalities.

VI. CONCLUSION

In the framework they are addresses the challenge issues web service composition and QoS aware automatic web service composition framework which is flexible in satisfying multiple QoS requirements and also considers the user preferences. User can specify his preferences over QoS parameters which makes the web service composition more flexible. In this approach dynamically composes web services and the composition plans are generated automatically. They have considered six QoS criteria and also considered user's constraints over these parameters. Our system also allows user to provide feedback after composition which updates the reputation of those services immediately. Evaluated our algorithm based on the computation time for ranking and composition based on different number of service providers and number of tasks. Our experiments showed that our QAWSC composition framework yields lower execution time and supports user preferences and In the future, We plan to test model proposed framework using standard datasets like WS-Challenge datasets.

VII. REFERENCE

[1]. K. Kritikos and D. Plexousakis, "Requirements for QoS-based Web service description and discovery," IEEE Transactions on Services Computing, vol. 2, no. 4, pp. 320–337, 2009. View at Publisher · View at Google Scholar · View at Scopus

[2]. D. Lee, J. Kwon, S. Lee, S. Park, and B. Hong, "Scalable and efficient web services composition based on a relational database," Journal of Systems and So□ware, vol. 84, no. 12, pp. 2139–2155, 2011. View at Publisher · View at Google Scholar

[3]. L. Sha, G. Shaozhong, C. Xin, and L. Mingjing, "A QoS based web service selection model," in Proceedings of the International Forum on Information Technology and Applications (IFITA '09), pp. 353–356, May 2009. View at Publisher · View at Google Scholar · View at Scopus

[4]. C.-F. Lin, R.-K. Sheu, Y.-S. Chang, and S.-M. Yuan, "A relaxable service selection algorithm for QoS-based web service composition," Information and So□ware Technology, vol. 53, no. 12, pp. 1370–1381, 2011. View at Publisher · View at Google Scholar · View at Scopus

[5]. W. Rong, K. Liu, and L. Liang, "Personalized web service ranking via user group combining association rule," in Proceedings of the IEEE International Conference on Web Services (ICWS '09), pp. 445–452, July 2009. View at Publisher · View at Google Scholar · View at Scopus

[6]. N. Laga, E. Bertin, and N. Crespi, "User-centric services and service composition, a survey," in Proceedings of the 32nd Annual IEEE So aware Engineering Workshop (SEW '08), pp. 3–9, November 2009. View at Publisher · View at Google Scholar · View at Scopus

[7]. Z. Gu, J. Li, and B. Xu, "Automatic service composition based on enhanced service dependency graph," in Proceedings of the IEEEInternational Conference on Web Services (ICWS '08), pp. 246–253, IEEE, Beijing, China, September 2008. View at Publisher · View at Google Scholar · View at Scopus

[8]. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, and P. Plebani, "PAWS: a framework for executing adaptive web-service processes," IEEE So aware, vol. 24, no. 6, pp. 39–46, 2007. View at Publisher · View at Google Scholar · View at Scopus

[9]. R. Berbner, M. Spahn, N. Repp, O. Heckmann, and R. Steinmetz, "Heuristics for QoS-aware web service composition," in Proceedings of theIEEE International Conference on Web Services (ICWS '06), pp. 72–79, September 2006. View at Publisher · View at Google Scholar · View at Scopus

[10]. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic service composition based on behavioral descriptions," International Journal of Cooperative Information Systems, vol. 14, no. 4, pp. 333–376, 2005. View at Publisher · View at Google Scholar · View at Scopus

[11]. R. Hamadi and B. Benatallah, "A petri net-based model for web service composition," in Proceedings of the 14th Australasian Database Conference (ADC '03), pp. 191–200, Adelaide, Australia, February 2003.

[12]. D. Skogan, R. Gronmo, and I. Solheim, "Web service composition in uml," in Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC '04), pp. 47–57, IEEE, September 2004. View at Publisher · View at Google Scholar

[13]. L.-Z. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality driven web services composition," in Proceedings of the 12th International Conference on World Wide Web (WWW '03), pp. 411–421, ACM, May 2003. View at Publisher · View at Google Scholar · View at Scopus

[14]. M. Zhenhua, C. Hongming, and J. Lihong, "Service selection problem with multiple QoS constraints based on genetic algorithm," Computer Applications and So□ware, 2009.

[15]. B.-Y. Wu, C.-H. Chi, S.-J. Xu, M. Gu, and J.-G. Sun, "QoS requirement generation and algorithm selection for composite

service based on reference vector," Journal of Computer Science and Technology, vol. 24, no. 2, pp. 357–372, 2009. View at Publisher · View at Google Scholar · View at Scopus 7/29/2016 An Automatic Web Service Composition Framework Using QoSBased Web Service Ranking Algorithmhttp://www.

[16]. J. M. Ko, C. O. Kim, and I.-H. Kwon, "Quality-of-service oriented web service composition algorithm and planning architecture," Journal of Systems and So□ware, vol. 81, no. 11, pp. 2079–2090, 2008. View at Publisher · View at Google Scholar · View at Scopus

[17]. L. Zhao, Y. Ren, M. Li, and K. Sakurai, "Flexible service selection with user-speci_c QoS support in service-oriented architecture," Journal of Network and Computer Applications, vol. 35, no. 3, pp. 962–973, 2012. View at Publisher · View at Google Scholar · View at Scopus

[18]. R. Chinnici, J. Moreau, A. Ryman, S. Weerawarana, "Web Service Description Language", W3C Recommendation (2007) 8. M. P. Carlson, A H. H. Ngu, R. M. Podorozhny, L. Zeng, "Automatic Mash Up of Composite Applications," International Conference on Service Oriented Computing (ICSOC) 2008, Sydney, Australia, December 1-5, 2008, pages: 317-330

[19]. C. Engelke and C. Fitzgerald, "Replacing Legacy Web Services with RESTful Services," WS-REST 2010 First International Workshop on RESTful Design

[20]. R. Ennals and D. Gay. "Building Mashups by Example", Proceedings of IUI (2008)

[21]. R. J. Ennals, M. N. Garofalakis, "MashMaker: mashups for the masses," Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM.