

Utilizing Linked Open Data for Web Service Selection and Composition to Support e-Commerce Transactions

Vesyropoulos Nikolaos¹, Georgiadis K. Christos¹, and Pimenidis Elias²

¹ University of Macedonia, Thessaloniki, Greece
{nvesyrop,geor}@uom.edu.gr

² University of the West of England, Bristol, UK
elias.pimenidis@uwe.ac.uk

Abstract. Web Services (WS) have emerged during the past decades as a means for loosely coupled distributed systems to interact and communicate. Nevertheless, the abundance of services that can be retrieved online, often providing similar functionalities, can raise questions regarding the selection of the optimal service to be included in a value added composition. We propose a framework for the selection and composition of WS utilizing Linked open Data (LoD). The proposed method is based on RDF triples describing the functional and non-functional characteristics of WS. We aim at the optimal composition of services as a result of specific SPARQL queries and personalized weights for QoS criteria. Finally we utilize an approach based on the particle swarm optimization (PSO) method for the ranking of returned services.

Keywords: Web Services · Linked Open Data · RDF

1 Introduction

The Service Oriented Architecture (SOA) paradigm provides the means for the removal of interoperability barriers between heterogeneous systems. As a result, it has been widely adopted by enterprises for the fulfillment of their business needs. WS can both be used as stand-alone components or as parts of a value added composition, comprising of a number of (internal or external to the enterprise) services. Through the reuse of existing WS as parts of a composition, an enterprise can lower the cost of complex operations and adjust to specific business needs that rise during online transactions with partners or customers [8].

For the discovery and composition of WS, semantic rules and ontologies have widely been used over the past few years. Through the application of semantic technologies, metadata describing the functionality and properties of WS are utilized for the automated processing of orchestration rules, thus resulting in the rapid development of compositions.

A recent approach towards the identification of WS which fulfill specific requirements, as well as their selection and composition is the use of LoD [7]. This method, which has been widely adopted in various fields, refers to the utilization of structured data that can be machine understandable [13]. An important aspect of LoD is that data share connections, based on semantic rules. Through these rules linked data form RDF graphs, representing the ties between the interlinked data. Such data can be identified and retrieved by issuing specific queries through an appropriate querying language, such as SPARQL.

In this work, we describe a proposed framework for the selection and composition of WS. The main contribution of our approach lies within the architecture of the theoretical framework that utilizes RDF-based repositories, combined with a novel representation of WS related data as RDF graphs. In addition, we demonstrate a methodology for integrating a performance-based ranking algorithm, as a step towards the automation of WS compositions.

This work is organized as follows: In chapter 2, we analyze related works and attempt to highlight the benefits of utilizing LoD for the optimal WS selection and composition. In section 3, the proposed methodology is analyzed and a framework is demonstrated. In the following section, a case study is presented, based on the proposed framework and pertaining to the composition of services in the context of e-commerce transactions. Finally we conclude with a discussion on the lessons learned and point towards future research and areas of interest.

2 Related Work

The notion of utilizing LoD in the process of the discovery, selection and composition of WS has recently gained the attention of researchers. Nonetheless, in contrast to the corresponding semantic approaches, no specific standards have been established and the optimal utilization of this approach is still being debated.

The work presented in [16] pertains to the examination of the benefits of a LoD-based composition framework for RESTful services. The author discusses the similarities between LoD and RESTful services as both technologies rely on URI-described resources and highlights the potential of a proposed composition approach, though an implementation is lacking. The construction of value-added services based on RESTful resources, through the utilization of LoD is also examined in [12]. The authors discuss a methodology, where services are identified using graph patterns which also aid the construction of the appropriate RDF messages for the exchange of data between WS. This is a

promising step towards automated compositions, but no elaboration is presented regarding the problem of selecting among WS with similar functionalities, which is a focal point in our approach.

In [14] the concept of Linked Open Services is presented, which is based on the utilization of LoD, RDF and SPARQ queries. Their approach is aiming at the discovery of WS that are described through interlinked data. In addition, in [15], authors elaborate on the applicability of SPARQL graph patterns for the description of such services. In more detail, they analyze the behavior of registries that are compatible with RESTful calls and LoD, and proceed to the implementation of such an approach using Hadoop. While authors also discuss composition methodologies, the focus of their approach is on the discovery of WS through RFD-based data.

In [10], authors describe the application of a LoD-as-a-service architecture. They highlight the challenges of applying domain specific datasets in a LoD format and developing appropriate architectures for the retrieval of these data. A dataset regarding WS descriptions can be added into such an implementation, thus providing WS discovery capabilities. We build on this notion, as we intent to provide enterprises with a methodology for the selection and composition of WS based on RDF descriptions. Our approach is differentiated, as we also include a methodology for the selection of optimal service compositions between a set of alternatives, utilizing those descriptions.

In [6], a service identification methodology is proposed, which enables the application of LoD and semantic links for the discovery of APIs. Authors describe the application of such an approach and demonstrate the affectability on two popular API repositories. While not tackling the problem of automated WS composition, it provides a cross-platform WS searching methodology.

Finally, in [1], Linked-OWL a modification to the OWL-S language is proposed which can describe both ontologies and LoD data. The proposed language takes advantage of REST-based services and is a promising approach towards the composition and dynamic reallocation of WS utilizing LoD. The approach, however, lacks a concrete selection mechanism. We believe that the integration of a ranking algorithm, as presented below, is an important step in the overall procedure, providing a performance-oriented solution.

3 Proposed Methodology

In this work, we propose a conceptual framework for the discovery of business oriented services, provided by a number of enterprises through an appropriate repository, based on descriptions stored in RDF format. Those descriptions provide details pertaining to both functional and non-functional charac-

teristics and can be accessed through the corresponding SPARQL requests. Through the application of a proposed algorithm, the selection and composition process of these services can be aided. In such an approach, information regarding the available WS must be converted in an RDF-based format, such as the following tuple structure:

{WS_Name : Property1 : Value}

{WS_Name : Property2 : Value}

...

{WS_Name : PropertyN : Value}

By converting the aforementioned information into an RDF format, an RDF graph can be created, modeling all the available WS, their functional and non-functional characteristics and their interconnections. Such a graph enables the monitoring of the overall data and the identification of requested information, through appropriate queries.

The proposed framework includes a LoD-based repository containing descriptions of all available WS. Using a SPARQL endpoint the end-user is capable of issuing queries regarding to specific operations while a number of requested QoS characteristics can also be included in such a query. The framework's overall architecture is depicted in figure 1.

An important module in the framework is the response parser. SPARQL responses can be in a number of predefined formats (such as XML, JSON, CSV and TSV) and the retrieved information will then be used as input for the composition methodology. In order to adhere to this, the parser is responsible for the transformation of the response to an appropriate format.

After the aforementioned transformation, a ranking of alternative solutions is required for the identification of the optimal composition. A number of researchers have utilized the PSO algorithm to adhere to similar problems in a variety of domain specific scenarios. The application of PSO for solving the WS selection and composition problem, based on QoS characteristics, is feasible as shown in [9] [18].

In this work we apply this algorithm as a part of the selection process, nonetheless other selection algorithms can also be applied, such as multicriteria decision analysis methods [17] or constraint algorithms [4]. Compared to these approaches, the application of the PSO algorithm is less effort demanding, while providing a more performance-oriented solution. Multicriteria and constraint methods, on the other hand, can be used in compositions where a higher level of personalization is required.

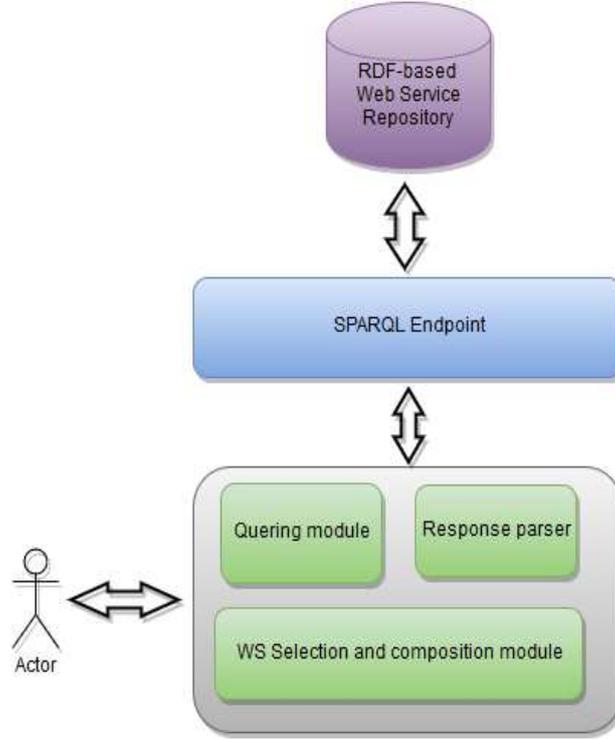


Fig. 1. The proposed framework architecture

The PSO algorithm, was introduced in [11] and is described by authors as an algorithm simulating a social model, and in particular the behavior of bird flocking and fish schooling.

Two important values in the algorithm are the personal best and the neighborhood best, p_{best} and p_{gbest} accordingly. Each simulated particle (corresponding to a member of the swarm and to a potential candidate solution), is attracted to these two values, as it tries to identify a position better than its present. This attraction influences the particle's position and current velocity.

By a random alteration of the p_{best} and p_{gbest} values, all particles are in constant movement trying to converge to the point of the optimal values. Each repetition of this procedure represents in iteration i . For each iteration, the updated velocity v and current position x of a particle is being calculated according to the following formulas:

$$v(t+1)_i = wv(t)_i + c_1r_1(p(t)_{best} - x(t)_i) + c_2r_2(p(t)_{gbest} - x(t)_i) \quad (1)$$

$$(t+1)_i = (t)_i + v(t-1)_i \quad (2)$$

where w is the inertia weight, a variable changing values from a maximum of 0.9 to a minimum of 0.4, thus controlling the influence of a particle's current velocity to the one it will gain in the following iteration. In addition c_1 and c_2 pertain to the cognition and social weights, allocated to the p_{best} and p_{gbest} values accordingly while r_1 and r_2 refer to random values in the $[0,1]$ scale. After a number of iteration all particles are drawn near to the optimal values [20].

In order to minimize the time required for the selection and composition of requested services, we propose the following series of steps that include the application of PSO:

1. Execute SPARQL query regarding the requested functional characteristics
2. If the number of results is higher than one, issue request regarding to non-functional characteristics
3. Retrieve a list of appropriate compositions and the values of non-functional characteristics in the predefined format
4. Parse the list and construct the utility function
5. Fetch the above to the PSO method
6. Return the optimal composition along with the corresponding WSDL files.

4 Case Study

We demonstrate the effectiveness of the proposed work through the application of a case study. We have used the dataset described in [2,3] which includes a number of real world WS and a value for their QoS attributes. We selected a subset of those services which can be used in the context of e-commerce transactions and converted their description and QoS values into appropriate RDF tuples. In real-life applications, the values of QoS parameters can be applied by the RDF-based repository. They can be attained by closely monitoring the behavior along with the invocation and response times of a WS (for parameters such as availability or response time), or can be assigned manually according to the corresponding attributes (for properties such as best practices and documentation). The structure of an e-commerce service in the converted RDF dataset is as follows:

- {WS1: Functionality: OrderProductService}
- {WS1: Reliability: 0.58} (*value is in a scale of 0-1*)
- {WS1: Resp_time: 0.24275} (*value in seconds*)
- {WS1: Best Practices: 0.84} (*value is in a scale of 0-1*)
- {WS1: Documentation: 0.38} (*value is in a scale of 0-1*)

The sum of all involved services comprise the RDF graph. Using the appropriate SPARQL queries a number of services can be retrieved by the end-user, that fulfill his given functional requirements. Utilizing the series of steps provided in the previous section, a test case is presented below, as proof of concept of the proposed framework. Suppose that an end-user request a composition for an e-commerce application. The user requires services for product ordering, credit card validation and payment, as well as a sms service in order to be informed of any alterations in his order's state. After issuing the request, and based on the modeled RDF dataset, a list of 96 potential compositions is returned. The user is then prompted to select his requested values in the following QoS characteristics: Reliability, Response Time, Best Practices, and Documentation.

Suggesting that all services follow a serial execution path, the overall Response time, based on [19,20], is:

$$RT_o = \sum_{i=1}^n RT_i \quad (3)$$

The corresponding best practices value is:

$$Bpr_o = \prod_{i=1}^n Bpr_i \quad (4)$$

The score pertaining to documentation is calculated based on:

$$Doc_o = \prod_{i=1}^n Doc_i \quad (5)$$

While the overall reliability, based on [5] is:

$$Rel_o = \prod_{i=1}^n Rel_i \quad (6)$$

Regarding the value of reliability, a multiplication aggregation type is applied, as in complex compositions a malfunction in one of the involved WS can greatly affect and compromise the whole value-added service. Thus, the overall reliability should be the product of the corresponding values of each involved service, instead of a simple mean. We opted to use a multiplication aggregation for the calculation of the overall documentation and best practices value as well, following the logic above, as those properties could play a pivotal role in the selection of services for some users.

By modeling the requested problem the optimal service composition can be returned. As a starting point, after the parsing of the response, the objective function must be constructed. As a serial execution flow is presumed this function receives the following form:

$$Of = Min \left(\left(\frac{1}{Bpr_o} \right) + \left(\frac{1}{Doc_o} \right) + \left(\frac{1}{Rel_o} \right) + RT_o \right) \quad (7)$$

In more detail, by inverting the values of the best practices, documentation and reliability attributes (where the higher the scores the better) and maintaining the state of the response time parameter, a minimum function is created. The optimal service composition, among the alternatives, is the one that produces the minimum value.

For the needs of this simplified case study, the values of the first three parameters were added and stored into a matrix (representing their summed value in all possible compositions), while a similar matrix was used for the response time parameter. The result of executing the, now less complex, objective function is shown figure 2.

As demonstrated in the figure below, the velocity and position of each particle is altered between iterations, as particles gradually converge towards the solution. Using the returned values, a composition engine can easily determine which WS best fulfils the user's business needs. The number of required iterations and the overall execution time is based on the complexity of the objective function.

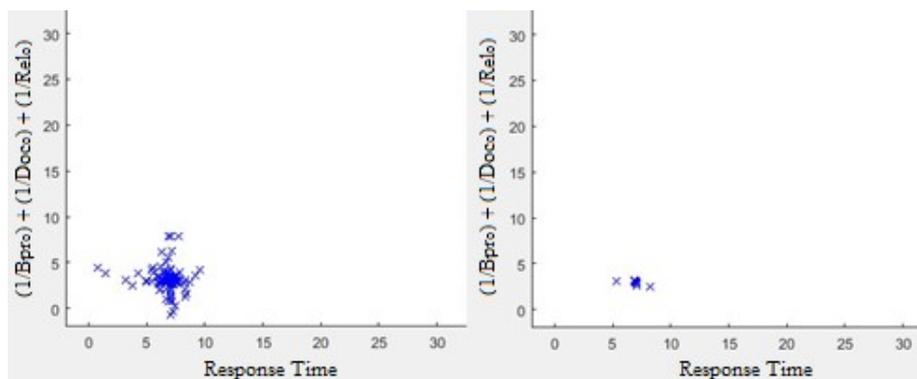


Fig. 2. Swarm converges to optimal values after 15 (left) and 30 (right) iterations

In this case study, most particles point towards the optimal composition after 30 iterations, proving both the feasibility of utilizing the PSO algorithm in the context of WS compositions, as well as its ability to provide results in reasonable time frames. In addition, the algorithm does not require as much human input as other similar approaches, such as multicriteria methods, that are more effort-demanding. While the case study demonstrates a scenario involving a few possible alternatives, in real life business scenarios a large number of WS can be involved and the size of required calculations rises exponentially. In order to provide scalability to the overall composition, the PSO algorithm can be used to enhance the performance of a composition engine,

especially when there is a need for dynamic alteration of involved services (often a necessity in the rapidly changing business environments). In addition, the PSO algorithm provides users with the much desired flexibility to differentiate the composition, even in real-time, by using personalized user weights. Combining such an algorithm with service repositories that utilize the, machine understandable, RDF format is a step towards the automation of WS compositions and the minimization of required human intervention.

5 Conclusion & Future Work

We have demonstrated an approach for the selection and composition of WS, based on LoD and SPARQL endpoints. Our approach combines the benefits of utilizing semantic descriptions as well as those offered by the structured nature of RDF data. Through the application of a case study, we examined the feasibility of the proposed methodology and the integration of a performance-based ranking algorithm, with promising results. Future work involves the implementation of the framework along with a corresponding GUI, utilizing the Apache Jena framework and the Fuseki server for the development of SPARQL endpoints that are accessible through HTTP calls. In addition, we plan to investigate modified versions of the PSO algorithm in order to support both serial and parallel execution flows and to enhance the overall composition performance. Towards this direction, we plan to evaluate the framework in more complex business scenarios, involving numerous execution tasks and alternative services. Finally, we work towards the automated generation of SPARQL queries, by transforming plain text requests. This will allow the utilization of the proposed framework by non-programmers and ease the overall procedure.

References

1. Ahmad, H., Dowaji, S.: Linked-OWL: A new approach for dynamic linked data service workflow composition. *Webology*, 10(1), 21-30. (2013)
2. Al-Masri, E., Mahmoud, Q. H.: Discovering the best web service. In: 16th International Conference on World Wide Web, pp. 1257-1258. (2007)
3. Al-Masri, E., Mahmoud, Q. H.: QoS-based Discovery and Ranking of Web Services. In: IEEE 16th International Conference on Computer Communications and Networks, pp. 529-534. (2007)
4. Alrifai, M., Risse, T., Nejdl, W.: A hybrid approach for efficient Web service composition with end-to-end QoS constraints. *ACM Transactions on the Web*, 6(2). pp 1-31. (2012)
5. Alrifai, M., Skoutas, D., Risse, T.: Selecting skyline services for QoS-based web service composition. In: 19th international conference on World wide web, pp. 11-20. (2010).

6. Bianchini, D., De Antonellis, V., Melchiori, M.: A linked data perspective for effective exploration of web APIs repositories. In: Daniel, F., Dolog, P., Li, Q. (eds.) LNCS, vol. 7977, pp. 506-509. Springer Berlin Heidelberg. (2013)
7. Chen, W., Paik, I.: Improving efficiency of service discovery using Linked data-based service publication. *Information Systems Frontiers*, 15(4), 613-625. (2013)
8. Fahad, M., Moalla, N., Ourzout, Y.: Dynamic execution of a business process via web service selection and orchestration. *Procedia Computer Science*, 51, 1655-1664. (2015)
9. Kang, G., Liu, J., Tang, M., Xu, Y.: An effective dynamic web service selection strategy with global optimal QoS based on particle swarm optimization algorithm. In: 26th IEEE International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 2280-2285. (2012)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, 1995, pp. 1942-1948. (1995)
11. Kim, S., Berlocher, I., Lee, T.: RDF based Linked Open Data Management as a DaaS Platform. In: First International Conference on Big Data, Small Data, Linked Data and Open Data, pp. 58-61. (2015)
12. Krummenacher, R., Norton, B., Marte, A.: Towards linked open services and processes. In: Berre, A.J., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) LNCS, vol. 6369, pp. 68-77. Springer Berlin Heidelberg. (2010)
13. Neubert, J.: Linked data based library web services for economics. In: International Conference on Dublin Core and Metadata Applications, pp. 12-22. Dublin Core Metadata Initiative. (2012)
14. Norton, B., Krummenacher, R., Marte, A., Fensel, D.: Dynamic linked data via linked open services. In: Workshop on Linked Data in the Future Internet at the Future Internet Assembly, pp. 1-10. (2010)
15. Norton, B., Stadtmüller, S.: Scalable discovery of linked services. In: 4th International Workshop on Resource Discovery, pp. 6-21. (2011)
16. Stadtmüller, S.: Composition of linked data-based RESTful services. In: Philippe Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) LNCS, vol. 7650, pp 461-464. Springer Berlin Heidelberg. (2012)
17. Vesyropoulos, N., Georgiadis, C. K.: QoS-Based Filters in Web Service Compositions: Utilizing Multi-Criteria Decision Analysis Methods. *Journal of Multi-Criteria Decision Analysis*, 22(5-6), 279-292. (2015)
18. Wang, S., Sun, Q., Zou, H., Yang, F.: Particle swarm optimization with skyline operator for fast cloud-based web service composition. *Mobile Networks and Applications*, 18(1), 116-121. (2013)
19. Wang, W., Sun, Q., Zhao, X., Yang, F.: An improved particle swarm optimization algorithm for QoS-aware web service selection in service oriented communication. *International Journal of Computational Intelligence Systems*, 3(1), 18-30. (2010)
20. Zhang, T.: QoS-aware web service selection based on particle swarm optimization. *Journal of Networks*, 9(3), 565-570. (2014)