An approach for Web Service selection and dynamic composition based on Linked Open Data

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. The wide adoption of the Service Oriented Architecture (SOA) paradigm has provided a means for heterogeneous systems to seamlessly interact and exchange data. Thus, enterprises and end-users have widely utilized Web Services (WS), either as stand-alone applications or as part of more complex service compositions, in order to fulfill their business needs. But, while WS offer a plethora of benefits, a significant challenge rises due to the abundance of available services that can be retrieved online. In this work, we propose a framework for the selection and dynamic composition of WS, by utilizing Linked open Data (LoD). In addition, we propose a hybrid algorithm that uses as input the user's personalized weights for non-functional characteristics and the results produced by appropriate SPARQL queries that are filtered results using a top-k approach. It then handles the ranking of alternatives based on their population. Finally, using two case studies and a dataset that describes real-world WS, we argue on the feasibility and performance of the proposed method.

Keywords: Web Services · Web Service Composition · Linked Open Data · RDF · Particle Swarm Optimization

This paper is a revised and expanded version of the following paper, Vesyropoulos N., Georgiadis C.K., Pimenidis E. (2016) Utilizing Linked Open Data for Web Service Selection and Composition to Support e-Commerce Transactions. In: Nguyen NT., Iliadis L., Manolopoulos Y., Trawiński B. (eds) Computational Collective Intelligence. ICCCI 2016. Lecture Notes in Computer Science, vol 9875. Springer, Cham https://doi.org/10.1007/978-3-319-45243-2_49

This work provides a much more detailed review of the current literature relevant to our research. It also provides a detailed presentation and analysis of the architecture of the proposed methodology and a thorough insight of the proposed algorithm. The results of the cases considered are discussed and evaluated in detail and further conclusions are drawn.

1 Introduction

In recent years, businesses and end-users have widely utilized the Service Oriented Architecture (SOA) paradigm, in order to enable heterogeneous systems to seamlessly exchange data and messages. Enterprises, especially, have reaped the benefits of adopting this approach for the fulfillment of their business needs, as the communication with various partners, such as suppliers and external affiliates, as well as with customers have been significantly simplified. An important aspect of the integration of Web Services (WS) into their business workflows is the fact that WS can be utilized as standalone components or as parts of more complex service compositions [10]. This enables the reuse of services that are internal or external to the enterprise, thus lowering the cost for research and development. In addition, service compositions provide the means for flexible and dynamic reallocation of individual services, in order for the value-added service to constantly comply with the business needs that arise due to the rapidly changing external environment of modern enterprises. These capabilities are of immense importance in scenarios that involve online transactions, such as in the context of e-commerce [29].

For the purpose of discovering services that comply to specific functional requirements, as well as for the selection and composition of WS, some researchers have focused on the application of semantic rules. This approach is based on the development of appropriate ontologies describing a service's operations and properties in a machine processable format. Furthermore it allows a composition engine to automatically process orchestration and choreography rules and construct compositions based on requested properties [26]. In this work we focus on an alternative approach, which is the use of Linked Open Data (LoD) for the identification of services that adhere to both functional and non-functional requirements and ease the discovery of similar services, an operation of high importance in dynamically reconfigurable service compositions [9].

LoD, which pertain to machine understandable structured data, have been widely adopted in an abundance of scientific fields, such as economics and business management [21]. The structured format used to describe data is the Resource Description Framework (RDF) format, which is based on the representation of data as machine-understandable structured triples, which enable the automated discovery of information and exchange of data [11]. This is possible, as in LoD scenarios data are interconnected through the application of semantic rules. Through the connections they share, data form RDF graphs, where nodes represent the individual data and arrows comprise the relations between them. Such a graph, depicting the functional and non-functional

characteristics of three WS (named WS1, WS2 and WS3) and their interconnections, is given in figure 1. Utilizing this structured approach, data can be easily identified and retrieved through appropriate SPARQL queries. SPARQL is a graph-matching query language for RDF formatted data [24]. It enables the identification of data through matching a query pattern to one or multiple data sources.

Fig. 1. RDF graph describing the interconnections of WS with similar functionality

As semantic descriptions can aid the automated selection and composition of WS and the interlinked nature of data in LoD can simplify the identification

of WS with similar functionality, significant opportunities can rise from the utilization of LoD in the domain of automated WS composition. Such opportunities include the development of value-added services, without the need for human intervention as well as the dynamic reconfiguration of existing compositions, in cases where an involved service ceases to fulfill certain requested properties and must be replaced [5].

In this work, we describe a proposed framework for the selection and composition of WS as well as a hybrid algorithm that combines notions from a meta-heuristic and a multi-criteria decision analysis approach.

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 hybrid algorithm, as a step towards the automation of WS compositions. We focus on WS that can be utilized in order to support e-commerce transactions, as in this domain any differentiation in the Quality of Service (QoS) characteristics of involved services in a WS composition can play a pivotal role for the completion or abnormal termination of a business transaction, thus highlighting the importance of selecting the optimal services according to predefined business needs. A preliminary version of this work has been reported in [30].

The remaining of this work is organized as follows: section 2 pertains to the related works description, which sheds light on the benefits related to the utilization of LoD for the identification of WS and to the application of ranking algorithms for the selection and composition process. In section 3, we present the proposed framework and hybrid algorithm. In the following section, we provide two case studies that demonstrate both the feasibility and the performance of our approach for the composition of services in the context e-commerce transactions. Finally, in section 4 we discuss the results of the utilization of the proposed method and point towards future research and areas of interest.

2 Related Work

A) Application of LoD in WS identification and selection

During the past few years, a number of researchers have examined the applicability of LoD for the identification, selection and composition of WS. Nonetheless, in contrast to previous semantic approaches, there has been a lack of specific standards that would lead to the optimal utilization of this approach.

In [25], the author describes the opportunities that could benefit enterprises and end-users through the application of an LoD-based composition framework, which is theoretically described and mainly focuses on the inclusion of RESTful WS. This work highlights the similarities between LoD and RESTful services, such as the fact that they both rely on the utilization of URIs for the identification of resources. As a result, a composition approach based on shared and interlinked resources is examined, though an implementation of such a framework is lacking.

The composition of services that are interlinked and adhere to the RESTful architecture constraints is examined in [18]. The researchers propose a methodology based on LoD and graph patterns for the identification of services that fulfill certain functional and non-functional requirements. These patterns aid the construction of appropriate RDF messages for the interoperation and information exchange between the involved WS of the value-added service. While the notions presented can enable the automated composition of services, there is limited discussion regarding the handling of the selection process between WS that provide similar operations and are only differentiated in QoS characteristics. This is a focal point in our proposed methodology, as the abundance of available WS can constitute the identification of the optimal alternative service a challenging task.

Authors in [22] introduce the notion of Linked Open Services, in which linked data, RDF structured information and SPARQL queries hold a pivotal role. They propose a mechanism for the identification of interlinked services based on the combination of the aforementioned technologies. In addition, in [23], SPARQL graph patterns are analyzed, that can provide a graphical representation of descriptive data for linked services. In addition, they elaborate on the architecture and behavior of LoD-based service repositories that can handle requests for RESTful services. An implementation of this approach in Hadoop is also being demonstrated. Nonetheless a discussion of the applicability of compositional approaches is lacking, as the focus of their approach is on the discovery services.

Researchers in [17] 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 the work presented by [7], a technique for the identification of services that can be included in value-added compositions is proposed. The methodology enables the discovery of API's that correspond to the user's needs, through the application of LoD and semantic rules. 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 [2], 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.

B) Utilization of MCDA methods and metaheuristic algorithms for WS selection and composition

As the vast number of alternative solutions in WS composition scenarios can often turn the selection of the optimal one into a challenging task, researchers have focused on the utilization of ranking algorithms in order to solve this complex issue. As different users may have different requirements in terms of QoS properties and place more weight on specific characteristics (e.g. in guaranteed secure transactions or high availability), these preferences must be taken into consideration when ranking the alternative solutions. This need can be fulfilled by MCDA methodologies, thus resulting into their high adoption in WS composition scenarios. In more detail, these methods enable the ranking of services or compositions, through the detailed evaluation of how alternatives cope in multiple criteria set by the end-user. As users can also set specific weights that correspond to the importance they place on certain criteria, those methodologies can provide solutions that are characterized by a high degree of personalization [28].

The benefits of applying such a methodology are highlighted in [16], where authors apply a hybrid methodology, based on the Analytical Hierarchy Process (AHP), in order to optimize the solution returned to an end-user, after a composition request, based on a number of predefined QoS requirements. They focus on service compositions in the domain of network architectures,

even though their proposed algorithm can be applied in multiple domains, such as for e-commerce transactions. In addition, authors in [8] demonstrate the applicability of the methodology even in scenarios where there are uncertainties regarding the QoS values of certain alternatives, using a variant of AHP, named Fuzzy AHP.

In [27] authors demonstrate a set of ontologies that describe QoS characteristics of SOAP-based WS and a technique to handle and evaluate those ontologies through the application of AHP. As a result, they provide a methodology for the ranking of alternative services, through the utilization of semantic descriptions. In doing so, they provide the means for discovering the optimal service to be included into a composition. Nevertheless, the applicability of the methodology for the evaluation of RESTful services is not being discussed and RESTful-specific descriptions are not included into the ontology. An alternative approach is demonstrated in [14], where authors apply the PROMETHEE methodology for the development of value-added services, based on QoS criteria. PROMETHEE enables the ranking of alternative solution and is less effort demanding than AHP, as the necessary input required by the end-user is significantly lower.

While MCDA algorithms, have the notion of personalization as a focal point, meta-heuristic algorithms are more performance oriented, as they are utilized in order to solve optimization problems. Since the identification of the optimal alternative composition can be regarded as such a problem, meta-heuristic algorithms have been widely used by researchers in WS composition scenarios. In more detail, swarm intelligence algorithms, such as the Ant Colony Optimization (ACO) and the Particle Swarm Optimization (PSO) algorithms have been applied in this specific domain [12], [13]. In [19], a variation of the PSO algorithm is proposed, which is based on the identification of local best values and the utilization of a number sub-swarms instead of a single swarm. The authors apply the algorithm in a service selection problem with promising results. In a similar manner, in the work presented in [34], the PSO algorithm is combined with a Local Best First strategy, and thus gives higher priorities to solutions found in a local level, that is to solutions that have a better local fitness value. In doing so, authors manage to significantly shorten the execution time of the PSO algorithm when applied in a service composition scenario.

Finally, metaheuristic algorithms are often combined with MCDA methods. In these approaches MCDA methods are used in order to minimize the available services that are provided as input to the metaheuristic algorithm, thus reducing the time required to solve the optimization problem. Such an approach have been demonstrated in [31], where a skyline operator is also

included, that is based on the SAW method. Our approach instead, provides a hybrid algorithm, that utilizes the notions applied in the SAW method in scenarios where a limited number of available services exist, while shifting to the application of PSO in cases more alternatives are present. In addition, it relies on a top-k approach [20] for filtering out individual services that are not considered attractive, according to the user's needs, thus reducing the composition candidates.

3 Proposed Methodology

In this section, we propose a novel hybrid algorithm that combines a meta-heuristic approach with an MCDA methodology, as well as 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 characteristics and can be accessed through the corresponding SPARQL requests. Through the application of the proposed algorithm, which is utilized by the framework, 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. As the graph connects services with identical functionality, it provides the means to a composition engine to dynamically reconfigure a composition, by removing a previously selected WS and replacing it with an alternative that is at the time more suited to fulfill a number of requirements set by the end-user, such as requirements in QoS characteristics.

A) Proposed framework

In order for businesses and end-users to reap the benefits provided by the utilization of LoD in WS composition scenarios, we propose a framework which includes repositories that contain descriptions of available services in structured RDF format. As the functional and non-functional characteristics of services are provided as RDF tuples, end-users or composition engines can identify and retrieve information regarding services that can fulfill certain operations and comply with requested QoS properties, using simple SPARQL queries. For this reason, the proposed framework also includes a SPARQL endpoint, responsible for the handling of such queries.

Fig. 2. The proposed framework architecture

In addition, as responses from a repository can be in a number of predefined formats (such as XML, JSON, CSV and TSV), a response parser, that handles the transformation of the response to a format suitable for the compo-

sition engine is also necessary. Finally, a composition engine is responsible for the selection of the optimal service composition, through the application of a ranking algorithm and for the implementation of composition based on the provided business logic. The framework's overall architecture is depicted in figure 2.

B) The proposed hybrid algorithm

As already mentioned, the wide adoption of the SOA architecture, by users and enterprises, have led to an abundance of existing WS available on the Web. The problem escalates greatly through the introduction of “physical” services, which are services that originate from the adoption of the Internet of Things (IoT) notions. In more detail, “physical” services pertain to services that are provided by “smart” objects that can remotely allow access to their functionality in the form of a WS, through online HTTP requests. As more and more objects of the physical world, equipped with sensors and embedded systems, can access the Web, the number of alternative services available for consumption grows even larger thus making the problem of service selection and composition even more challenging. As a result, any request for a value added service may return a vast number of alternative compositions. Thus, algorithms that can rank alternative compositions and identify the one that is better suited for the end-user based on his preferences, is required.

In order to address this issue we propose a novel algorithm that differentiates its operation according to the number of returned alternative compositions, thus is able to handle both small-scale and large-scale composition problems. When a limited number of alternatives are returned, those are ranked using the principles behind the Simple Additive Weighting (SAW) method. SAW is one of the most commonly used MCDA methods and has been applied in a large variety of fields [1]. The method can provide results in reasonable time frames and is not effort demanding. As a result it is ideal for the automated service composition scenarios that are performance oriented, especially when there is a limited number of composition candidates. SAW is based on the following formula:

$$A_i = \sum w_j x_{ij} \quad (1)$$

where A_i is the overall score of alternative i , w_j is the weight of criteria j and x_{ji} is the score of alternative i regarding criterion j .

When a larger number of alternative compositions exist, those can be ranked using the PSO algorithm. PSO was introduced in [15] 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:

$$v(t+1)_i = wv(t)_i + c_1r_1(p(t)_{best} - x(t)_i) + c_2r_2(p(t)_{gbest} - x(t)_i) \quad (2)$$

$$(t+1)_i = (t)_i + v(t)_i \quad (3)$$

where w is the inertia weight, a variable changing value 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 [33].

In both cases, an effective way to reduce the number of required calculations is the application of a top-k approach. Using this approach, unattractive alternatives are being filtered out and thus the number of alternative solutions is significantly lowered. SPARQL has an embedded method to help calculate the top-k returned results, using the ORDER BY command. We opt to utilize this technique when executing a query and before constructing the list of available compositions. We are then able to determine if it is more effective to rank the alternative compositions using function 1, or functions 2 and 3. In order to calculate the top-k alternatives, the two higher weighted QoS characteristics are taken into consideration.

The hybrid algorithm that is implemented in the proposed framework and enables the ranking of alternatives that are returned in structured format by SPARQL requests is presented below:

Algorithm 1. A hybrid algorithm for the selection of optimal compositions

Input: FC[n] /* Requested functional characteristics as string values*/
NC[m] /* Requested QoS characteristics*/
W[m] /*The corresponding weights for the QoS requirements*/
k /*number of top accepted results*/
thres /*threshold for the selection of appropriate method

Output: The optimal WS composition

```

    max1 = max_element (W[m]) /* identifies highest weighted criterion*/
    max2 = max_element (W[m-1]) /* identifies second best criterion*/
Load SPARQL request and return top-k results /*based on max1 & max2*/
If (length(results)==0)
    Return void;
Else if (length(results)==1)
    Return results[0];
Else if (length(results)<=thres)
    Load function defined in Eq. (1)
Else
    Load function defined in Eq. (2)
    Load function defined in Eq. (3)
End if

```

The above algorithm defines a number of execution steps. Firstly, the user is prompted to insert his requested functional and non-functional requirements as well as weights for the latter, signifying the importance of each criterion. In addition the user inputs a value for variable k that will later be used in order to define the number of top rated alternatives that the SPARQL request will return. The user is also prompted to define a threshold value. Whether the number of returned alternative solutions are higher or lower than this threshold the methodology that will handle the ranking process will differ. The SPARQL request is then issued and returns the top-k alternative compositions based on the two highest weighted QoS characteristics. In case the returned results are more than one, the ranking algorithm is then determined based on the aforementioned comparison.

4 Case Studies

A) Case study 1

In order to demonstrate the feasibility of the proposed algorithm and the applicability of the framework we present their application in a small-scale case study. For the needs of this case study, we have used the dataset described in [3], [4] 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 comprises 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 algorithm 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 [32], [33], is:

$$RT_o = \sum_{i=1}^n RT_i \quad (4)$$

The corresponding best practices value is:

$$Bpr_o = \prod_{i=1}^n Bpr_i \quad (5)$$

The score pertaining to documentation is calculated based on:

$$Doc_o = \prod_{i=1}^n Doc_i \quad (6)$$

While the overall reliability, based on [6] is:

$$Rel_o = \prod_{i=1}^n Rel_i \quad (7)$$

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:

$$Of = Min_{((\frac{1}{Bpr_o}) + (\frac{1}{Doc_o}) + (\frac{1}{Rel_o}) + RT_o)} \quad (8)$$

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 3. As demonstrated in the figure, the velocity and position of each particle is altered between iterations, as particles gradually converge towards the solution.

Fig. 3. Swarm converges to optimal values after 15 (left) and 30 (right) iterations

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. 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. Furthermore, the algorithm does not require as much human input.

This case study demonstrates a scenario involving a few possible alternatives, while in certain business environments, a large number of WS can be involved and the size of required calculations could rise 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.

B) Case study 2

After the evaluation of the feasibility of the proposed framework and hybrid algorithm and in order to demonstrate the effectiveness of our work, we present a case study that compares the proposed algorithm with similar approaches in a variety of composition scenarios, which differ in the number

available alternatives. For the needs of this case study we utilize the dataset described in [3], [4] which includes a number of real world WS and a value for their QoS attributes.

As in the previous case study, 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. We argue that in real-life applications, the values of QoS parameters can be applied by the RDF-based repository, either through extensive monitoring or by manual assignment. In this case study we suppose that the end-user is interested in more QoS characteristics than in the one presented in the previous sub-section. The structure of an e-commerce service in this converted RDF dataset is as follows:

- {WS1: Functionality: IssueInvoiceService}
- {WS1: Reliability: 0.49} (*value is in a scale of 0-1*)
- {WS1: Availability: 0.60} (*value is in a scale of 0-1*)
- {WS1: Successability: 0.87} (*value is in a scale of 0-1*)
- {WS1: Compliance: 0.68} (*value is in a scale of 0-1*)
- {WS1: Latency: 0.34} (*value in seconds*)
- {WS1: Resp_time: 0.223} (*value in seconds*)
- {WS1: Best Practices: 0.90} (*value is in a scale of 0-1*)
- {WS1: Documentation: 0.46} (*value is in a scale of 0-1*)

For this case study, we suppose that the end-user requires a composition of an e-commerce application, pertaining to the online ordering of an excursion in three different scenarios, while taking into consideration more QoS characteristics. In the first scenario the user requires a product ordering and a sms service. In the second scenario he also requests a credit card validation and payment service. Finally, in the third scenario he also requests a weather forecast service. In the first scenario returns a limited number of available solutions and in more detail 12 compositions, the second scenario returns 96 compositions and the third scenario returns 672 compositions. The user is then prompted to select his requested values in the QoS characteristics described in the previous case study along with Availability, Successability, Compliance and Latency.

Suggesting that all services follow a serial execution path, the overall Latency is:

$$L_o = \sum_{i=1}^n L_i \quad (9)$$

The corresponding availability value is:

$$Av_o = \prod_{i=1}^n Av_i \quad (10)$$

The overall Successability value is calculated based on:

$$Succ_o = \prod_{i=1}^n Succ_i \quad (11)$$

And finally, the overall Compliance is:

$$Compl_o = \prod_{i=1}^n Compl_i \quad (12)$$

As in the case of the reliability, best practices and documentation criteria, the availability, successability and compliance values are calculated based on a multiplication aggregation type, since in value added services, a malfunction in one of the involved WS can compromise the whole composition. When a large number of alternative solutions are returned, as is the case in the second and third scenario, and in order for the PSO algorithm to be executed the objective function must be calculated. As more QoS criteria are selected by the end-user, the resulting function is more complex than in the previous case study.

As a serial execution flow is presumed this function receives the following form:

$$Of = Min_{((\frac{1}{Av_o}) + (\frac{1}{Succ_o}) + (\frac{1}{Compl_o}) + (\frac{1}{Bpr_o}) + (\frac{1}{Doc_o}) + (\frac{1}{Rel_o}) + RT_o + Lo)} \quad (13)$$

By inverting the values of the availability, successability, compliance, best practices, documentation and reliability attributes (where the higher the scores the better) and maintaining the state of the response time and latency parameters, a minimum function is created. The optimal service composition, among the alternatives, is the one that produces the minimum overall score. The values of the first six 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 and latency parameter.

Figures 4, 5 and 6 demonstrate the performance of our algorithm, compared to a simple PSO approach and to that of a combination of PSO with a skyline operator [31]. Experiments were conducted on an Intel Core 2 Duo processor clocked at 2.8 GHz with 4GB RAM running Windows 7 SP3.

Fig. 4. Composition time in seconds, using 12 alternatives

Fig. 5. Composition time in seconds, using 96 alternatives

Fig. 6. Composition time in seconds, using 672 alternatives

As seen in the figures above our approach outperforms the application of a simple PSO algorithm in all three scenarios. In addition it outperforms the combination of PSO with the skyline operator in scenarios where a limited number of alternative services exist, while its results are comparable when a larger number of alternative compositions are returned. As a result, it can be considered a reliable approach, in business scenarios with a high degree of uncertainty regarding the expected number of alternative composition solutions.

5 Conclusion & Future Work

In this work, we have proposed a framework and a hybrid multi-level algorithm for the optimal selection and composition of WS, that are in the form of Linked Open Data and which can be identified through SPARQL queries. The selection process is based on functional and QoS criteria that are formatted in a structured RDF format, thus enabling the automated processing of semantic descriptions which is a step towards compositions that require limited human intervention. Through the application of two case studies using a dataset that describes real world services, we examined the feasibility and performance of the proposed methodology with promising results.

As future work we aim at the implementation of the framework using the Apache Jena framework and the Fuseki server for the development of SPARQL endpoints that are accessible through HTTP calls. Furthermore, we

plan to develop a modified algorithm that would take into consideration the possible parallel execution of involved services and thus modifying the estimated QoS criteria of the resulting compositions. In order to accomplish this task, we aim to examine the behavior of such an approach in large-scale business scenarios that involve a significant number of alternative services and required QoS criteria. Finally, we plan to implement a privacy-aware recommender engine for available WS, based on the properties of the RDF graph, the end-user's preferences and the anonymous evaluation of previous SPARQL queries from users with similar requirements.

References

1. Afshari, A., Mojahed, M., Yusuff, R. M.: Simple additive weighting approach to personnel selection problem. *International Journal of Innovation, Management and Technology*, 1(5). pp 511-515. (2010)
2. Ahmad, H., Dowaji, S.: Linked-OWL: A new approach for dynamic linked data service workflow composition. *Webology*, 10(1), 21-30. (2013)
3. Al-Masri, E., Mahmoud, Q. H.: Discovering the best web service. In: 16th International Conference on World Wide Web, pp. 1257-1258. (2007)
4. 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)
5. 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)
6. 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)
7. 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)
8. Büyüközkan, G., Çifçi, G.: A combined fuzzy AHP and fuzzy TOPSIS based strategic analysis of electronic service quality in healthcare industry. *Expert Systems with Applications*, 39(3), pp. 2341-2354. (2012)
9. Chen, W., Paik, I.: Improving efficiency of service discovery using Linked data-based service publication. *Information Systems Frontiers*, 15(4), 613-625. (2013)
10. 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)
11. Haase, P., Broekstra, J., Eberhart, A., Volz, R.: A comparison of RDF query languages. In: International Semantic Web Conference, pp. 502-517. Springer Berlin Heidelberg. (2004)
12. Jatoth, C., Gangadharan, G. R., Buyya, R.: Computational intelligence based QoS-aware web service composition: A systematic literature review. *IEEE Transactions on Services Computing*, (99). (2015)
13. 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)

14. Karim, R., Ding, C., Chi, C. H.: An enhanced PROMETHEE model for QoS-based web service selection. In: IEEE International Conference on Services Computing (SCC) (pp. 536-543). IEEE. (2011)
15. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, 1995, pp. 1942-1948. (1995)
16. Khondoker, R., Mueller, P., Bayarou, K.: Selecting the best communication service in future network architectures. In: Proceedings of ITU Kaleidoscope: Building Sustainable Communities (K-2013) (pp. 1-8). IEEE. (2013)
17. 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)
18. Krummenacher, R., Norton, B., Marte, A.: Towards linked open services and processes. In: Bere, A.J., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) LNCS, vol. 6369, pp. 68-77. Springer Berlin Heidelberg. (2010)
19. Liao, J., Liu, Y., Zhu, X., Wang, J.: Accurate sub-swarms particle swarm optimization algorithm for service composition. *Journal of Systems and Software*, 90, pp. 191-203. (2014)
20. Magliacane, S., Bozzon, A., Della Valle, E.: Efficient execution of top-k SPARQL queries. In: Proceedings of the International Semantic Web Conference (ISWC 2012), pp. 344-360. (2012)
21. 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)
22. 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)
23. Norton, B., Stadtmüller, S.: Scalable discovery of linked services. In: 4th International Workshop on Resource Discovery, pp. 6-21. (2011)
24. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and Complexity of SPARQL. In: Proceedings of the International semantic web conference (pp. 30-43). Springer Berlin Heidelberg. (2006)
25. 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)
26. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1), pp. 27-46. (2003)
27. Tran, V. X., Tsuji, H., Masuda, R.: A new QoS ontology and its QoS-based ranking algorithm for Web services. *Simulation Modelling Practice and Theory*, 17(8), pp. 1378-1398. (2009)
28. 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)
29. Vesyropoulos, N., Georgiadis, C. K., Ilioudis, C.: Analyzing the selection and dynamic composition of web services in e-commerce transactions. In: Proceedings of the Fifth Balkan Conference in Informatics (pp. 130-135). ACM. (2012)
30. Vesyropoulos, N., Georgiadis, C. K., Pimenidis, E.: Utilizing linked open data for web service selection and composition to support e-commerce transactions. In: International

Conference on Computational Collective Intelligence (pp. 533-541). Springer International Publishing. (2016)

31. 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)
32. 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)
33. Zhang, T.: QoS-aware web service selection based on particle swarm optimization. *Journal of Networks*, 9(3), 565-570. (2014)
34. Zhao, X., Song, B., Huang, P., Wen, Z., Weng, J., Fan, Y.: An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition. *Applied Soft Computing*, 12(8), pp. 2208-2216. (2012)